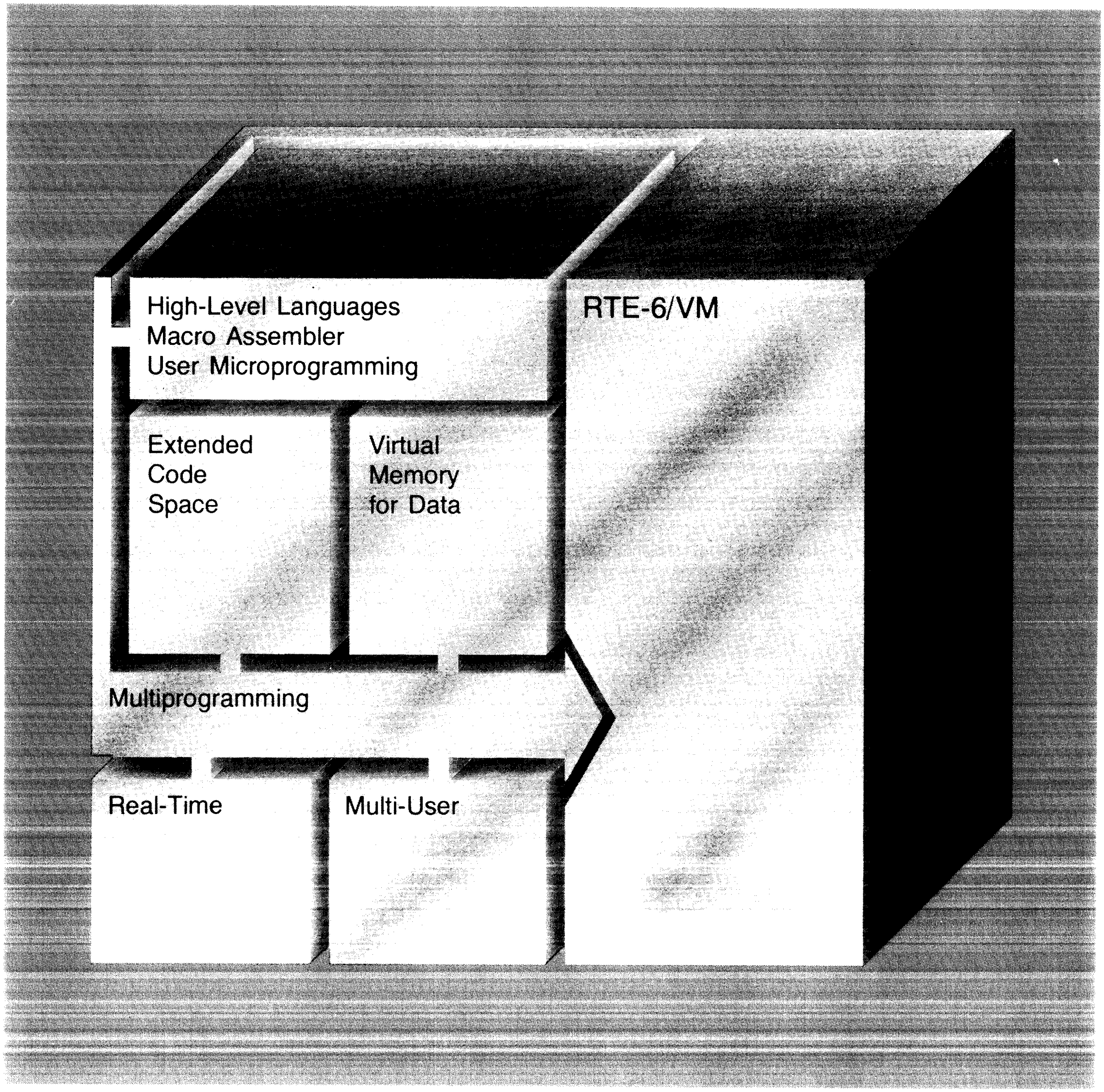
HEWLETT
PACKARD

RTE-6/VM
Terminal User's

Reference Manual



HEWLETT PACKARD

RTE-6/VM Terminal User's

Reference Manual

Data Systems Division
11000 Wolfe Road
Cupertino, CA 95014-9974

Part No. 92084-90004
U0186

Printed in U.S.A. October, 1984
Updated January, 1986

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 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	Add VL and CU Commands
Update 2	Jan 1983	Add utility LUPRN
Reprint	Jan 1983	Updates 1 and 2 Incorporated
Update 3	Dec 1983	
Update 4	Oct 1984	Manual Errata
Reprint	Oct 1984	Updates 3 and 4 Incorporated
Update 5	Jan 1986	

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 language without the prior written consent of Hewlett-Packard Company.

Preface

This manual describes the scope, format, and use of the interactive operator commands available with the RTE-6/VM operating system. It is intended to be the primary reference source for operators who will be using the computer to perform data entry, file manipulation, program development, and other interactive tasks.

File management information in this manual is based on the File Management Package (FMP). Refer to the CI User's Manual for details of the Command Interpreter hierarchical file system.

The manual is organized into four separate sections to facilitate the referencing of information.

Chapter 1 gives a general description of the RTE-6/VM operating system features.

Chapter 2 describes the operating environment when under session control and not under session control. It describes how to log on and off the system, restricted access to operator commands based upon command capability level, and cartridge ownership when operating under session control.

Chapter 3 describes the file management commands that allow the operator to interactively manipulate data files and disc cartridges. File creation, global parameters, and procedure files are also described.

Chapter 4 describes the RTE and break-mode commands. Program states and I/O processing are also briefly discussed.

Also included are the following Appendices:

Appendix A Error Messages

Appendix B Logical Source and Load-and-Go Areas

Appendix C Tables and Directories

Appendix D Scheduling FMGR Interactively

For additional information on the RTE-6/VM operating system or any of its subsystems, refer to the RTE-6/VM Index to Operating Systems Manuals. This guide contains a complete glossary, commonly used tables, a general index to all RTE-6/VM manuals, and a documentation map. Some available manuals offering other levels of information that may be directly relevant to system operation and user applications are briefly summarized below:

* RTE-6/VM Programmer's Reference Manual:

  Describes the features provided by the RTE-6/VM Operating System which allow the user to programmatically use system services and/or control system resources. Detailed descriptions of the calling sequences which invoke system action, as well as, FORTRAN, Pascal/1000, and Macro/1000 examples are provided.

* RTE-6/VM Batch and Spooling Reference Manual:

  Describes the uses and requirements of the Batch Spool Monitor Subsystem for those who wish to use the batch processing and spooling features.

* RTE-6/VM System Manager's Manual:

  Describes planning, generating, and maintaining a system. In conjunction with the RTE-6/VM On-Line Generator Reference Manual, all system generation information is available.

* RTE-6/VM On-Line Generator Reference Manual:

  Describes detailed procedures for generating a new RTE-6/VM operating system without shutting down the existing RTE-6/VM system. Complete examples of each phase and heavily annotated worksheets are provided. The manual is primarily intended for System Managers and system programmers who are involved in the design and maintenance of total system configurations.

Table of Contents

Chapter 1 General Description

Chapter 2 Operating Environment

Chapter 3 File Manager Commands

Chapter 4 System and Break-Mode Commands

Appendix A Error Messages Index

Appendix B Logical Source and Load-And-Go Areas

Appendix C Tables and Directories

Appendix D Scheduling FMGR Interactively

List of Illustrations

List of Tables

Chapter 1
General Description

Introduction

RTE-6/VM is a disc-based operating system that provides the supervisory functions necessary to coordinate requests for, and allocation of, system services and resources. RTE-6/VM processes all decision and scheduling tasks internally unless overridden by user intervention. User requests for system action can be made by a "call" from within a program or interactively via an operator command.

Additional software packages available with RTE-6/VM complement the basic operating system. Program development capabilities are enhanced by an interactive editor, language compilers and assembler, and relocating loaders. File management and batch processing are provided through the file management system, spooling of input and output is provided by the spool monitor. The multi-user environment is enhanced by the Session Monitor. RTE-6/VM has the ability to run large programs via the MLS-LOC loader, and manage huge data areas via Virtual Memory, without user interaction or special programming.

The RTE-6/VM Index to Operating Systems Manuals contains a complete glossary, a general index to all RTE-6/VM manuals, and a documentation map. This guide should be used as a general reference document and should be kept "on-hand" when working with an RTE-6/VM system.

Operating System Features

As the major control element within the operating environment, RTE-6/VM provides the user with various services and automatically handles the machine-related functions associated with each service. The major features of RTE-6/VM are:

- Real-Time Programming
- Multiprogramming with Timeslicing
- Program Partitions
- Program Segmentation
- Resource Management
- Extended Memory Area
- Large shared data areas
- Very large programs (larger than 64K bytes) in memory
- Demand-paged virtual memory

A brief discussion of each of these features follows. For a more detailed description of the system and its features, refer to the RTE-6/VM Programmer's Reference Manual and the RTE-6/VM System Manager's Manual.

Real-Time Programming

The Real-Time Executive allows event and time scheduling of programs, in addition to program, operator and batch scheduling. In RTE-6/VM, external events are serviced after they cause interrupts to the operating system. Time scheduling is implemented through a time list which is checked every 10 milliseconds by the operating system.

Multiprogramming With Timeslicing

The Multiprogramming feature of RTE-6/VM allows several programs to be active concurrently; each program executes during the unused central processor time of the others. Programs can be memory-resident or disc-resident.

Disc-resident programs are swapped in and out of partitioned memory in accordance with availability of system resources, program priority, and time scheduling criteria. Programs may be scheduled by predetermined time intervals, an external event, operator command, or another program.

Depending on the program priority level, program scheduling may be conducted on a linear (i.e., first in, first out), or circular (with timeslicing) basis. Linear scheduling prevails above a specified program priority level (i.e., at a higher priority), and circular scheduling prevails below it.

Linear scheduling means that at a given priority level, the first program entering the queue has priority over another program subsequently entering. Circular scheduling means that all programs in the queue (at the same priority level) take turns executing for a specified time period (timeslice) regardless of they entered the queue. Programs of higher priority always have precedence over lower priority programs.

Program Partitions

Partitions are blocks of physical memory reserved for disc-resident programs. Partitions are defined during system generation but may be redefined during the reconfiguration process at system boot-up (refer to the RTE-6/VM System Manager's Manual for system reconfiguration details).

Partitions can be defined as real-time or background. In addition, they may also be specified as mother partitions or shareable EMA partitions (refer to the discussion below). Real-time programs run in real-time partitions. Background programs run in background partitions. If only one type of partition is defined, all programs will run in that type of partition.

One or more programs can be assigned to a partition and will only run in that partition. The partition may also be reserved so that no programs other than the ones assigned to it can run in it. Therefore, if there is only one program assigned to a reserved partition, that program will be memory-resident at all times.

For example, if only one real-time partition is defined in your system and it is reserved with certain programs assigned to it, all other real-time programs will run in any available background partitions.

A mother partition is a collection of smaller partitions called subpartitions. Mother partitions are generally used for running large programs or VMA and EMA programs. When the mother partition is not in use, programs can run in each of the subpartitions. A mother partition or its subpartitions can also be defined as a shareable EMA partition.

Shareable EMA partitions are used for running EMA programs that declare EMA data as shareable data. Every shareable EMA partition is associated with a label. If EMA data is to be shared with other programs, the declaring program will execute in one partition and the EMA data will reside in a shareable EMA partition. When a shareable EMA partition is not being used for data, other programs can run in this partition. Refer to the AS and UL system commands described in Chapter 4 for more information on shareable EMA partitions. Shareable EMA is explained in detail in the RTE-6/VM Programmer's Reference Manual.

Demand-Paged Virtual Memory

The Demand-paged Virtual Memory system provides RTE-6/VM with the capability to access very large data areas up to 128 megabytes. Program data resides in a Virtual Memory Area (VMA) located on disc and the page containing the data is swapped into memory when needed.

Extended Memory Area (EMA)

An Extended Memory Area is a subset of virtual memory where all the VMA data resides in memory. With EMA, program data can extend beyond the logical address space of a program up to the available physical memory. This allows an EMA program to process large amounts of data in main memory rather than swapping in blocks of data from disc. An EMA program can declare EMA data as local to the program or as a system wide labeled common area shareable by other programs. Without EMA the program and its data would be limited in size to 64K bytes unless loaded with the MLS-LOC loader (MLLDR). Refer to the RTE-6/VM Programmer's Reference Manual for further information on EMA programming.

Program Segmentation

The program segmentation feature allows large programs to be separated into a main program and related segments, thereby allowing them to execute in partitions smaller than their total size. While a segmented program is executing, its main is kept in the program's partition and its segments reside on disc or in memory. The segments on disc are brought into memory when needed. Segmentation can be implemented automatically by the MLLDR Loader or programmatically via an EXEC call. Refer to the RTE-6/VM Loader Reference Manual for a description of program segmentation.

Resource Management

The RTE-6/VM operating system allows cooperating programs to manage common system resources. A resource is any element within the RTE-6/VM environment that can be accessed by a user's program, e.g., an I/O device, a file, a program, a memory location, subroutine, etc.

This feature allows the user or program to manage a specific resource shared by a set of cooperating programs, so that utilization of these resources occurs in a coordinated manner.

Session Monitor

The Session Monitor provides a multi-user environment by controlling access to the system, separating users, and controlling system resources.

User access to the system is controlled by the LOGON program. To gain access to the system, the user must provide LOGON with the appropriate user identification information as stored in the Account File on disc.

Separation of users is aided by a private, group and system cartridge structure. A user is prohibited from accessing other users' or groups' disc cartridges.

Control of system resources is aided by confining the user to only those resources defined in his or her respective Session Switch Table (SST). The SST provides a mapping between session logical unit numbers (LUs), which the user addresses, to system logical unit numbers (LUs) where requests are processed.

File Management

File management can be accomplished either programmatically via calls to the File Management Package (FMP), or interactively via the program FMGR. This manual describes the interactive FMGR commands. For a description of FMP calls refer to the RTE-6/VM Programmer's Reference Manual.

Interactive File Manager (FMGR) commands can be used to create and manipulate disc or non-disc files, develop programs and procedure files, save programs and procedure files, initialize disc cartridges, and manipulate disc cartridges.

Batch Processing

Batch processing, the entry of one or more jobs for processing in a single job stream, is controlled by FMGR commands. The jobs may be stored on disc or on a peripheral input device. In either case, batch job operation is controlled through FMGR commands that delimit the job and that may be included with the job. Refer to the RTE-6/VM Batch and Spooling Reference Manual for batch processing details.

Spool Monitor

The Spool Monitor can increase the throughput of a job stream that is limited by slow peripheral devices by temporarily spooling the data onto a disc file. This allows input and output to be performed independently of each other and of job processing. Refer to the RTE-6/VM Batch and Spooling Reference Manual for spooling details.

Interactive Editor

The Interactive Editor, EDIT/1000, can be used to create, edit, and merge ASCII files. The Interactive Editor is described in detail in the EDIT/1000 User's Guide.

System Utility Programs

Standard system utilities are on-line programs that run under the RTE operating system and are called by the user to perform various processes, such as: program preparation, system status and program housekeeping. The presence of any utility program in the system is optional, depending upon site-specific requirements. The programs available are:

* Compile Utility (COMPL)
* Compile and Load Utility (CLOAD)
* MLS-LOC Loader (MLLDR)
* MLLDR Command File Utilities (SGMTR, SXREF)
* Relocating Loader (LOADR)
* Indexed Relocatable Library Utility (INDXR)
* Merge Utility (MERGE)
* File Backup Utility (FC)
* Disc Cartridge Save/Restore Utilities (WRITT/READT)
* Disc Backup (PSAVE, PRSTR, PCOPY)
* On-Line Generator (RT6GN)
* System Status Program (WHZAT)
* Interactive Debugger (DBUGR)
* Soft Key Programs (KEYS and KYDMP)
* Track Assignment Table Log Program (LGTAT)
* Source File Comparison Utility (SCOM)
* On-Line Driver Replacement Utility (DRREL, DRRPL)
* Help Utilities (HELP,GENIX,CMD)
* System Configuration Display Utility (LUPRN)

For information concerning the On-Line Generator, refer to the RTE-6/VM On-Line Generator Manual; for the loader utilities, refer to the RTE-6/VM Loader Reference Manual. All other interactive utility programs are described in detail in the RTE-6/VM Utility Programs Reference Manual. A short description of each of the system utility programs follows.

Compile Utility (COMPL)

The compile program enables the user to invoke any HP-supported compiler or Macroassembler. The utility will select the appropriate compiler or Macroassembler by checking the control statement (first statement required in the program), the specified compiler option, or the compiler parameter specified in the COMPL run string. In addition, COMPL will automatically outspool the listing to the LU requested if spooling is enabled in the system.

Compile and Load Utility (CLOAD)

CLOAD performs a composite function. It performs all the functions of COMPL and inputs the relocatable results to the loader (LOADR) for the creation of an executable memory-image program.

MLS-LOC Loader (MLLDR)

The MLS-LOC Loader, MLLDR, allows loading of very large programs on RTE-6/VM. MLLDR provides multilevel segmentation for large programs. Multilevel segmentation (MLS) allows as much code as required to be memory-resident. The MLS-LOC Loader accepts relocatable code for a large program from a command file, handles all program segment loading for the user, and produces an absolute load module that is ready for execution.

MLLDR Command File Utilities (SGMTR, SXREF)

The SGMTR utility is used to assist in generating the MLLDR Command Files necessary to load large programs using the MLS-LOC relocating loader (MLLDR). Another utility, SXREF, may be used to verify the correctness of user-created MLLDR command files and provide cross reference information on program modules.

Relocating Loader (LOADR)

The Relocating Loader program accepts relocatable programs, and outputs absolute load modules in conformance with loader control commands specified by the user. The loader will also purge unwanted or permanently loaded programs from the system.

Indexed Relocatable Library Utility (INDXR)

The INDXR utility program is used to build an indexed relocatable library from user files (type 5), increasing the speed at which the MLS-LOC Loader operates. When the MLS-LOC Loader searches an indexed library for a particular module, it need only search a directory instead of the whole file.

MERGE Utility (MERGE)

The Merge utility provides a simple way for files to be merged. Taking its input from a command file or interactively from a terminal, MERGE can concatenate files within an existing file or a file created by MERGE.

File Backup Utility (FC)

The FC utility program can be used to easily copy selected files between various media that includes discs and certain tape devices such as magnetic tapes, cartridge tape drives integrated in CS80 discs, and minicartridge tapes.

Disc Cartridge Save/Restore Utilities (WRITT/READT)

The disc cartridge Save/Restore utilities allow the user to save and restore peripheral disc cartridges using magnetic tape. Saving disc cartridges on magnetic tape is accomplished using the WRITT program. Cartridges are restored using the READT program.

Disc Backup (PSAVE, PRSTR, PCOPY)

The Disc Backup programs can be used either on-line or off-line to transfer data from disc to magnetic tape or vice-versa, copy data from disc to disc, verify successful transfer or copy operation, and to initialize a disc cartridge.

On-Line Generator (RT6GN)

The On-Line Generator permits use of an existing RTE-6/VM System to generate a new RTE-6/VM System according to user specifications. Generation can be directed from an answer file, logical unit or operator console.

System Status Program (WHZAT)

The WHZAT program provides status information regarding the current system environment. It will list active programs associated with the user's session or list all active programs and the current status of each program in the system. A list of programs having "father-son" relationship and a list of the status of every program partition in the system (occupied, non-occupied, or shared) can also be displayed. WHZAT can also provide a list of every program known to the system.

Interactive Debugger (DBUGR, MLSDB)

The DBUGR subroutine can be appended to a user program by specifying an option in LOADR. The MLSDB subroutine can be appended to a user program by the MLS-LOC Loader. The debugger subroutines can aid the user in checking for logical errors in a program through interactive control commands. Debugging is performed at the Macro/1000 level. Refer to the Debug Subroutine Reference Manual for a complete description of all DBUGR and MLSDB functions.

Soft Key Programs (KEYS and KYDMP)

The KEYS and KYDMP programs are used to create user-defined command sets for programming the soft keys on the HP 2645A/48A Display Station. Soft Keys provide the capability to enter entire sequences of commands with a single key stroke. The advantages are speed of entry and a reduction in operator errors during terminal entry sessions.

Track Assignment Table Log Program (LGTAT)

The LGTAT program displays the status of the system and auxiliary disc tracks (LUs 2 and 3).

Source File Comparison Utility (SCOM)

The SCOM utility is used to compare two text files and print the similarities and differences betweem them. The SCOM utility detects which words or lines have been inserted, deleted, or modified.

On-Line Driver Replacement Utility (DRREL, DRRPL)

The purpose of the On-Line Driver Replacement utility is to make it possible for the user to replace a driver on-line instead of executing a new system generation. This utility is a useful tool for developing and debugging drivers. Refer to the RTE-6/VM Driver Writing Manual for a complete description of this utility.

Help Utilities (HELP, GENIX, CMD)

The RTE-6/VM Help Utilities provide a means of generating help functions for any interactive program on the RTE operating systems. The HELP and CMD programs allow the user to interactively request more information concerning any system error or interactive command in the system. The GENIX utility creates an index file used by the HELP and CMD programs. This allows users to add their own application errors or commands to an indexed file, which can then be searched by HELP or CMD programs.

System Configuration Display Utility (LUPRN)

The configuration display utility LUPRN supplies the current system device configuration and identifies each device driver by its true name. The configuration tables can be ordered by system LU number, session LU number, or by device select code. LUPRN will execute in both session and non-session environments.

Chapter 2
Operating Environment

Introduction

Provided with the RTE-6/VM Operating System are two software packages which handle multi-terminal operation: the Session Monitor and the Multi-Terminal Monitor. The features provided by each of these multi-terminal handlers will be described in this section along with a description of their respective operating environments.

Operating With Session Monitor

Significant features provided by Session Monitor include:

* Controlled access to the system.

* Separation of users.

* Control of system resource usage.

The LOGON program controls access to the system by checking the identification provided by a user when attempting to log on with that stored in the Account File on disc. The System Manager initializes and maintains the Account File, which includes entries for all authorized users as well as other session related information such as the maximum number of users authorized to access the system at one time.

Separation of users is accomplished through the private, group and system cartridge structure. An automatic program renaming feature allows each user to have a copy of the program, so the user need not wait for another user to finish with the program first. Also provided is the capability to interactively set up spool files for I/O operations.

Control of system resource usage is accomplished through the Session Switch Table (SST) which is defined for each user. The SST defines which I/O devices are accessible to a specific session user. It provides mapping from session logical unit numbers (LUs), addressed by the user, to system logical unit numbers (LUs), where the I/O requests are processed. Another important Session Monitor feature, which provides control over system resource usage, is a capability level for each user restricting the user to a prespecified subset of the operator commands. Other Session Monitor features are: a spare cartridge pool where users may request temporary disc area, and the timeslicing of programs.

In addition to the above features, the Session Monitor also provides the following:

* On-Line help utilities (HELP, GENIX, CMD) which provide the user with a simple means (through the HE command) of receiving error explanations and suggested corrective action for any interactive program on the system. The user can also list the syntax and explanation of every interactive command in the system via the CMD program.

* A user HELLO file (FMGR procedure file) which the system automatically executes when the user logs on.

* Access to 254 system logical unit numbers (63 LUs at one time) through the session to system logical unit number mapping feature of the Session Switch Table.

* Station configuration independence which allows the user to refer to certain I/O devices by the same logical unit number (LU) regardless of the terminal on which he has logged on. For example, a session terminal's CRT can always be refered to as LU 1, its left cartridge tape unit as LU 4, and right cartridge tape unit as LU 5. This is accomplished through the Session Switch Table (SST) described above.

* User message files which allow session users to send and receive messages from other session users via the File Manager SM and ME commands.

Terminals which are not controlled by the Session Monitor terminal handlers (PRMPT and RPN) will not have access to most of these features. Examples of non-session terminals are the system console, which can be set to operate in or out of the session environment, or dedicated application terminals. Operation under both of these types of terminals are further discussed later in this chapter.

Log-On Procedure

The log-on procedure is initiated by pressing any key on a currently enabled (but not currently logged-on) session terminal.

```
+-------------------------------------------------+
|                      NOTE                       |
|                                                 |
|    If the terminal is a Multipoint terminal,    |
|    the enter key must be pressed.               |
+-------------------------------------------------+
```

The PRMPT program then issues a prompt string to the terminal requesting the user's identification. The prompt string can be defined by the System Manager or the default prompt, "PLEASE LOG ON:", may be used.

The LOGON program then reads the user's identification and checks the Account File for a match. The format for the user identification is as follows:

```
+------------------------------------------------------------+
|                                                            |
|         "PLEASE LOG ON:"  XXXX.ZZZZ                        |
|                                                            |
|------------------------------------------------------------|
|                                                            |
|  XXXX    is the user account name defined  in the Account  |
|          File (up to 10 characters), upper or lower case.  |
|                                                            |
|  .       is the separator for user and group information.  |
|                                                            |
|  ZZZZ    is the group name  the user  is  associated with  |
|          (up to 10 characters), upper or lower case.  The  |
|         default group name is GENERAL.                     |
|                                                            |
+------------------------------------------------------------+
```

An example of a valid user identification is:

```
       PLEASE LOG ON:  BROWN.MANUF
```

In addition to the user identification, a password may be required for additional security. If the Account File indicates that a password is required by the user, the user is prompted with "PASSWORD?". The format of the password is as follows:

```
+-----------------------------------------------------------+
|        PASSWORD?  YYYY                                    |
|-----------------------------------------------------------|
|                                                           |
|  YYYY    is the user password  defined in the  Account File |
|          (up to 10 characters),  upper or lower case.  Note |
|          that YYYY will not be echoed to the terminal. Echo |
|          suppression  is used to  maintain  privacy  of the |
|          password.                                        |
+-----------------------------------------------------------+
```

The password can also be provided with the user identification. If the password was COST, the user would type in:

```
          PLEASE LOG ON: BROWN.MANUF/COST
```

In this case, the user will not be prompted for the password, and the password will be echoed back to the terminal.

If the user identification is not recognized as being valid by LOGON, the following error message will be returned to the user's terminal:

```
   SESSION     9  LGON 04 NO SUCH USER
                  UNABLE TO COMPLETE LOG ON
```

If the password provided by the user is incorrect, the following error message will be returned to the user's terminal:

```
   SESSION     9  LGON 05 ILLEGAL ACCESS.
                  UNABLE TO COMPLETE LOG ON
```

The user can try to log on again by repeating the process. After a second unsuccessful attempt the user should contact his System Manager for assistance.

After a successful log on, the system will send a message to the session terminal containing the session identification number, the time of day that the session started, and the user's cumulative connect time to date. An example would be:

```
    SESSION    9 ON  9:54 AM  THU., 15  FEB., 1980

    PREVIOUS TOTAL SESSION TIME: 00 HRS., 40 MIN., 19 SEC.
```

A line containing the session identification number, the time of day that the session started, and the user's name is also sent to the system console:

```
SESSION    9 ON  9:54 AM  THU., 15 BROWN.MANUF
```

The following messages may then be issued to the session terminal:

a. A system message file defined by the System Manager. For example:

```
THE SYSTEM WILL BE DOWN FOR MAINTENANCE AT 10:00 PM

TONIGHT. PLEASE LOG OFF BEFORE THEN.
```

b. "MESSAGES WAITING" if there is any text in the user's message file. For more information on the Message File, refer to the report and clear messages command (ME) and the send message command (SM) in Chapter 3 of this manual.

The user's copy of FMGR is then scheduled and the following services are performed by LOGON:

a. The user HELLO file is transferred to if it has been defined for the user. This file can be unique to a single user or can be shared among several users. The HELLO file is a FMGR procedure file which is transferred to when the user logs on. It allows the System Manager to tailor the Session Monitor environment to individual users and their applications.

b. The default log and list devices are set to the user's terminal (session LU 1).

c. The severity code is set to 0, which will cause all error codes to be displayed and command to be echoed on the log device (refer to the SV command in Chapter 3 for a description of severity codes).

Log-Off Procedure

The LGOFF program is responsible for severing the user's connection with the system and returning to the system all temporary resources allocated to the session. This includes:

a. Terminating any active session programs,

b. Releasing main program ID segments assigned to temporary session programs,

c. Dismounting private and/or group disc cartridges (optional),

d. Releasing all System Available Memory (SAM) or disc tracks allocated to any temporary session programs,

e. Updating all accounting information stored in the session Account File such as total CPU and connect time.

The log-off procedure is initiated by entering the following command:

```
+-------------------------------------------------------+
|                                                       |
|                  ,SP                                  |
|           :EX <      > [,RG [,KI ]]                   |
|                  ,RP                                  |
|-------------------------------------------------------|
|                                                       |
|  SP = save private cartridges (see note below).       |
|                                                       |
|  RP = release private cartridges (see note below).    |
|                                                       |
|  RG = Release  group  cartridges;  default is to not  |
|       release group cartridges.                       |
|                                                       |
|  KI = Abort any active session programs.              |
|                                                       |
|                                                       |
|                     *** NOTE ***                      |
|                                                       |
|         The specification of either SP or RP          |
|         is only required  if the user  has a          |
|         private cartridge currently mounted.          |
|                                                       |
+-------------------------------------------------------+
```

Before a user can be logged off, all active session programs must be terminated. The following message is issued if any session programs are active:

```
PROG1
PROG2
ABOVE SESSION PROGRAMS ACTIVE
OK TO ABORT ? (Y OR N)
```

If "N" is entered, the log-off procedure is terminated (the user is not logged off). If a "Y" is entered, the active programs (PROG1 and PROG2) will be aborted, and the log-off sequence continued.

If the kill option (KI) is specified, the above message will not be issued and all active programs will be aborted.

The LGOFF program then updates the session's user and group accounting information by storing the log-off time, cumulative connect time, and CPU usage in the Account File.

An example log-off message issued to the session terminal is:

```
SESSION    9  OFF 10:09 AM  THU., 15  FEB., 1980
CONNECT TIME:              00 HRS.,  04 MIN.,  20 SEC.
CPU USAGE:                 00 HRS.,  00 MIN.,  19 SEC.,   40 MS.
CUMULATIVE CONNECT TIME:   06 HRS.,  04 MIN.,  03 SEC.
END OF SESSION
```

The first line of the message is also sent to the system console.

Command Capability Levels

Command capability levels define which set of interactive operator commands can be executed by the user. A user has access to all commands at or below his capability level. For example, a user with capability level 30 has access to all commands available at that level, as well as those of the lower capability levels of 10 and 20.

A user's command capability level is defined by the System Manager and stored in the Account File. As a user's needs change, his command capability level can be altered. Table 2-1 shows the command capability level assignments provided by Session Monitor for FMGR and System interactive commands.

Table 2-1. Command Capability Level Assignments

FMGR	SYSTEM and BREAK MODE	CAPABILITY LEVEL
EX SY TR HE	HE OP	1
AC MC TE CL ME WH DC *SL ?? DL SM LI	$BL RS *TO +BR *SL UP *EQ ST WH FL TE $QU TI	10
AN DP RN CN DU ST CO LL SV CR PK CT PU		20
CS RT SL +OF RU SP RP	+GO RT +SS +OF RU SZ +WS +VS	30
CA PA SE IF		40
LO SL	AS ON UR IT PR	50
IN VL OF CU	*AG GO TM BL LU TO BR OF UL EQ SS	60

* Single parameter only

+ Program must be under session's control

$ No parameters permitted

Cartridge Ownership

There are four categories of cartridges in the session environment:

1. System cartridges,

2. Private cartridges,

3. Group cartridges, and

4. Non-session cartridges.

System cartridges can be further broken down into two categories: those available to all users with unrestricted access (system global cartridges), and those available to all users with restricted access (LU 2 and 3). Cartridges with system logical units (LUs) 2 and 3 are primarily read-only, while system global cartridges have both read and write access. System LU 2 is where the disc-resident operating system resides. The remaining tracks on LU 2 and LU 3 are where stored programs, user message files, system manager defined procedure files, and other system manager files, reside.

A private cartridge can be accessed solely by the user who has it mounted to his session, and by the System Manager who has access to all cartridges in the system.

A group cartridge is only accessible by members of the group who have it mounted to their sessions and to the System Manager.

Non-session cartridges are only accessible to non-session users and the System Manager. Since the system console has the capability of operating in both the session and non-session environment, non-session cartridges can be accessed through the system console.

Session-Related Data Structures

There are four session-related tables which the user should be familiar with. These are:

1. Account File
2. Session Control Block (SCB)
3. Session Switch Table (SST)
4. Configuration Table

The Account File is a disc-resident file created and maintained through the account setup program (ACCTS) by the System Manager. It contains the user, group and password names for all users under session control, their respective command capability levels, group and user Session Switch Tables (SSTs), and their cumulative connect and CPU times. Also contained are group cumulative connect and CPU times, and the LUs of cartridges in the spare disc cartridge pool.

A user's Session Control Block (SCB) is created by the LOGON program and stored in System Available Memory (SAM) after the user has successfully logged on. It contains the user's command capability level, user and group name, CPU usage, SST, and a list of cartridges mounted to the user.

The Session Switch Table (SST), located in the SCB, is built up from the user and group SSTs associated with the user, and from the Configuration Table, all of which are taken from the Account File. Each entry in the SST describes a session LU which the user addresses and an associated system LU where the I/O request is actually directed. Only those devices entered into the user's SST can be accessed by him or her. The group SST defines those devices available to all members of a specific group, and the user's SST defines those additional devices accessible to the user.

The Configuration Table is similar to the group and user SSTs in that each entry describes a session LU which the user addresses and an associated system LU where the I/O request is actually directed. However, the Configuration Table relates session and system LUs only for those devices associated with a specific terminal (station). Examples are the right and left cartridge tape units on a HP 2645A/48A terminal or a terminal auxiliary printer.

System Console Operation

The system console can operate in either a session or non-session mode when Session Monitor software is generated into the system.

Immediately following boot-up, the system console will be in non-session mode. The user must strike a key to get the system's attention. The system will prompt the user by issuing an asterisk (*) to the terminal CRT.

EN (Enable System Console)

In order for the system console to operate under session, the terminal must be enabled as a session terminal using the EN command.

```
+-------------------------------------------------------------+
|                                                             |
|   *EN,security code [,option]                               |
|                                                             |
|-------------------------------------------------------------|
|                                                             |
|  security code = the FMP master security code specified in the |
|                  File Manager IN command (refer to Chapter 3). |
|                                                             |
|  option        = 0 or 1.                                    |
|                                                             |
|                  0 = master security code not required in "OP" |
|                      commands (default).                    |
|                                                             |
|                  1 = master security code  is required in "OP" |
|                      commands.                              |
|                                                             |
+-------------------------------------------------------------+
```

Once this command has been entered, the system console is enabled as a session terminal. To reset the system console to operate non-session, the user can reset it through the ACCTS program or re-boot the system (refer to the RTE-6/VM System Manager's Manual for more details).

Once logged on, all break-mode and FMGR requests are checked for session capability restrictions. Any attempt to execute a command requiring a capability greater than the user possesses will result in the error message "CAPABILITY ERROR".

OP (Allow System Level Operator Command)

The "OP" command provides a method of entering a single system level command from a low capability session. This may be required to correct a system level problem while LU 1 is being used as a session terminal. Unauthorized use of this command can be prevented by requiring knowledge of the master security code when session mode is enabled.

```
+----------------------------------------------------------------+
|                                                                |
| S=01 COMMAND ?OP [,security code [,command ]]                  |
|                                                                |
|----------------------------------------------------------------|
|                                                                |
|     S=01 COMMAND ? is the break-mode prompt.                   |
|                                                                |
|  security code  is the FMP master security code defined by the |
|                 File Manager IN command  (refer to Chapter 3). |
|                 If specified in the EN command,  the  security |
|                 code is required.                              |
|                                                                |
|  command        is the system command to be executed.          |
|                                                                |
+----------------------------------------------------------------+
```

AB (Abort Executing Batch Job)

The AB command which aborts the currently executing batch job is only valid when entered at the system console. Refer to the RTE-6/VM Batch and Spooling Reference Manual for details on batch job processing.

```
+-------------------------------------------------------------------+
|                                                                   |
|                ,0                                                 |
|           AB <   >                                               |
|                ,1                                                |
|                                                                   |
|-------------------------------------------------------------------|
|                                                                   |
|  0   is the default case.  It terminates the current batch job    |
|      immediately  if it is executing,  scheduled  or  operator    |
|      suspended.  If the job is I/O, memory or disc-suspended,     |
|      the batch job is terminated the next time it is scheduled.|
|      Disc tracks are not released.                                |
|                                                                   |
|  1   immediately  terminates  the batch job  and  releases all    |
|      disc tracks.  If suspended  for I/O,  a system generated     |
|      CLEAR request is issued to the driver.                       |
|                                                                   |
+-------------------------------------------------------------------+
```

The AB command is only valid when input from the following modes:

```
+-------------------------------------------------------------------+
|                                                                   |
| System Console in non-session mode:                               |
|                                                                   |
|    *AB                                                            |
|                                                                   |
|-------------------------------------------------------------------|
|                                                                   |
| System Console in session mode:                                   |
|                                                                   |
|    S=01 COMMAND ?OP,master security code,AB                       |
|                                                                   |
+-------------------------------------------------------------------+
```

Non-Session Terminals in Session Environment

When Session Monitor is generated into the system, any non-session terminal (other than the system console) will be intended for running dedicated application programs. These programs will normally be scheduled from the system console to run at the non-session dedicated terminal or can be set up to automatically run by the operator causing an asynchronous interrupt on the terminal.

An example of a dedicated application would be data entry functions such as processing of sales orders or material purchases.

Because of the dedicated applications on these terminals, the user will not be able to input any interactive operator commands.

Operating Without Session Monitor

If the user does not desire the features provided by Session Monitor in his system, he may include the Multi-Terminal Monitor (MTM) instead. The following section assumes that MTM is included in the user's operating system and describes interactive operation under the MTM environment.

Multi-Terminal Monitor (MTM)

When operating under MTM, the user can input any interactive operator command, which is not solely related to Session Monitor regardless of the command capability level. Chapters 3 and 4 of this manual describe operator commands available to the user under MTM and Session Monitor.

MTM will perform several services for the user in conjunction with a terminal's copy of FMGR. A terminal with logical unit number xx has its own copy of FMGR if the program FMGxx exists. For example, the copy of FMGR for LU 9 would be FMG09.

If a copy of FMGR named FMGxx does not exist for a terminal (whose LU is xx), the standard MTM prompt (xx>) will be issued and the user will be conversing with the RTE operating system. The remainder of this section assumes that the terminal has its own copy of FMGR named FMGxx.

Available MTM Services

In an MTM environment, a user terminal with its own copy of FMGR has access to three services:

1. Automatic scheduling of FMGxx when the user terminal interrupts the operating system.

2. Automatic renaming of user programs scheduled from FMGxx.

3. Automatic execution of transfer file named HI.

If a copy of FMGR called FMGxx exists for the terminal, striking a key on the terminal causes FMGxx to be scheduled for execution.

If the terminal's copy of FMGR is available for execution (not busy or suspended), two events will occur:

First, the prompt

```
    xx>FMGxx
```

will be issued to the terminal.

Second, control is transferred to a FMGR procedure file named HI, which must exist on LU 2, the system disc.

The HI file is usually written by the System Manager. Although the file may be empty, it must nevertheless exist or an FMGR -006 error will result. When the end of the HI file is reached, control is transferred to the interrupting terminal.

AB (Abort Executing Batch Job)

The format of the AB command for aborting the currently executing batch job is the same as with session (refer to the AB command description in the OPERATING WITH THE SESSION MONITOR section in this manual). The command must be input from the system console to be valid. For details on batch job processing, refer to the RTE-6/VM Batch and Spooling Reference Manual.

Interactive Operator Commands

Interactive operator commands are the means by which the user in a session or MTM environment communicates with the operating system. The commands provided with the RTE-6/VM system allow the user to create and edit files; in conjunction with the compiler and loader utilities, develop programs and schedule programs.

File Manager (FMGR) Commands

Functions provided by the FMGR commands include:

- Create files
- List file contents
- Send messages
- Request error messages
- Mount cartridges
- Dismount cartridges
- List mounted cartridges
- List file directories of mounted cartridges
- Run programs
- Terminate programs

The FMGR prompt is a colon (:). Chapter 3 describes all the FMGR commands.

RTE System Commands

Functions provided by the RTE system commands include:

- Schedule programs to run at a specific time
- Schedule programs to run immediately
- Abort a program
- Suspend a program
- Reschedule a suspended program
- Change a program's priority
- Assign a program to a partition
- Check the status of program
- Check the status of a device
- Down a device
- Up a device
- Set the real time clock
- Display the time
- Set the timeslice quantum and fence

All interactive RTE system commands can be entered in two ways:

1. On the system console (in the non-session mode) or an MTM terminal in the operating mode. System commands solely related to Session Monitor cannot be used in a MTM environment. Operator mode prompt is the asterisk (*) on the system console, (LU) on an MTM terminal.

2. On any terminal in the FMGR mode. The command must then be preceded by SY. The FMGR prompt is a colon (:). An example would be:

   ```
   :SYTI
   ```

   where TI is the RTE system command to display the time.

Most of the RTE system commands can also be input during break-mode operation.

Chapter 4 of this manual describes all the RTE system and break-mode commands.

RTE Break-Mode Commands

Break-mode is reached by interrupting the system (by striking any key) while the system does not have a read request pending from the terminal (i.e., no prompt is issued to the CRT). An example of break-mode operation would be to interrupt the system during a list process to the terminal CRT. After the interrupt is recognized by RTE, the break-mode prompt:

```
S=xx COMMAND ?        (for a Session terminal)

        or

LU>                   (for an MTM terminal)
```

where xx is the session identification number and LU is the logical unit number of the MTM terminal, is issued to the terminal CRT. At this point, any break-mode command can be issued. Also, many of the RTE system commands can be issued (refer to Chapter 4 for more details).

For instance, if the session identification number is 15 and the operator wishes to terminate a listing on his CRT, the user could issue the BR command in response to the break-mode prompt. For example:

```
S=15 COMMAND ?BR
```

If using a MTM terminal, defined as logical unit 15, the operator can terminate a listing on his CRT by issuing the command:

```
15>BR
```

Interactive Editor Commands

The interactive editor commands are used for editing files. The user can edit files either by running the program EDIT (or a copy of EDIT) or by transferring to a procedure file which runs EDIT (or a copy of EDIT). Some of the features of the RTE Editor include:

* File creation capability.
* User defined tab stops.
* Merging of files.
* Selectively displaying file contents by one or more lines at a time.
* Editing of files using line edits, character edits, or pattern edits.
* Screen Mode editing
* Line range options
* Program scheduling capability
* Finding and editing of text patterns.
* Capability of moving blocks of text.
* Automatic line sequencing in columns 73 through 80.
* User defined windows to facilitate pattern searching and editing.
* Time stamping of files

The EDIT prompt is a slash (/). The EDIT/1000 User's Guide contains a description of the interactive editor commands.

Chapter 3
File Manager Commands

Introduction

File management is performed through program calls to the File Management Package (FMP) library and by interactive operator commands to the program File Manager (FMGR). The main function of FMP calls is to control input to and output from disc files or peripheral devices treated as files. File management capability is increased by using FMGR for interactive program development, disc cartridge manipulation, and batch job control.

This Chapter describes the available FMGR operator commands. For more details on FMP program calls, refer to the RTE-6/VM Programmer's Reference Manual. For more details on batch job control, refer to the RTE-6/VM Batch and Spooling Reference Manual.

Certain terms used in discussing file organization for file management are defined in Table 3-1. The File Manager Commands are summarized in Table 3-2. Note that the table categorizes the commands by function. Their descriptions in this chapter are organized alphabetically.

Table 3-1. File Management Terms

DISC	A rotating random access storage device on which files may be stored and from which they may be retrieved. Discs may be fixed or removable.
CARTRIDGE	A cartridge is a set of contiguous cylinders on a disc unit. Cartridges contain disc files with a directory of the files stored on each cartridge. All files on the same FMP cartridge must have unique names. The system disc on logical unit 2 (LU 2) contains the RTE operating system, and may contain FMP files.
CYLINDER	A cylinder is the area that passes under all disc heads during one revolution of the disc surfaces.
TRACK	A subdivision of the disc. A physical track is the area that passes under one head during one revolution of the disc surfaces. The File Management System handles all tracks as logical tracks. For MAC and ICD discs a logical track is identical to a physical track. For CS80 discs, a logical track is a set of contiguous blocks which may or may not align with a physical track.
SECTOR	A further subdivision of the disc. A physical sector is a portion of a disc track consisting of 64 or 128 words depending on the type of disc. The File Management System treats all sectors as 64 words.These sectors are referred to as logical sectors.Two logical sectors make up one FMP block, the unit of information that may be physically transferred between the disc and the memory.
FILE	A disc file is a collection of records terminated by an end-of-file mark. Non-disc devices are treated by the File Management System as if the device were itself a file. The device, like a file, is a collection of records terminated by an end-of-file mark that depends on the device. Any file can have zero or more records and is designated by a name of six characters or less.

Table 3-1. File Management Terms (continued)

NON-DISC DEVICE	A peripheral device, not a disc, that may be treated as a file and controlled by the File Management Package. Non-disc devices include the line printer for output, magnetic tape and mini-cartridges for input or output, and terminals through which an operator can interact with the system.
LOGICAL UNIT	An integer assigned to each input-output device or disc cartridge at system generation by which the cartridge or device can be referenced.
BLOCK	A subdivision of a disc file containing 128 words that is the smallest unit that can be physically transferred between disc and memory during file access.
RECORD	A logical collection of 16-bit words in a file or device. A record may have zero or more words. A record is the smallest unit that may be accessed by an FMP call from a program.
EXTENT	An extension to a file automatically provided by FMP as needed. Each extent is the same length and type as the file and is identified by a positive integer called the extent number.
DATA CONTROL BLOCK	FMP uses an array called the Data Control Block (DCB) as an interface between FMP and the user's program. It contains control information for the file and also serves as a buffer for the transfer of data between a file and the user's program. Space must be allocated for the Data Control Block in any program making an FMP file access call. This is often referred to as a packing buffer.
USER BUFFER	An area in the calling program to hold data during file access.
ID SEGMENT	A block of words, associated with each program, that is used by the system to keep track of the program's name, priority, current scheduling status, and other characteristics. Every program must have it own ID segment.

Table 3-2. Summary of File Manager Commands

INFORMATIONAL COMMANDS

COMMAND	CAPABILITY LEVEL	DESCRIPTION
??	10	Request explanation of FMGR error code.
HE	1	Request error explanation and possible corrective action.
WH	10	Display current system status.

FILE AND DEVICE MANIPULATION

COMMAND	CAPABILITY LEVEL	DESCRIPTION
CN	20	Control non-disc device.
CO	20	Copy files from one cartridge to another.
CR	20	Create disc file.
CR	20	Create non-disc file.
CT	20	Control terminal.
DU	20	Transfer data to existing file.
LI	10	List file contents.
LL	20	Change list device.
LO	50	Change log device.
PU	20	Purge file and its extents from system.
RN	20	Rename file.
SP	30	Save program as disc file.
ST	20	Transfer data and create file.

Table 3-2. Summary of File Manager Commands (continued)

FMP CARTRIDGE MANIPULATION

COMMAND	CAPABILITY LEVEL	DESCRIPTION
AC	10	Allocate cartridge from spare cartridge pool.
CL	10	List mounted cartridges.
CO	20	Copy files from one cartridge to another.
DC	10	Dismount cartridge.
DL	10	List contents of file directory.
IN	60	Initialize cartridge.
MC	10	Mount cartridge.
PK	20	Pack one or all mounted cartridges.

PROCEDURE FILE COMMANDS

COMMAND	CAPABILITY LEVEL	DESCRIPTION
**	10	Declare a line of comments.
CA	40	Calculate globals.
DP	20	Display parameters.
IF	40	Conditional skip.
PA	40	Pause and send message.
SE	40	Clear or set globals parameters.
SV	20	Change severity code.
TR	1	Transfer control to a file or logical unit.

Table 3-2. Summary of File Manager Commands (continued)

PROGRAM AND RESOURCE CONTROL

COMMAND	CAPABILITY LEVEL	DESCRIPTION
OF	30/60	Terminate Program.
RP	30	Restore program and assign ID segment or release ID segment.
RT	30	Release disc tracks assigned to a program.
RU	30	Run program.
SP	30	Save program as disc file.

MESSAGE RELATED COMMANDS

COMMAND	CAPABILITY LEVEL	DESCRIPTION
AN	20	Send message to list device.
ME	10	Display contents of message file.
SM	10	Send message to session user.
TE	10	Send message to system console.

MISCELLANEOUS COMMANDS

COMMAND	CAPABILITY LEVEL	DESCRIPTION
CS	30	Modify spool options set up by SL command.
EX	1	Terminate session or FMGR.
SL	10	Display session LU information
SL	30	Set-up Spool File for I/O Device.
SL	30/50	Modify Session Switch Table.
SY	1	Execute system command from FMGR.

Files

A file is a collection of information logically organized into records. The information in files may be programs or the data used by programs. Data may be in binary or ASCII code. Programs may be in the form of ASCII source code, or binary code in either relocatable or absolute form. Programs may also be in memory-image form, a form used by RTE for programs ready to be executed.

Files can be stored on disc or they may reference non-disc peripheral devices. FMGR can be used to control and access disc files and non-disc files.

Types of Files

Eight file types are defined by the file system. Additional types may be defined by the user. Only the first four types differ in format; all subsequent types differ only in the type of data the file system expects the file to contain. The file types may be divided into three categories as shown in Table 3-3. The first category contains one type, type zero. This type includes all non-disc devices defined as files and accessible by name. The second category contains two file types, types 1 and 2. These fixed-length record files are used for quick random access. The remaining file types all belong to the third category of files with variable-length records designed for sequential access. All files can be extended automatically as needed. Disc file record formats are illustrated in the RTE-6/VM Programmer's Reference Manual.

Table 3-3. Categories of File Types

CATEGORY	TYPE	DESCRIPTION
Control	0	Non-disc files
Fixed-length, random access	1	Fixed-length 128 word record files
	2	Fixed user-defined record length files
Variable length, sequential access	3	Variable-length records; any data type
	4	Source program file; ASCII
	5	Object program file; relocatable binary
	6	Executable program file; memory-image code
	7	Absolute binary
	8-32767	User-defined data format

Type 0 Files

Type 0 files are used to reference non-disc devices by name. They afford a measure of device independence in that the standard file commands can be used to control the device. A directory entry is made for the device as if it were a file. A type 0 file is created and purged with a FMGR command, not with an FMP call. The File Directory entry for a file of this type contains special entries that specify logical unit number (LU) and the operations allowed on the particular device.

Type 1 Files

Type 1 files have fixed length records of 128 words and are extendable. Because the File Management Package transfers data to and from disc in 128-word blocks, this file type allows direct access between disc and the user's buffer area in his program, thereby eliminating the need to go through a packing buffer (the Data Control Block). As a result, type 1 files have the fastest transfer rate. Any disc file (not type 0), may be opened and accessed as a type 1 file in order to take advantage of the faster transfer rate. However, if the file being transferred has less than 128-word logical records, the user must be able to recognize where the records begin and end within the 128-words. If the records are longer, the user must be able to work with part of a record at a time. The end-of-file mark is the last word of the last block allocated to the file.

Type 2 Files

The record lengths of type 2 files are fixed, but the length is defined by the user at file creation. Type 2 files are also extendable. Like type 1 files, the end-of-file is the last word of the last block allocated to the file. Only one logical record is transferred at a time, but unlike type 1 files, the transfer must go through a packing buffer (the Data Control Block). For this reason, type 2 files have a slower transfer rate than type 1 files.

Type 3 Files

These files have variable length records, are extendable, and may contain data, source code, relocatable or absolute binary code. Only one logical record is transferred at a time and the transfer must be made through the packing buffer (Data Control Block). The first and last words of each record as written on disc always contain the number of words in the record (minus the two length words). A zero-length record, consisting of two zero words, can be used to separate groups of records into sub-files. The end-of-file is marked by a -1 as the first length word in the next record. Words following the end-of-file are undefined. However, FMP can write records beyond the end-of-file by replacing the end-of-file with a new record followed by an end-of-file mark.

Type 4 Files

This file type is the same as type 3, except the file system expects these files to contain ASCII data. Typically, source program files are type 4.

Type 5 Files

This file type is the same as type 3, except the file system assumes type 5 files contain relocatable binary code. Typically compiler object program files are type 5. The FMGR expects relocatable binary code in these files and always checks the type of code when transferring data to or from these files.

Type 6 Files

This type file is the same as a type 3 file, except the system assumes it contains a program in memory-image format that is ready to run. Type 6 files are created by the Save Program (SP) command which can copy a temporary program or segment, created by the LOADR or MLLDR and stored on the system scratch tracks, or a permanently loaded program or segment into a type 6 file on the FMP tracks of any disc cartridge. These files are always accessed by the File Management Package as type 1 files. The first two sectors of a type 6 file are used to record ID segment information for the program. Thus, this file type can be used for programs that do not have a permanent ID segment.

Type 7 Files

This file type is the same as type 3, except the system expects type 7 files to contain absolute binary code.

Type 8 and Greater Files

This file type is the same as type 3, but the content is user-defined. FMP does no special processing based on file type for types greater than 7. For example, any checksum operations must be specifically requested. Content is also user-defined; it may be source, relocatable binary, memory-image format, etc.

File Access

Type 1 and type 2 files contain fixed-length records. This makes it possible to calculate the position of a desired record (random access), and thereby "randomly access" components of the file. On the other hand, type 3 files and above contain variable length records, so the system must access the disc at least once, and in some cases several times, in order to position to the desired record location. For this reason, random access takes longer for file types greater than type 2.

File Extents

All disc files can be automatically extended whenever a write request points to a location beyond the range of the currently allocated file. The extent is created by FMP with the same name and size as the main file, and the write request continues. FMP numbers each extent starting with 1. The extent number and location is kept in the file directory entry for the file extent. When a file with extents is referred to by its file name, any extents are provided automatically. A file may have a maximum of 255 extents. Refer to the RTE-6/VM Programmer's Reference Manual for a description of file extents.

Cartridges

Files managed by the File Management Package, whether they are program files, data files, or spool files, are kept on FMP disc cartridges. An FMP disc cartridge is a logical entity that may correspond directly to a set of contiguous disc cylinders on a disc platter or may be a subdivision of the disc platter. On some discs, a cartridge may even cross platter boundaries. Care must be taken if the cartridge is partly on a removable platter and partly on a fixed platter. Refer to the RTE-6/VM System Manager's Manual for further details.

Each cartridge is defined as a contiguous block of tracks. Each is assigned a logical unit number (LU) and a Cartridge Reference Number (CRN); either of which may be used to reference the cartridge. Files on the same cartridge must have unique names. Duplicate names may be used as long as the duplicates are on separate cartridges so each file can be uniquely identified by its filename and cartridge identifier.

Locations of Files on Disc Cartridges

Files are stored in a consecutive fashion on disc cartridges. Generally, files are located contiguously, but if the file has extents, the extents may not be contiguous. For example:

```
+----------------------------------------------------------------+
|                                                                |
|           +------------------------------------------------+   |
| First --> |       |        |          |        |           |   |
| Track     |File A | File B |  File A  | File C |  File A   |   |
|           |       |        |1st extent|        |2nd extent |   |
|           +------------------------------------------------+   |
|           ^       ^        ^          ^        ^           ^   |
|           |       |        |          |        |           |   |
| Sector:   0       8       12         20       24          32   |
|                                                                |
+----------------------------------------------------------------+
```

Note that File A and its extents are not contiguous on the disc. As files are stored on disc, file directory entries are created in the file directory of the cartridge. The file directory starts on the last track of the cartridge.

Removable cartridges containing FMP files are interchangeable between drives of the same type within a system, or between drives on different systems provided that logical track 0 refers to the same physical track on every disc unit. (Refer to Figure 3-1 for an illustration of disc organization using one cartridge on the system disc starting at the first FMP track, and one on a peripheral disc starting at track 0.)

At cartridge initialization, the number of directory tracks for that cartridge is specified. The first track must be assigned at initialization; the number of sectors per track may be specified at this time, but is supplied by FMP as a default if not.

Files may cross track boundaries, but in a multi-cartridge environment, no one file may cross cartridge boundaries. Files are subject to being moved whenever a cartridge is packed. This causes files to be relocated within a cartridge and no absolute file addresses should be kept in any file or program.

Files always start on even sector boundaries and all accesses are multiples of 128 words addressed to even sectors.

Disc errors are passed back to the user for corrective action. Error codes are printed on the system log device when using the FMGR operator commands, or passed to the user program when calling a File Management Package library routine. You may report bad tracks to the system through the FMGR Initialization command. Bad tracks discovered by the system result in an error returned to the calling program.

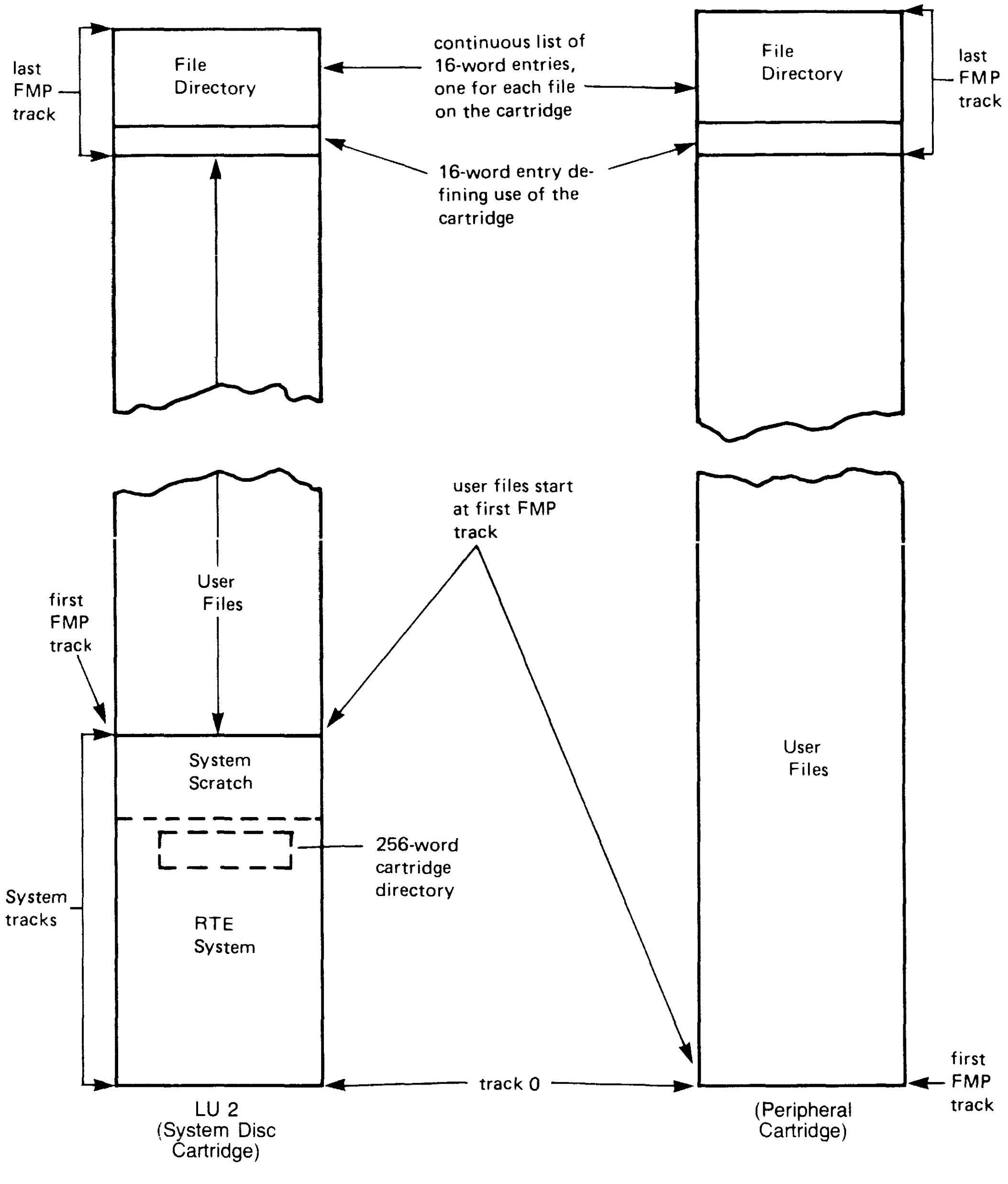


Figure 3-1. Disc Cartridge Organization

Directories

Two types of directories are maintained by FMP: the FMP cartridge directory on the system disc, and the file directory on each cartridge.

Cartridge Directory

The cartridge directory is a master index to all currently mounted FMP cartridges. It is maintained in the system area of LU 2 (refer to Figure 3-1). Its length is two blocks and it has an entry for each currently mounted cartridge. The directory has room to describe up to 63 cartridges using four words for each. Appendix C contains the format for the cartridge directory.

File Directory

A file directory, maintained by FMP for each cartridge, contains information on each file on that particular cartridge. Each directory starts in sector 0 of the last track available to FMP. The first 16-word entry in this directory contains label and track information for the cartridge itself. Each subsequent 16-word entry has information on a user file. The last entry is followed by a zero word. When a file is purged, the directory entry for the file is flagged as unused. When the cartridge is packed, the directory entry for any purged file is cleared and the cartridge area where the file was located is overwritten by non-purged files wherever possible.

Security

Besides the security features provided by Session Monitor, FMP provides two levels of security.

File Security

Every file has a security code. This one word code may be zero, positive, or negative. A security code of zero allows the file to be opened to any user or program that can access the disc cartridge containing that file. This code provides zero security. A positive security code, an integer or two ASCII characters, restricts writing on the file but not reading; that is, a user or program that does not supply the code can read the file, but cannot write on the file. A negative code restricts all access to the file; this code must be specified in order to open a file protected by it. An attempt to open a file so protected without the correct security code results in an error message.

System Security

During FMP initialization a master security code is assigned to the file system. If the master security code is zero, no security is provided by FMP. If the code is non-zero, it must be supplied with commands used to obtain a directory listing that includes specific file security codes.

Types of Cartridges

There are four categories of cartridges in the session environment:

1. System cartridges.
2. Private cartridges.
3. Group cartridges.
4. Non-session cartridges.

When a disc cartridge is mounted to the system, a label is entered in the cartridge list maintained by the system. This label identifies the category of the cartridge and to whom the cartridge was mounted.

System Cartridges

A system cartridge can be mounted or dismounted only by the System Manager. The cartridge where the system tracks reside must be defined during system generation as LU 2. An auxiliary system cartridge, can also be defined as LU 3 during system generation.

LU 2 and LU 3 are generally read-only cartridges for session users. Only the System Manager and non-session users can write on these cartridges. LU 2 contains:

1. Operating System
2. System Library
3. System Scratch Tracks
4. FMP Tracks
5. Cartridge Directory
6. Logical Source Area

LU 3 can be defined if more system scratch tracks are needed or to hold files to be protected from session users.

In addition to LU 2 and LU 3, the System Manager can define system cartridges that provide both read and write access to all users on the system.

Private Cartridges

A private cartridge can be accessed only by the user who mounts it to his session, and by the System Manager, who has access to all cartridges in the system.

Group Cartridges

A group cartridge can be accessed only by members of the group to whom the cartridge is mounted, and by the System Manager.

Non-Session Cartridges

A non-session cartridge can be accessed only by a non-session user and by the System Manager. A non-session cartridge can be accessed through the system console (when it is not enabled under session) or by programs running in a non-session environment.

Session Cartridge Allocation and Access

A session user may interactively request disc cartridge allocation from the system through either the File Manager MC (Mount Cartridge) command or AC (Allocate Cartridge) command. Three basic differences between these commands are:

1. MC requests that a specific cartridge, identified by an unique LU, be mounted. AC requests that the first available cartridge in the spare cartridge pool meeting the user's specified size requirement be mounted. When using the AC command, it is unnecessary for the session user to know the LU associated with a specific cartridge.

2. MC requires that a cartridge entry exists in the user's Session Switch Table (SST) prior to the request, AC does not. AC will make the appropriate SST entry after a free cartridge has been allocated from the spare cartridge pool.

3. MC allows the user to mount any cartridge defined during system generation subject only to availability and SST constraints. AC only allows users to mount cartridges from the spare cartridge pool.

The spare cartridge pool can be thought of as a source of temporary disc work space available to the session user. When the user mounts a pool cartridge with the AC command, the file directory on the cartridge is automatically cleared for the user. By using the disc cartridge save and restore utilities, WRITT and READT, in conjunction with the spare cartridge pool, the user can make efficient use of disc space.

Other cartridges defined during system generation, not contained in the spare cartridge pool, are usually reserved for longer duration usage. Examples are system cartridges and cartridges assigned to specific users or groups by the System Manager.

Session Monitor provides the session user with protection for his mounted cartridges by restricting cartridge access to only those cartridges mounted in the user's Session Control Block (SCB). When a cartridge is specified in a call, FMP checks that the cartridge is mounted not only to the system, but is also mounted to the user's SCB.

There are some exceptions to the above check:

1. The System Manager is allowed access to all cartridges mounted in the system.

2. Some internal subsystems (for example, Spooling) have access to all cartridges mounted in the system.

3. With a few exceptions, write access is not allowed on LU 2 and LU 3, but read and write access will be allowed on all other system global cartridges when operating under session.

4. User message files residing on LU 2 or LU 3 are available for session users to open, read, and write into via the ME and SM commands described later in this Chapter.

5. Type 6 files (programs) may be created and purged by any user on LU 2 and LU 3 via the SP and PU commands described later in this chapter.

6. Procedure files set up by the System Manager on LU 2 or LU 3 can be run by any session user. The commands in this file are not subject to normal cartridge or command capability level restrictions. Through these commands the user may be given read and write access to LUs 2 and 3 in addition to his own private and group cartridges.

If a particular cartridge is not specified when accessing a file, the user`s private cartridges are searched in the order they were mounted in the system cartridge directory. The user's group cartridges are then searched, and finally, system cartridges are searched. Note that if a cartridge is locked, that cartridge will be skipped when searching for the file.

Non-Session Cartridge Allocation and Access

Without Session Monitor, no cartridge protection is provided to prevent users from accessing all cartridges mounted in the system. In addition, all non-session users will have read and write access to all system cartridges including LU 2 and LU 3.

Logical Unit Numbers

A logical unit (LU) number is an integer value assigned to an input/output device or disc cartridge by which the cartridge or device can be referenced. In the session environment, there are two different types of LUs: system LUs and session LUs. The Session Switch Table (SST) maps a session LU, which the user addresses, to the system LU, where the system processes the request. In the non-session environment, the user addresses the system LU. In either environment the user can directly address only LU numbers between 1 and 63.

System Logical Unit Number

Each I/O device and disc cartridge is assigned a unique system LU during system generation. The system uses these LU numbers to identify each I/O device and disc cartridge for data transfers and other operations. A system LU can be an integer between 1 and 254, and each must be unique.

Session Logical Unit Number

Session LUs need only be unique within a user's session. In conjunction with the Session Switch Table (SST), session LUs allow a session user to access cartridges and devices by LUs different than the system LUs defined at system generation (except for cartridges). Thus, every user can refer to his or her session terminal as LU 1, regardless of what the system LU for that terminal is.

Session LUs can be any integers between 1 and 63. Mappings between session and system LUs can be created and altered in the following ways:

1. At Account File setup.
2. At log on (via configuration table entries).
3. By the operator (via the SL command).

The session LU can differ from the system LUs only for non-disc devices; they must be identical for disc cartridges.

Session Switch Table

The Session Switch Table (SST) provides a mapping from the session LU, which the user addresses, to the system LU, where the system processes the call. Through this mapping scheme, the SST allows session users access to system LUs greater than 63. It also restricts the session user from accessing a device or disc cartridge which is not listed in the users SST.

The SST, placed in the user's Session Control Block (SCB) by the LOGON program, consists of up to 63 system LUs and session LUs. For every session LU, there is a system LU corresponding to the actual device or cartridge. This may be a direct map (session LU 25 and system LU 25), or an indirect map (session LU 1 and system LU 41). A disc cartridge always has a direct map. Below is an example of a user SST:

```
+-------------+
|  41  |   1  |  <---Session terminal definition.
|-------------|
|   2  |   2  |  <---System disc cartridge.
|-------------|
|   3  |   3  |  <---Auxiliary disc cartridge.
|-------------|
|  42  |  42  |  <---"Direct Map"
|-------------|
| 179  |  45  |  <---"Indirect Map"
+-------------+
       |
SYSTEM | SESSION
  LU      LU
```

In the above example, the user has access to system LU 41, 2, 3, 42, and 179. The user would address requests to those LUs by specifying session LU 1, 2, 3, 42 and 45, respectively. In another user's SST, the session LUs could refer to different system LUs.

Global Parameters

Global parameters are variables that can be set, examined, and altered by FMGR commands. Global parameters can replace any FMGR command parameter. When used in procedure files, they are similar to formal parameters in a procedure or subroutine to which actual values are passed through the TR command. There are three ways to access the global parameters: as G globals, as P globals, or S globals. Each global is identified by a variable name that combines an integer with the letter G, P, or S (refer to Table 3-4 for the relationship between G, S, and P globals).

G Globals

There are 11 G globals named 0G through 10G. Globals 1G through 9G can be set or altered with the TR, CA, or SE commands and may be assigned any values: they may be null (no value assigned), have an integer value, or contain up to six ASCII characters. Refer to Table 3-5 for the format of the G globals.

Globals 0G and 10G have particular values: 0G is set to the LU of the input device which scheduled FMGR or for batch job processing, to the job input LU. 10G can be set by a PRTN call in a program executed by the FMGR RU command. Notably, it is set to the name of the most recently loaded program following the command :RU,LOADR or :RU,MLLDR. 10G always has an ASCII value.

P Globals

There are nine P globals, 1P through 9P. The first five P globals can contain integer values returned by a program executed with the RU command via a call to the PRTN routine. Global 6P is the current FMGR error code and 7P is the current severity code. Global 8P is the session identifier, or the terminal LU if not in session. Global 9P is the session user's command capability level, or is 0 if not under session. The contents of 8P and 9P can be displayed with the DP command and tested with the IF command, but cannot be modified with the CA command. The first three P globals correspond to global 10G; that is, 1P is the integer equivalent of the first two characters in 10G, 2P of the second two characters, and 3P of the last two characters. When it is necessary to refer to any of these three values separately, the P globals can be used instead of 10G.

A program can pass back parameter values to globals 1P through 5P by calling PRTN. Other programs can retrieve these values by calling RMPAR. For example:

```
:RU,PROG1 <-------------------PROG1 must call PRTN  to set up 1P
                              through  5P and  return values  to
                              FMGR.

:RU,PROG2,1P,2P,3P,4P,5P <---PROG2 can retrieve  values from 1P
                              through 5P (equivalent to 10G, 4P,
                              and 5P), by calling RMPAR.
```

The P globals are particularly useful when one-word parameters are expected. For example:

```
:RU,XYZ,-27P,-26P,-25P <-------Passes global 3G to program XYZ.

:RU,ABC,1P,2P,3P       <------Passes global 10G to program ABC.
```

S Globals

S globals are set by the commands JO or SL when running under control of the Spool Monitor. They may be referred to by FMGR commands within spooled jobs.

Two S globals have particular meanings: 1S is null or is the ASCII file name of the last created spool file; 0S is null or is the spool LU of the last spool file set up. 0S and 1S are always null when spooling is not being used.

Refer to the RTE-6/VM Batch and Spooling Reference Manual for more details on batch job processing and using the Spooling System.

Table 3-4. Global Equivalence.

S	G	P		
0	-2	-48	Type	
		-47	1	
		-46	2	
		-45	3	
1	-1	-44	Type	
		-43	1	
		-42	2	
		-41	3	
2	0	-40	Type	
		-39	1	
		-38	2	
		-37	3	
3	1	-36	Type	
		-35	1	
		-34	2	
		-33	3	
4	2	-32	Type	
		-31	1	
		-30	2	
		-29	3	
5	3	-28	Type	
		-27	1	
		-26	2	
		-25	3	
6	4	-24	Type	
		-23	1	
		-22	2	
		-21	3	
7	5	-20	Type	
		-19	1	
		-18	2	
		-17	3	
8	6	-16	Type	
		15	1	
		-14	2	
		-13	3	
9	7	-12	Type	
		-11	1	
		-10	2	
		- 9	3	
10	8	- 8	Type	
		- 7	1	
		- 6	2	
		- 5	3	
11	9	- 4	Type	
		- 3	1	
		- 2	2	
		- 1	3	
12	10	0	Type	
		1	1	
		2	2	
		3	3	
13	11	4	4	
		5	5	
		6	6	Last FMGR error
		7	7	Severity code
		8	8	Session identifier
		9	9	User's capability level

The standard values are shown within dark lines.

8100-3

Global Format

All globals are stored in an array in memory in the user's copy of FMGR. G and S globals occupy four words. P globals occupy one word. Table 3-5 illustrates the format of G and S globals.

Table 3-5. G and S Global Format

word 0	global type=0 (null)	1 (numeric)	3 (ASCII)
word 1	0	integer	characters 1,2
word 2	0	0	characters 3,4
word 3	0	0	characters 5,6

Word zero defines the global type. If the type is null, the remaining words are also 0. If the type is numeric, then word one contains the integer value and the last two words are zero. An ASCII global (type 3) can contain up to six ASCII characters. This format applies only to G and S globals. Each P global corresponds to one word of the S and G globals (refer to Table 3-4).

Since the system checks a global access by its position in the array (Table 3-4), a P global can be used to reference one word of an S or G global, and S global can be used to reference a G global, or vice versa.

File Manager Commands

The program FMGR can be run from the system console or from a terminal in a multi-terminal environment. It responds to a set of over 40 commands. Command can be entered directly from the terminal to perform a particular function, or one or more commands may be stored in a file called a "procedure file". Commands can also be included in jobs to be entered from peripheral devices in order to provide batch job control. In addition, FMGR commands which are entered from a terminal are placed into a command stack (residing in the FMGR program). Commands in the command stack can later be displayed, modified and executed (refer to the section on Command Stacking later in this chapter). The program FMGR allows the user to perform the following basic functions:

1. Send messages to the terminal or list device, request error explanations, change the log or list devices or error severity.

2. Create and maintain files, both disc and non-disc, including maintenance of the file directory.

3. Maintain the cartridge directory by keeping track of the disc cartridge on which files are placed.

4. Transfer data or programs between files, creating new files as needed.

5. Establish and transfer to procedure files, and manipulate the global parameters used in these files to receive and return data from other commands or procedures.

6. Control execution of jobs in the batch stream, including assigning a job time limit, switching logical units, and aborting the job.

7. Schedule programs for execution.

8. Interactively set up spooling to and from an I/O device (available only with Session Monitor).

Since the files controlled by FMGR include data files, program files, non-disc devices, procedure files, and batch jobs, FMGR can provide full control over all input to and output from FMP. It uses many of the same FMP library calls available to the user.

Command Structure

Each FMGR command is specified by a mnemonic code consisting of at least two letters to indicate the operation to be performed. Depending on the command, parameters may be entered to further specify the command operation. For example, STORE can be specified, but ST is always sufficient.

The following syntax conventions are used in this manual to specify command parameters.

<table>
<tr><th></th><th></th></tr>
<tr><td>UPPER-CASE BLOCK LETTERS</td><td>Literals that must be specified exactly as shown; underlined characters can be omitted.</td></tr>
<tr><td>lower-case letters</td><td>Describes the type of information to be supplied by the user; most parameters are of this form.</td></tr>
<tr><td>[,parameter]</td><td>Optional parameters are enclosed in brackets; FMGR supplies a default value if omitted.</td></tr>
<tr><td>parameter 1
parameter 2
parameter 3</td><td>One and only one of the stacked parameters can be specified.</td></tr>
<tr><td>+-   -+
| parameter 1 |
| parameter 2 |
| parameter 3 |
+-   -+</td><td>All bracketed parameters are optional; if all are omitted, FMGR supplies default value, or only one may be specified.</td></tr>
<tr><td>[,param1[,param2]]</td><td>Series of optional parameters; the last parameter may be omitted with no indication; embedded parameters must be indicated by a comma when omitted.</td></tr>
<tr><td>...</td><td>Ellipsis indicates that the previous parameter or series of bracketed parameters can be repeated.</td></tr>
</table>

Table 3-6. Command Syntax

Parameter Syntax Rules

FMGR parses commands according to the following syntax rules:

1. The first parameter is separated from the command code by a comma (,) or a colon (:). Subsequent parameters are separated by commas.

2. Commands can be entered as upper or lower case characters.

3. Subparameters are legal in the first two parameters. Also, they are legal anywhere within a privileged command (see below). Subparameters are separated by a colon (:). The first two subparameters may be ASCII or numeric, the rest must be numeric.

4. Blanks on either side of a delimiter or the command code are deleted from the command entry; they are not transmitted or echoed back.

5. Parameters are first assumed to be an integer value, but if the parameter fails to convert, it is treated as ASCII. Exceptions are numbers immediately followed by B, G, P, S, or preceded by a plus (+) or minus (-) sign.

6. Integer parameters observe the following rules:

   a. A leading plus sign (+) is ignored; a number is assumed to be positive unless preceded by a minus sign (-).

   b. A number followed by the letter B is octal.

   c. A number followed by G, P, or S is a global reference.

   d. The range of integer parameters is -32768 through +32767.

7. ASCII parameters are parsed to a maximum of six characters; only the first six characters are interpreted. If the parameter contains fewer than six characters, it is padded with trailing blanks (octal code 40B, the ASCII space). Spaces within a number are ignored. In some commands (AN, TE, SM, CT, HE, or PA), a parameter may contain more than six ASCII characters since the parameter contains a message which is not interpreted.

8. The total number of characters in any parameter must be less than: 128 - (8 times the parameter number) i.e., parameter 10 must be less than 48 characters: 128 - (8*10) = 48.

9. The maximum number of parameters in one command is 14.

10. Comments can be entered following the last parameter as long as they do not replace an omitted optional parameter. They are subject to the length, number, and subparameter restriction on all parameters. Like messages, they are not interpreted.

11. For privileged commands (**, AN, DP, PA, RU, SY, HE, SM, CT and TE), minimal syntax checking occurs before the command is processed. Privileged commands are not subject to command capability level restrictions when executed from a procedure file. The only syntax requirements for privileged commands are:

    a. If issued from a non-interactive device, the first character must be a colon (:).

    b. Global values specified in the command string must be within the legal range (refer to section Global Parameters).

    c. The "constructed" command line length (after globals are replaced and blanks on either side of the delimiter's are removed) must be less than 80 characters.

Namr Parameter

A special parameter, namr, can be used to identify a file or device in a FMGR command. It uses subparameters and can appear only as the first or second parameter.

The format for namr when identifying a file is:

```
file name[:security[:cartridge[:file type[:file size[:record size]]]]]
                                \-----------------v-----------------/
                                          required for file creation
```

The format for namr when identifying a device is:

```
     logical unit number
```

When a namr is used to identify a file only, it is referred to as a file descriptor.

Unless specifically noted, each subparameter has a default value of zero. This value is selected so that, as closely as can be predicted, it provides the most general case. This means that in many cases, all subparameters can be omitted and the file be completely specified by name alone.

Namr Subparameters

filename 6-character ASCII file name; restricted as follows:

* only printable characters, (A-Z),(0-9),!,",#,$,%, &,',(,),=,^,\, ,[,,;,*,],<,>,.,/,?

* plus (+), minus (-), colon (:), or comma (,) not allowed.

* first character must not be a blank (space) or a number.

* embedded blanks not allowed.

* must be unique to FMP cartridge.

logical unit positive integer specifying LU of non-disc device.

security code positive or negative integer or 2 printing ASCII characters. The range is -32767 through 32767. The security code can be:

zero file is unprotected (default).

+integer write-protected. File can be read with any specified security code or none; but can only be written on with correct code or negative value (2's complement) of correct code.

-integer file is fully protected; can be referenced only with correct security code.

cartridge positive or negative integer or 2 printing ASSCI characters; range is from -63 through 32767; used to identify FMP disc cartridge; it may be:

zero first available cartridge that satisfies the request is used (default).

+integer Cartridge Reference Number (CRN) by which the cartridge is identified.

-integer LU associated with the cartridge.

file type positive integer in range 0 through 32767; default depends on command. (Refer to the section Type Of Files.)

file size

decimal number of blocks in range 1 through 16383, or 128-block multiples in range 2 through 32767; a block is 128 words (two 64-word sectors); indicates space allocated to file:

+integer

allocate specified number of blocks to file; minimum is 1.

-integer

allocate specified number of 128 - block multiples to file, integer > 1.

-1

allocate remainder of available space on the cartridge (up to the maximum allowed file size), to the file when creating or storing a file.

record size

decimal number of words in range 1 through 32767; applies only to type 2 files; type 1 files use 128-word records, other types use variable length records.

Examples:

10

logical unit 10

20B

logical unit 16 (octal 20)

$XYZ:AA

file name $XYZ is write protected by ASCII code AA (040501 octal or 16705 decimal).

ABS:-10:-3:2:40:64

file named ABS fully protected by security code -10, is located on LU 3, is a type 2 file 40 blocks long, each record has 64 words.

```
+-------+
| LEVEL |
|  10   |
+-------+
```

?? (Request Error Explanation)

Provides a brief explanation of a FMGR error code.

```
+-------------------------------------------------------------------+
|                                                                   |
| ??[,error#]                                                       |
|                                                                   |
|-------------------------------------------------------------------|
|                                                                   |
| error#  FMGR error code for which explanation is requested; if    |
|         omitted, the explanation of the last error issued is      |
|         given. If two error codes are entered the explanation     |
|         usually refers to the first number only.                  |
|                                                                   |
+-------------------------------------------------------------------+
```

EXAMPLE:

```
   :LI,PROGA       <--------------------List contents of file PROGA.
   FMGR-006        <------------------------------FMGR error message.
   :??             <------------------Request for error explanation.
   FMGR -006 FILE NOT FOUND. <----------------Explanation of error.
```

COMMENTS:

After FMGR assumes control of terminal communication, if a command cannot be interpreted (due to an input error) or has caused a recognized problem, an error message is printed in the form:

```
      FMGR nnn
```

where nnn is a three-digit number. (Refer to Appendix A for a list of all FMGR error codes, their meaning and corrective action.)

In some cases, when an error code is issued, it is followed automatically by additional information. This may consist of the line in which the error occurred, up to the point where the error was detected, or it may be a second FMGR error code.

Any error code explanation can be requested by entering the code number as the error# parameter. A list of all FMGR error codes and their explanations is printed at the list device if error# 99 is entered. Be sure to include the comma separator or the current error will be explained.

For additional error information, the HE command or CMD program can be used. Refer to the FMGR or break-mode HE command for a description of this command.

```
+-------+
| LEVEL |
|  10   |
+-------+
```

** (Comments)

Allows user to include lines of comments within a FMGR command entry list to explain command flow. This command is especially useful in procedure files.

```
**comment line
     or,
*,comment line
     or,
* comment line
```

comment line — This is any string of alphanumeric characters which must be separated from the first asterisk (*) by another asterisk, a comma, or a blank.

*** NOTE ***

This is a privileged command and is subject to the rules described in the Parameter Syntax Rules section.

Any ASCII characters can be used in the comment line.

EXAMPLE:

```
:**CHANGE LIST DEVICE TO LU 6, THE LINE PRINTER
:LL,6
:* TRANSFER CONTROL TO PROCEDURE FILE NAMED TEST
:TR,TEST
```

COMMENTS:

The lines denoted as comments do not affect command execution. They are most commonly used in procedure files (refer to the Procedure Files section in this chapter).

```
+--------+
| LEVEL  |
|  10    |
+--------+
```

AC (Allocate Cartridge)

Allocates a cartridge to the session user from the spare cartridge pool. Before the cartridge is mounted to the system and the user's session, any files residing on the allocated cartridge will be purged. The AC command is only valid when operating under session control.

```
+------------------------------------------------------------------+
|                                                                  |
| AC,crn [,P/G [,size [,id [,#dir tracks ]]]]                      |
|              \-------------v-------------/                       |
|                 (Used for specifying size                        |
|                 of cartridge required from                       |
|                 spare cartridge pool)                            |
|                                                                  |
|------------------------------------------------------------------|
|                                                                  |
| crn          Cartridge Reference Number (CRN) to be assigned to  |
|              the  allocated  cartridge  (1 to 32767  or 2 ASCII  |
|              characters).   Must  be  a  unique  CRN within the  |
|              session.                                            |
|                                                                  |
| P/G          Private  (P)  or group  (G)  cartridge designation  |
|              (default is private cartridge).                     |
|                                                                  |
| size         Requested number of tracks needed on the allocated  |
|              disc cartridge.  If defaulted, the first available  |
|              cartridge  in  the  spare  cartridge pool  will be  |
|              allocated.                                          |
|                                                                  |
| id           ASCII identifier assigned to the cartridge;  up to  |
|              6 ASCII characters.   If defaulted, the id will be  |
|              DC00XX,  where XX is the system LU of the terminal  |
|              from which the AC command is input.                 |
|                                                                  |
| #dir tracks  Number of tracks to be used  by the file directory  |
|              on the cartridge (if omitted, 1 track is assumed).  |
|                                                                  |
+------------------------------------------------------------------+
```

EXAMPLES:

```
:AC,1000,G <--- The first available cartridge in the spare
                cartridge pool is cleared (files are purged),
                assigned 1000 as its CRN, assigned and mounted
                to the caller as a group cartridge.

:AC,DD,P <----- The first available cartridge in the spare
                cartridge pool is cleared (files are purged),
                assigned DD as its CRN, assigned and mounted to
                the caller as a private cartridge.

:AC,50,,75 <--- Search the spare cartridge pool for a free
                cartridge that has at least 75 tracks. The
                first one found is cleared and 50 is assigned
                as the CRN. Cartridge is defaulted to being
                private and is mounted to the system and the
                user's session. If the cartridge has more than
                75 tracks, the entire cartridge is made
                available.
```

COMMENTS:

The AC command is a convenient way to mount cartridges to one's session. Unlike the MC command, it is not necessary to know the LU assignment for any disc cartridge.

The AC command searches the spare cartridge pool for a free cartridge (one that is not mounted to any other user or group) that is large enough to accommodate the "size" requirement, if specified. If a cartridge large enough is found, it is given the specified CRN, cleared (all files on the cartridge are purged), entered in the system cartridge list, and mounted to the user's session by creating an entry in the user's Session Control Block (SCB). The first track on the cartridge is track 0 and the file directory track will start on the last track of the cartridge (defined at system generation).

The file directory will always start on the last track of the cartridge (defined at system generation). Thus, if an available cartridge is larger than the requested size, the entire cartridge is made available.

If "size" is not specified in the AC command, the first available cartridge in the spare cartridge pool is cleared and mounted. The file directory starts on the last track of the cartridge (defined during system generation).

If the CRN specified is already mounted in the system cartridge list with the user's private or group ID, but if not currently active (refer to the DC command for an explanation of active and inactive), the specified cartridge will simply be activated. A new cartridge will not be allocated and cleared.

In the session environment, there are certain restrictions that the user should be aware of before mounting a cartridge. First, each user is restricted to a specific number of private and group cartridges that can be mounted at one time to his Session Control Block (SCB). The number is assigned by the System Manager to the user's account file entry. If the user attempts to mount more than the specified number of cartridges, an FMGR 063 error results.

Second, there must be room in the user's Session Switch Table (SST) to post the LU number of the cartridge being mounted. Also, there cannot be another session LU currently in the SST that has the same session LU as the cartridge being posted. These two conditions can cause FMGR 066 and FMGR 065 errors, respectively.

Third, if a user has a particular CRN for a cartridge he has mounted to his session as a private cartridge, he cannot mount a group cartridge to his session that also has that CRN assigned to it or vice versa. Such an attempt causes an FMGR 012 error. Also, a user cannot mount to his session a private or group cartridge with the same CRN as a system cartridge.

A system cartridge can be allocated using the AC command by the System Manager.

NOTE

> If an account in one group is linked to an account in another group, all users in both groups must be careful not to use duplicate CRNs when allocating (AC) or mounting (MC) cartridges.

```
+--------+
| LEVEL  |
|  20    |
+--------+
```

AN (Send Message to List Device)

Sends a message to the list device.

```
+-------------------------------------------------------------------+
|                                                                   |
| ANNOTATE,message                                                  |
|   ------                                                          |
|-------------------------------------------------------------------|
|                                                                   |
| message   Message to be sent to list device; maximum length of |
|           72 characters following FMGR colon (:) prompt.          |
|                                                                   |
+-------------------------------------------------------------------+
```

EXAMPLE:

```
   :LL,6<-----------------------------Change  list device  to line
                                      printer, LU 6. (Default list
                                      device is user's terminal).

   :AN,MESSAGE TO LINE PRINTER<--Message to be sent to list device

    MESSAGE TO LINE PRINTER <-----Message printed on line printer.
```

COMMENTS:

AN differs from the TE command in that it is sent to the list device, not the system console. Since it is printed on the list device, it is useful within batch job command files to annotate the job.

Another command (PA) may be used to send messages. PA suspends current operation and transfers control to a specified device. Optionally, it sends a message. This command is particularly useful to request operator intervention during non-interactive operations.

AN is a privileged command. (Refer to the Parameter Syntax Rules section in this chapter.)

```
+-------+
| LEVEL |
|  40   |
+-------+
```

CA (Calculate Globals)

Individual G and P global parameters can be assigned values or nulled with the CA command. The values assigned can be the result of arithmetic or logical calculations.

```
+------------------------------------------------------------------+
|                                                                  |
| CALCULATE,global#[,p1[,op1,p2[,op2,p3[...[op(n),p(n+1)]]]]]      |
|   -------                                                        |
|------------------------------------------------------------------|
|                                                                  |
| global#     G or P global  which  is to be set to the result of  |
|             the calculation.  Integers 1 through 9 identify the  |
|             globals  1G  through  9G  (0G  and  10G  cannot  be  |
|             modified).  Globals -36P through -1P and 1P through  |
|             6P also may be set to the result of the calculation  |
|             using the following entry form for global#:          |
|                                                                  |
|                     n:P                                          |
|                                                                  |
|             where n is the P global  to be set in the range -36  |
|             through +6 (excluding 0).   For example,  global 6P  |
|             would be specified as 6:P.                           |
|                                                                  |
| p1-p(n+1)   Values used in calculations; if omitted, global# is  |
|             nulled.                                              |
|                                                                  |
| op1-op(n)   Operations allowed on operands are:                  |
|                                                                  |
|             +       add                                          |
|             -       subtract                                     |
|             /       divide                                       |
|             *       multiply                                     |
|             O[R]    inclusive OR                                 |
|             X[OR]   exclusive OR                                 |
|             A[ND]   AND                                          |
|                                                                  |
+------------------------------------------------------------------+
```

EXAMPLES:

```
1. :CA,6,FTN7X      <---------Set global 6G to ASCII value "FTN7X".
   :DP,-16P         <--------------------Display global type of 6G.
    3               <------------------Global 6G is ASCII (type 3).
   :CA,6            <----------------Clear global 6G to null value.
   :DP,-16P         <--------------------Display global type of 6G.
    0               <-------------------Global 6G is null (type 0).


2. :CA,2,15         <------------Set global 2G to integer value 15.
   :DP,-32P         <--------------------Display global type of 6G.
    1               <----------------Global 2G is numeric (type 1).
   :CA,7,2G         <---------Set global 7G to current value of 2G.
   :DP,-12P         <--------------------Display global type of 7G.
    1               <----------------Global 7G is numeric (type 1).
   :DP,2G,7G        <--------Display contents of globals 2G and 7G.
    15,15           <------------------------Contents of 2G and 7G.
   :CA,1,2G,*,14,+,1 <------Set 1G to product of 2G and 14 plus 1.
   :DP,1G           <-----------------------Display contents of 1G.
    211             <---------------------Contents of 1G (15x14+1).
   :CA,7,7G,-,1     <----------------------------Decrement 7G by 1.
   :DP,7G           <-----------------------Display contents of 7G.
    14              <------------------------Contents of 7G (15-1).


3. :CA,1,7,OR,15    <-------------Inclusive OR 7 and 15 (octal 17),
                                                     assign to 1G.
   :CA,2,7,XOR,15   <-------Exclusive OR same values, assign to 2G.
   :CA,3,7,AND,15   <---------------AND these values, assign to 3G.
   :DP,1G,2G,3G     <---------------Display globals 1G, 2G, and 3G.
    15,8,7          <-----------Contents of globals 1G, 2G, and 3G.


4.  :CA,1:P,8P      <-------Set global 1P to value stored in 8P (8P
                                 is the system LU of the terminal).
```

COMMENTS:

Within each CA command the order of the evaluation proceeds from left to right until the arithmetic or logical calculations have been completed.

The type of the result depends on the type of the operands. If operand types differ in any one CA statement, the highest type value is used, where type 0=null, type 1=numeric, and type 3=ASCII in ascending order from 0 to 3.

Except for divide and multiply, calculations are performed separately on each word of three-word ASCII globals. For divide and multiply, all three words of the first operand are divided or multiplied by word 1 of the second operand.

In its simplest form, CA is used to null an individual global parameter or to set an individual global to the value of pl.

Globals and global types are discussed in the Global Parameters section. The CA command is generally used within procedure files. For a discussion of procedure files, refer to the Procedure Files section in this chapter.

```
+--------+
| LEVEL  |
|  10    |
+--------+
```

CL (Cartridge List)

Displays list of mounted cartridges.

```
+------------------------------------------------------------------+
|                                                                  |
| CL      <----------Displays list of all cartridges accessible    |
|                    by user.                                      |
|                                                                  |
| CLALL   <----------Displays list of all cartridges in system     |
|   --               cartridge list.                               |
|                                                                  |
|------------------------------------------------------------------|
|                                                                  |
| No parameters required.                                          |
|                                                                  |
+------------------------------------------------------------------+
```

EXAMPLES:

```
   :CL
     LU   LAST TRACK    CR    LOCK  P/G/S

     35      00202   00033          P
     31      00202   01000          G
     02      00202   00002          S
     10      00202   00050          S


   :CLAL
     LU   LAST TRACK    CR    LOCK  P/G/S  USER/GROUP

     02      00202   00002          S    MANAGER.SYS
     10      00202   00050          S    MANAGER.SYS
     32      00202   01000          G    ACCTG
     39      00202   02000
     31      00202   01000          G    MANUF
     35      00202   00033          P    BROWN.MANUF
     33      00202   01000          G    QA
     38      00202   00016          P    SMITH.QA
                                         JOHNSON.QA
     40      00101   05000          P    WILSON.MANUF
     42      00501   00000          S    FILE SYSTEM DISC LU
```

COMMENTS:

The cartridge list is issued to the list device (default is the user's terminal). The list contains the following categories:

LU <---------------------------------LU of the cartridge.

LAST TRACK <---Last track assigned to the FMP on that cartridge.

CR <--------------------------------CRN of the cartridge.

LOCK <-----Name of program locking the cartridge; blank if not locked.

P/G/S <-------Indicates whether the cartridge is mounted as private, group or system. If blank, indicates non-session cartridge.

USER/GROUP <--Name of the user or group to whom the cartridge is mounted. If blank, indicates non-session mounted.

The CL command lists only those cartridges mounted to the user, the user's group or the system. With the CLAL command, all cartridges mounted to all users of the system are included in the list including non-session users. Note that LU 39 in the previous example is a non-session cartridge.

For non-session users, only non-session and system cartridges are included in the list with the CL command. The CLAL command lists all cartridges mounted to the system including session cartridges. If the Session Monitor software modules are not included in the system, the CL and CLAL commands will produce the same lists.

Some cartridges can be shared between session users by linking the users' accounts together (linked by the System Manager). Note that LU 38 in the previous example is shared by the two users SMITH.QA and JOHNSON.QA. Only these users (and the System Manager) can access this cartridge. When either of these two users logs on, LU 38 will be mounted to his session as a private cartridge. For more details on account linking refer to the RTE-6/VM System Manager's Manual.

```
+-------+
| LEVEL |
|  20   |
+-------+
```

CN (Control Non-Disc Device)

Controls non-disc devices such as magnetic tape, minicartridges and type 0 files.

```
+-----------------------------------------------------------------+
|                                                                 |
| CN [,namr [,function [,subfunction ]]]                          |
|                                                                 |
|-----------------------------------------------------------------|
|                                                                 |
| namr         LU of the device  to be controlled  or its type 0 |
|              file name  previously  defined  in a  CR command; |
|              default is 8  (recommended LU for magnetic tape). |
|              The range of session LUs is from 1 to 63.          |
|                                                                 |
| function     Control  function  to  be  performed  on non-disc |
|              device.  Can be either two characters mnemonic or |
|              octal function code.  The mnemonic codes are:     |
|                                                                 |
|              RW - Rewind (default for mag tape, terminal cart- |
|                   ridge tape unit, and mass storage devices).  |
|              EO - End-of-file                                   |
|              TO - Top-of-form (default for line printer and    |
|                   terminal CRT).                                |
|              FF - Forward Space File                            |
|              BF - Backspace File                                |
|              FR - Forward Space Record                          |
|              BR - Backspace Record                              |
|              LE - Leader (default for paper tape punch).        |
|                                                                 |
| subfunction  Carriage control  characters  for line printer or |
|              terminal; use if "function" is "TO" (top-of-form).|
|              The subfunction can be:                            |
|                                                                 |
|              0  - To suppress spacing on next print  operation |
|                   only.                                         |
|              +n - To space n lines before next print operation.|
|              -n - To page eject  on  line printer  or  space n |
|                   lines on terminal.                            |
|                                                                 |
+-----------------------------------------------------------------+
```

EXAMPLES:

1. Rewind magnetic tape:

```
:CN      <--------Defaults to LU 8 (mag tape) which has default
 or                function of RW (rewind).
:CN,8
 or
:CN,8,RW
```

2. Eject to top of new page on the line printer:

```
:CN,6    <------Default function for line printer  (LU 6) is TO
 or             (top-of-form). For line printer, TO causes page
:CN,6,TO        eject.
```

3. Space two lines on your terminal and return the carriage:

```
:CN,1    <---Default function for terminal is TO (top-of-form).
 or          For terminal,  TO causes spacing  down two  lines
:CN,1,TO     and carriage return.
```

4. Skip 5 spaces on the line printer (no page eject):

```
:CN,6,TO,5
```

5. Space forward one record on magnetic tape:

```
:CN,8,FR
```

6. Backspace one file on type 0 file MT assigned to LU 8 at creation.

```
:CN,MT,BF
```

7. Write end-of-file mark on magnetic tape:

```
:CN,,EO
```

COMMENTS:

The CN command is similar to the FCONT subroutine call (refer to the RTE-6/VM Programmers Reference Manual for details on this subroutine). The function codes used in the CN command correspond to the FCONT function codes.

The function default values are determined from the driver type of the device. For the octal function codes and default values refer to the appropriate driver manual for the device to be controlled.

```
+-------+
| LEVEL |
|  20   |
+-------+
```

CO (Copy Disc Files)

All or selected files on a currently mounted cartridge can be copied to or moved to another currently mounted cartridge with the CO command. When operating in a session environment, the caller must have both cartridges mounted to his session. The CO command cannot copy type 0 files.

```
+---------------------------------------------------------------------+
|                                                                     |
|   COPY,cartridgel,cartridge2[,options[,namel[,name2[,msc]]]]        |
|     --                                                              |
| or                                                                  |
|          file                                                       |
|   COPY,descriptor,cartridge2[,options[,namel[,name2[,msc]]]]        |
|     --                                                              |
|---------------------------------------------------------------------|
|                                                                     |
| cartridgel  Cartridge Reference Number (CRN) of mounted             |
|             cartridge containing files to be copied; if             |
|             negative, identifies cartridge LU.                      |
|                                                                     |
| file        Name of the file to be copied. File descriptor          |
| descriptor  subparameters are optional. (Refer to the Namr          |
|             Parameter section for details.)                         |
|                                                                     |
|             A minus sign (-) can be placed in any position of       |
|             the file name and is used to match any character        |
|             (refer to the DL FMGR command).                         |
|                                                                     |
| cartridge2  Cartridge Reference Number (CRN) of mounted             |
|             cartridge to which files are to be transferred; if      |
|             negative, identifies cartridge LU.                      |
|                                                                     |
+------------------CONTINUED NEXT PAGE--------------------------------+
```

CONTINUED FROM PREVIOUS PAGE

options — Various copy options (Refer to the comments section below for a more detailed explanation of these options.

- C Clear destination cartridge before copying files.
- D Dump-mode (store file even if it exists on destination cartridge).
- E Eliminate extents on copied files.
- P Purge source files after copy.
- V Verify that files are copied correctly.

name1 — Starting file name (first file to be copied to destination cartridge). If specified, it must reside on the source cartridge, or files will not be copied.

name2 — Ending file name (last file to be copied to destination cartridge). If specified, it should reside beyond name1 in the source cartridge file directory.

msc — Master security code; must be 2 ASCII characters. 'msc' is used in conjunction with the purge and/or clear options and only when running outside of session.

*** NOTE ***

Files are transferred record by record. Records longer than 128 words are truncated.

EXAMPLE:

Assume files &A, %A, &B, %B, &C, %C, and &D on cartridge LA are to be copied to cartridge LB, and a file &C already exists on cartridge LB:

```
1. :CO,LA,LB <---Copy files contained on cartridge LA to cartridge
                 LB.
   &A
   %A <-------------System  prints  file names as they are copied.
   &B
   %B
   &C
   FMGR -002 <------Error  message  indicating  &C  is a duplicate
                    name; it is not copied.
   %C <-------------System indicates that file %C is being copied.
   &D
   :
```

2. Move program source files (&) on cartridge LA to cartridge LB. Replace files with the same name on the destination cartridge (must have privileged access to the destination cartridge if selected files have non-zero security codes). The msc is SC and is required if not in session.

```
   :CO,&---::LA,LB,PDV,,,SC
   &A
   &B
   &C <--------------------------------Note that &C was replaced.
   &D
   :
```

3. Copy all files on cartridge LA starting with the file &B and stopping with &D, to the cartridge LB. Also, verify the copies.

```
   :CO,LA,LB,V,&B,&D
   &B <----------------------------------First file to be copied.
   %B
   &C
   FMGR-002 <----------------Duplicate file name, file not copied.
   %C
   &D <-----------------------------------Last file to be copied.
   :
```

COMMENTS:

As each file is copied, its name is displayed on the log device. If a file on cartridge2 has the same name as a file being transferred from cartridge1, the file is not transferred and an informative message is sent to the log device.

If you specify two ASCII characters for the first parameter and omit file descriptor subparameters, it will be interpreted as the source CRN and all files residing between name1 and name2 (inclusive) on the source cartridge will be copied. However, if you specify two ASCII characters for the first parameter and include a cartridge identifier subparameter with it, it will be interpreted as file descriptor format.

The files being copied are not affected by the copy except by the 128 words record length restriction. For type 2 files containing records longer than 128 words, a warning message is displayed indicating that records are truncated to 128 words. If files already exist on the cartridge to which files are being copied, these files are not affected. Entries for the copied files are added to the file directory on cartridge2. If there were any entries for purged files in this directory, entries for the copied files may be interspersed with entries for existing files. To know where the new file entries are placed, request a directory list with the DL command. The CO command cannot be used to copy type 0 files.

For copying files from disc cartridges to or from tape devices, the FC program, described in the RTE-6/VM Utilities Manual, can also be used.

The following options give you additional control over the transfer of files. The different options available are:

C option The C (clear) option will clear the destination cartridge before any files are copied to it (similar to using the IN command). If this option is specified from an interactive terminal, you will receive a FMGR 060 message. A NO answer will abort the CO command. A YES answer will clear the destination cartridge of all files. If any other response is given the FMGR 060 message will be repeated. The user must have privileged access (see below) to the destination cartridge in order to use this option. If privileged access is not allowed then either a FMGR 46 error (if in session) or a FMGR 51 error (if not in session) will occur.

D option When the D (dump) option is specified the normal copy process still takes place. In addition when a duplicate file name is encountered on the destination cartridge, an attempt is made to purge that file and replace it with the file on the source cartridge. Purging and replacing of the destination file only occurs if the security code on both files match or the security code of the destination file is zero, otherwise, a FMGR-007 error will occur leaving the destination file unchanged and the source file uncopied.

E option The E (eliminate extent) option will eliminate extents from the resulting destination files. A destination file's size is calculated from the source file's size and the number of extents used.

P option The P (purge) option will attempt to purge the selected files from the source cartridge as they are copied to the destination cartridge. This results in "moving" the file from the source to the destination cartridge. To purge a file, the security code of the file must match the security code in the subparameter of file descriptor, or the user must have privileged access (see below) to the source cartridge. If privileged access is allowed, files are purged regardless of their security codes. The purge option does not affect whether a file will be copied or not.

V option The V (verify) option will compare the source and destination files after the copy to insure that transfer was correct. If the files differ, CO will terminate with a FMGR-049 error.

The P (purge) and C (clear) options check if the user has privileged access to either the source or the destination cartridges. Privileged access to a disc cartridge is granted under the following conditions. If in session, privileged access is allowed if the user is the System Manager, the cartridge is a private cartridge belonging to the user, or the cartridge is a group cartridge and the user is the group manager for the group. If not in session, then privileged access is allowed only if the user supplies the msc (master security code) parameter.

The option characters may be combined together (in any order) to specify several options. They are supplied in the third parameter with no delimiters between them.

Option Examples:

DV Copies files in dump-mode, and verifies their transfer.

CP Clears the destination cartridge first, and purges the source files after copying.

CDEVP Clears the destination cartridge first, copies files in dump-mode, elimates any extents, verifies their transfer and purges the files after copying.

Terminating the COPY Operation

There are several ways to terminate the CO command:

1. Normal termination occurs when the selected files have been correctly copied to the destination cartridge. The 1P global parameter will be set to 0 (zero) to indicate a successful termination.

2. Abnormal termination occurs when a V option is specified and the file was not verified correctly. A FMGR-049 error will be generated, and the name of the bad file will be placed into the 10G parameter.

3. OF,FMGxx or BR command is requested from break-mode. If operating under session, the break-mode RS command may also be used. The CO command will terminate before it starts to copy the next file. A FMGR 022 error will be generated, and the name of the next file to be copied will be placed into the 10G global parameter.

4. The destination runs out of file space or directory space. If a selected file cannot be copied to the destination cartridge because of space problems, the CO command will abnormally terminate with a FMGR-014 or FMGR-033 error. The name of the file which could not be copied will be placed into the 10G global parameter.

   To recover from one of the above FMGR errors, you may restart the CO command to the same destination cartridge by specifying the 10G global parameter for the starting file name. To display the file name, type "DP,10G" to the file manager.

Speeding Up the COPY Operation

CO uses any available memory after the last word of the FMGR program as disc buffers for the copy operation. Therefore, the speed of CO can be increased by using the SZ command to increase the size of the partition in which FMGR is run.

Non-Sequentially Extended Type 1 and 2 Files

By doing random writes, it is possible to create a type 1 or 2 file whose extents do not all exist. For example, a file may consist of the main extent, followed by extent 1, followed by extent 5. By default, the CO command will copy this type of file "as is". The destination file will consist of the same number of actual extents as the source file. Note this differs from the ST or DU FMGR commands which create all intermediate missing extents when copying a file. If the E (eliminate extents) option is specified then the CO command will create the destination file with only a main extent (large enough to contain all intermediate missing extents), as well as all actual extents present in the source file. In this case the destination file will require more disc space than the source file. A subsequent comparison of the source and destination files may show that they are not identical since the data which is written to the destination file is undefined for extents missing in the source file. The V (verify) option on the CO command will still function properly in this case.

```
+-------+
| LEVEL |
|  20   |
+-------+
```

CR (Create a Disc File)

Creates a disc file of specified type and size (no data is transferred to the file).

```
+------------------------------------------------------------------+
|                                                                  |
| CREATE,file descriptor                                           |
|   ----                                                           |
|------------------------------------------------------------------|
|                                                                  |
| file         Omitted subparameters  default to zero;  file type |
| descriptor   and file size  must be specified  as  greater than |
|              zero;  record size need be specified only for type |
|              2 files.                                            |
|                                                                  |
+------------------------------------------------------------------+
```

EXAMPLES:

1. Create a disc file named MYFILE which is a type 4 file, has a security code of -25 (read and write are restricted to users knowing the code), is allocated to a cartridge with CRN 100, and uses 10 blocks (20 sectors) of disc space.

   ```
   :CR,MYFILE:-25:100:4:10
   ```

2. Create a disc file named URFILE which is a type 2 file, uses 20 blocks, and has 72 words per record. Security code and cartridge are defaulted.

   ```
   :CR,URFILE:::2:20:72
   ```

3. Create a disc file named MYFILE which is a type 3 file with security code EJ (only write restricted) on cartridge 100 and allocate the remaining unused portion of the cartridge up to the maximum allowed file size to the file.

   ```
   :CR,MYFILE:EJ:100:3:-1
   ```

COMMENTS:

When a disc file is created, an entry is made in the file directory on the cartridge to which the file is allocated. If the cartridge is specified, the file is allocated to that cartridge; otherwise, user private cartridges are searched until one is found with enough room to accommodate the file. If no private cartridge is available or none can accommodate the file, group cartridges and then system global cartridges are searched. If the user is not under session control, the file is sent to the first non-session or system cartridge found with enough room starting at the head of the cartridge directory on the system disc. If a file with the given name already exists on the first cartridge with enough space, a FMGR error -002 is issued.

When creating a file with a size of -1 (all of the unused space on the cartridge), disc space may be wasted if a bad track exists on the cartridge and has been flagged with the IN (initialize cartridge) command. When a file is being created, the location of the starting track is increased until the file contains no bad tracks. Therefore, if a file is to use the rest of the cartridge (size of -1), it is allocated an area above the highest numbered bad track. Any space between the first available track and the highest numbered bad track is wasted.

When bad tracks exist on a disc LU, the actual file size should be specified in the CR command or the bad tracks should be spared using a disc formatting utility (for example FORMT). For more information on how the file system compensates for bad tracks, consult the IN command description in this chapter.

If a file is type 3 or greater, an end-of-file mark is written at the beginning of the file. As data is entered serially in the file, the mark is moved to the end of the data.

The format of a file directory entry for a created file is illustrated in Appendix C. The information in the entire file directory (all files on the cartridge) can be listed with the DL command. Information in an individual file can be listed with the LI command.

```
+--------+
| LEVEL  |
|  20    |
+--------+
```

CR (Create a Non-Disc File)

Creates non-disc (type 0) files by creating a file directory entry that specifies device control information.

```
+-----------------------------------------------------------------+
|                                --                          --   |
|                                |        --              --|    |
|                                |        |         --    --||   |
|          file       ,RE[AD]    |,BS[PACE]|,EO[F]    |,BI[NARY]|||  |
| CREATE,descriptor,lu,WR[ITE]|,FS[PACE]|,LE[ADER]|,AS[CII]  |||  |
|   ----               ,BO[TH]    |,BO[TH]   |,PA[GE]    |,contword|||  |
|                                |        |,contword--      --||   |
|                                |        --              --|    |
|                                --                          --   |
|-----------------------------------------------------------------|
|                                                                 |
| file         Only the file name and, optionally, the security   |
| descriptor   code and the cartridge are specified; file type is |
|              default value 0 and other subparameters do not     |
|              apply.                                             |
|                                                                 |
| lu           LU of the non-disc device; a positive integer.     |
|                                                                 |
| READ/WRITE   Specify the legal input/output mode of the device;|
| or BOTH      it must be specified, there is no default.         |
|                                                                 |
|              RE - device accepts input only; forward spacing is |
|                   assumed.                                      |
|              WR - device is output only;  no forward spacing is |
|                   supported.                                    |
|              BO - device is used for  input  or  output;  back- |
|                   spacing and forward spacing are legal.        |
|                                                                 |
| BSPACE,      Specify  the type  of spacing the device supports; |
| FSPACE, or   if omitted, FSPACE is assumed for READ devices and |
| BOTH         no spacing for other devices.                      |
|                                                                 |
|              BS - backspacing is supported.                     |
|              FS - forward spacing is supported.                 |
|              BO - both forward and backspacing are supported.   |
|                                                                 |
+-----------------------------------------------------------------+
```

CR (non-disc file) . . . cont'd

EOF, LEADER, PAGE or contword — Specify the particular type of end-of-file to be written on the device; if omitted, default depends on driver type.

EO - end of file mark for magnetic tape (default if device has driver type greater than 16 octal, magnetic tape, or mass storage device).
LE - leader on paper tape (default if driver type 02, paper tape punch).
PA - page eject for line printer or two line feeds on teleprinter (default if not a punch or if driver type less than 17 octal).

contword (control word)-control subfunction (equivalent to function code in FCONT, refer to the RTE-6/VM Programmer's Reference Manual); supplied if further end-of-file definition needed; specify as octal integer of which only least 5 bits are used.

BINARY, ASCII or contword — Specify the type of data on the device; ASCII is the default.

BI - binary data
AS - ASCII data

contword (control word) - subfunction (equivalent to bits 6-10 of IOPTN parameter in FMP OPEN call, refer to the RTE-6/VM Programmer's Reference Manual); supplied if further data definition needed; specify as decimal or octal integer of which only the lowest five bits are used.

EXAMPLES:

1. Create non-disc file named LP as output file for LU 6 (line printer); defaults are no spacing and ASCII data. The type 0 file is created on CRN 20.

   ```
   :CR,LP::20,6,WR,,PA
   ```

2. Create non-disc file named MT as input/output file for LU 8 (mag tape); both forward and back spacing supported; security code is 32107.

   ```
   :CR,MT:32107,8,BO,BO
   ```

3. Create a read-only magnetic tape file.

   ```
   :CR,MAG:JT,8,RE
   ```

4. Create non-disc file named READS as input only file on LU 5.

   ```
   :CR,READS,5,RE
   ```

COMMENTS:

Programs can use non-disc (type 0) files as a means of controlling access to a device. Thus, type 0 files provide a measure of device independence in that the standard file calls can be used to control a peripheral device.

When a type 0 file is created, an entry is made in the file directory on the disc.

The type 0 file entry differs from the disc file entry in that control information replaces the track, sector, and record length information. The directory entry for type 0 files is illustrated in Appendix C.

In general, a type 0 file can be specified with only the required parameters. FMGR needs a name, the LU, and whether the device is read only, write only, or both. The other parameters usually follow from this information.

```
+-------+
| LEVEL |
|  30   |
+-------+
```

CS (Modify Spool Options)

Allows a user to modify the spool options set up by the SL command. The CS command is used to complete the spooling operation. For more details on spooling, refer to the RTE-6/VM Batch and Spooling Reference Manual.

```
+------------------------------------------------------------------+
|                                                                  |
|      CS,lu,attribute[,outlu[,priority]]                          |
|                                                                  |
|------------------------------------------------------------------|
| lu          LU defined at spool set up by the SL command.        |
|                                                                  |
| attribute   One of the following:                                |
|                                                                  |
|             EN  END;   write  a  final  EOF  on  spool file and |
|                 terminate the spool.   Spool file  is placed in |
|                 outspool queue. (This is the default attribute.)|
|                                                                  |
|             RW  REWIND; reset the file to the first record.      |
|                                                                  |
|             PU  PURGE;  file  is to be purged  on completion of |
|                 outspooling.  Change the save flag  set by the |
|                 SL command to purge.                             |
|                                                                  |
|             SA  SAVE;  file  is to be saved  on  completion  of |
|                 outspooling.  Change the purge flag set by the |
|                 SL command to save.                              |
|                                                                  |
|             PA  PASS; remove the hold option.                    |
|                                                                  |
|             BU  BUFFERED; change file to buffered.               |
|                                                                  |
|             NB  NOT BUFFERED; change file to not buffered.       |
|                                                                  |
|             NP  NOT PASS; change LU and/or priority information |
|                 by specifying  the  two  additional parameters, |
|                 outlu and priority.                              |
|                                                                  |
|  outlu      new LU                                               |
|                                                                  |
|  priority   new priority                                         |
+------------------------------------------------------------------+
```

EXAMPLES:

1. In a session, dump file ASC to the line-printer via spooling:

```
:SL,20,ASC,,6 <---Set up spool equivalence (refer to the FMGR
                  SL command).

:CS,20 <----------Close spool file.
```

2. In a session, assign LU 6 to user file LIST for write only. File is to be held from outspool queue and saved at set-up.

```
:SL,6,LIST,WR

:CS,6,PA <--------Pass file LIST to outspool queue.
```

COMMENTS:

The CS command can be used to change certain of the spool file attributes either defaulted or assigned by the SL command. The CS command can also be used to specify the type of outspool device to be used as well as the device LU and outspool priority. The outlu and priority parameters are valid only if an outspool LU was specified when the spool file was set up. Refer to the RTE-6/VM Batch and Spooling Reference Manual for further details on spooling.

```
+-------+
| LEVEL |
|  20   |
+-------+
```

CT (Control Terminal)

Issues a control request to an interactive terminal, optionally writing a message to the terminal.

```
+--------------------------------------------------------------+
|                                                              |
| CT, namr [,function [,subfunction [,message ]]]              |
|                                                              |
|--------------------------------------------------------------|
|                                                              |
| namr        Type 0 file name or LU of an interactive device. |
|                                                              |
| function    Control function to be performed on interactive |
|             device.  Default is to enable terminal (octal   |
|             function code 20B).                              |
|                                                              |
|             11B - Space down a specified number of lines (used |
|                   in conjunction with subfunction parameter). |
|                                                              |
|             20B - Enable terminal.                           |
|                                                              |
|             21B - Disable terminal.                          |
|                                                              |
|             22B - Set time-out for terminal (used in conjunc- |
|                   tion with subfunction parameter).          |
|                                                              |
| subfunction Additional parameter which may be required with |
|             function. Default is no subfunction.             |
|                                                              |
|             function subfunction                             |
|                                                              |
|                11B       0  - To space down 2 lines before next |
|                               print operation.               |
|                          +n - To space down n+1 lines before |
|                               next print operation (i.e.,skip n |
|                               lines).                        |
|                          -n - To space down n+1 lines before |
|                               next print operation (i.e.,skip n |
|                               lines).                        |
|                                                              |
+--------------------------------------------------------------+
+---------------------------------------------------------------+
|                                                               |
| CT (control terminal) . . . cont'd                            |
|                                                               |
+---------------------------------------------------------------+
|                                                               |
|            function subfunction                               |
|                                                               |
|               20B  Only  required  when  enabling  multipoint |
|                    terminals   (refer  to  91730A  Multipoint |
|                    Terminal Interface Subsystem User's Guide; |
|                    Guide; part no. 91730-90002).              |
|                                                               |
|               22B  Timeout value in units of 10 milliseconds. |
|                                                               |
| message    Message  to be written to terminal;  default is no |
|            message.                                           |
|                                                               |
+---------------------------------------------------------------+
```

EXAMPLES:

```
  :CT,20,,,TERMINAL READY <---Enable terminal  with system  LU 20,
                              sending   "TERMINAL  READY"  message
                              (input from non-session mode).

  :CT,20,21B              <---Disable terminal  with system  LU 20
                              (input from non-session mode).
```

COMMENTS:

The CT command is similar to the CN command. CT is a specialized command used to enable terminals. Unlike the CN command, CT allows the user to specify system LUs greater than 63 when operating out of session. It is useful for enabling terminals in the system WELCOM file (refer to the RTE-6/VM System Manager's Manual for more details).

```
+-------+
| LEVEL |
|  10   |
+-------+
```

DC (Dismount Cartridge)

Logically removes a disc cartridge from a user's environment.

```
+--------------------------------------------------------------+
| DC, cartridge <-----sets inactive bit in Session Control Block |
|                     entry;  if operating non-session,  deletes |
|                     cartridge entry in system cartridge list.  |
|                                                              |
| DC, cartridge, RR <-deletes cartridge entry in Session Control |
|                     Block and system cartridge list  releasing |
|                     cartridge resource back to system; in non- |
|                     session environment,  RR option is ignored |
|                     and DC operates same as DC, cartridge.     |
+--------------------------------------------------------------+
|                                                              |
| cartridge  Cartridge identifier;  positive or alphanumeric CRN |
|            assigned  to  cartridge  or  negative LU associated |
|            with cartridge.                                     |
|                                                              |
| RR         Optional parameter for session users only; specifi- |
|            cation of  RR deletes  the cartridge's entry in the |
|            system cartridge list.  If the cartridge originally |
|            came from the spare cartridge pool, it is returned. |
|                                                              |
|                      * * * W A R N I N G * * *                 |
|                                                              |
| If the cartridge  being  dismounted  originally  came from the |
| spare cartridge pool,  before  specifying  the  RR option make |
| sure that there are no files on the cartridge  that need to be |
| saved.   If the RR option is specified,  the cartridge will be |
| released back  to the spare  cartridge  pool.   If a user then |
| mounts that cartridge with the  AC command  or  with the READT |
| utility, all previous files on that cartridge will be lost.    |
|                                                              |
| If LU 2 is dismounted  out of session,  it  is  taken  off the |
| cartridge list and thus not available  to  session users.  The |
| System Manager must do a  "DC,-2,RR" to remount it as a system |
| disc.                                                          |
|                                                              |
+--------------------------------------------------------------+
```

EXAMPLE:

```
   :CL
     LU    LAST TRACK      CR       LOCK     P/G/S
     35     00202        00033                P
     31     00202        01000                G
     02     00202        00002                S
     10     00202        00050                S

   :DC,33 <--set inactive bit in cartridge entry in user's Session
                Control Block (SCB).

   DISC CRN 33 LU 35 INACTIVE

   :CL
     LU    LAST TRACK      CR       LOCK     P/G/S
     31       00202      01000                G
     02       00202      00002                S
     10       00202      00050                S
```

Cartridge 33 is no longer in the user's operating environment, but is still mounted to the user in the system cartridge list.

```
   :CLAL
     LU    LAST TRACK      CR       LOCK     P/G/S   USER/GROUP
     02       00202      00002                S      MANAGER.SYS
     10       00202      00050                S      MANAGER.SYS
     32       00202      01000                G      ACCTG
     31       00202      01000                G      MANUF
     35       00202      00033                P      BROWN.MANUF
     33       00202      01000                G      QA
     38       00202      00016                P      SMITH.QA
     39       00202      02000
```

Cartridge 33 is still mounted to user (BROWN.MANUF), but has been set inactive by prior DC,33 command.

```
   :MC,35 <----Resets inactive bit; places cartridge back into
                  user's operating environment. (Note that the MC
                  Command requires that the cartridge be specified by
                  its LU.)

   :CL
     LU    LAST TRACK      CR       LOCK     P/G/S
     35       00202      00033                P
     31       00202      01000                G
     02       00202      00002                S
     10       00202      00050                S
```

To return the cartridge back to the system, the RR option (release resource) must be specified.

```
   :DC,33,RR
   DISC CRN 33 LU 35 DISMOUNTED FROM SYSTEM (POOL)

   :CLAL
     LU   LAST TRACK      CR       LOCK    P/G/S   USER/GROUP
     02      00202      00002                S     MANAGER.SYS
     10      00202      00050                S     MANAGER.SYS
     32      00202      01000                G     ACCTG
     31      00202      01000                G     MANUF
     33      00202      01000                G     QA
     38      00202      00016                P     SMITH.QA
     39      00202      02000
```

Cartridge 33 is no longer mounted to BROWN.MANUF. It has been returned to the system and is now available for any user to mount it.

COMMENTS:

When the session user (not the System Manager) does a dismount cartridge operation using the DC command, FMGR checks to see whether the cartridge specified is mounted as a private or group cartridge within the user's Session Control Block (SCB). If it is not, a FMGR 054 error (DISC NOT MOUNTED) results. The System Manager can dismount any cartridge in the system.

If the session user does not specify the RR option in the DC command, the cartridge will not be returned to the system. An inactive bit is set in the cartridge entry in the user's SCB to logically remove it from the user's operating environment. The cartridge would then be omitted from any file search operation which occurs when a cartridge is not specified.

If the RR option is specified, FMGR checks to see whether the cartridge specified is mounted as a private or group cartridge within the user's SCB. If it is, then FMGR removes its entry from the user's SCB. FMGR then checks to see whether the cartridge can be removed from the system cartridge directory. If one or more users have the cartridge mounted to their session, the cartridge is not removed from the system cartridge directory when dismounted. If no other user has it currently mounted, the entry for the cartridge is removed from the system cartridge directory. FMGR then searches for any ID segments restored by the RP command which point to the cartridge. If such a file exists a FMGR 011 error is displayed followed by a list of all restored programs pointing to the cartridge and the cartridge is not dismounted. If a file is open and the user tries to dismount a cartridge, a FMGR-008 error is returned and the cartridge is not dismounted.

When a non-session user does a dismount cartridge operation using the DC command, FMGR checks the list of non-session cartridges for the specified cartridge to be dismounted. If the cartridge specified is not a non-session cartridge, a FMGR 054 error results. If it is, the cartridge entry in the system cartridge list is removed.

LU 2 and LU 3 (if used) cannot be removed from the system cartridge directory. The command DC,2,RR will remove the entry for LU 2, but it will also put a new entry for LU 2 at the bottom of the directory. Therefore, the DC command can be used to place LU 2 or LU 3 at the bottom of the directory but not to remove it. Note that when an entry for LU 2 or LU 3 is put back into the cartridge directory, the ID of the user issuing the DC command is used in the cartridge directory entry (i.e., System Manager or non-session user). This is how LU 2 or 3 can be made system type or non-session type disc.

LEVEL
10

DL (Directory List)

Provides a list of all FMP files on a specified cartridge, a list of all FMP files on all cartridges within the user's operating environment, or optionally, a list of files with common file descriptor characteristics. DL locks then immediately unlocks the cartridge being accessed (this clears any invalid open flags).

```
DL [,cartridge [,master security ]]
   or
DL,file descriptor [,master security ]
```

cartridge	Cartridge identifier; positive for CRN, negative for LU; if omitted or zero, the directories of all cartridges mounted to the user are listed.
master security	Code assigned to the system at initialization; if correctly specified, directory list includes file security code and track and sector address for each file (long list).
file descriptor	Omitted subparameters default to zero. As described below, minus signs (-) can be used as place holders in the file name to allow more flexibility.

EXAMPLES:

```
   :DL,2    <-------------------List all FMP files on cartridge 2.

CR=00002
 ILAB=SYSTEM NXTR= 00101 NXSEC=040 #SEC/TR=096 LAST TR=00255 #DR TR=01

  NAME   TYPE   SIZE/LU    OPEN TO

  +#CCT! 00001 00057 BLKS LOGON LGOFF
  WELCOM 00004 00001 BLKS
  "WELCO 00004 00002 BLKS
  "DEV   00004 00002 BLKS
  "LCHEL 00004 00003*BLKS
  "HELLO 00004 00003 BLKS
```

```
:DL,A            <-------------List all files whose name is A.
:DL,A-----       <----List all files whose name starts with A.
:DL,--A--A:-5:2  <---List  all files  whose 3rd  and 6th  name
                     characters are A  and whose security code
                     is -5 or +5 and which are on cartridge 2.
:DL,------:::1   <-----------------List all files of length 1.
:DL,------::::16 <-------List all files with record length 16.
```

COMMENTS:

The DL,file descriptor format shown above allows the user more flexibility when specifying which directory entries are to be listed. When this format is used, the following conditions must be met before a given file entry will be listed:

a. The file name must match the name portion of file descriptor except that the minus sign (-) if used in file descriptor "matches" any character.

b. Zero as a subparameter matches any actual subparameter, however, if a non-zero subparameter is used, it must match the file's actual parameter.

c. The standard security code match is used, i.e.,
-n matches   n and -n
+n matches   n only

The directory list is provided in two formats: a short list and a long list. Both lists have the same header information describing the cartridge itself; they differ in the file information provided. The long list includes a file security code for each file and the track and sector address for the file.

For session users, only the directories of those cartridges mounted to the user, the user's group or the system are listed. For non-session users, only the directories of non-session cartridges and system cartridges are listed.

The asterisk preceding the word BLKS in the directory listing is printed if the file size is in 128-block multiples. In the first example, note that the file size for "LCHEL is 3 128-block multiples.

```
+-------+
| LEVEL |
|  20   |
+-------+
```

DP (Display Parameters)

Displays parameters onto the log device.

```
+-------------------------------------------------------------------+
|                                                                   |
| DP [,p1 [,p2... [,p14 ]]]                                         |
|                                                                   |
|-------------------------------------------------------------------|
|                                                                   |
| p1-p14  Parameter values  or  global names to be displayed; if    |
|         omitted, nothing is displayed (up to 14 parameters can    |
|         be displayed).                                            |
|                                                                   |
+-------------------------------------------------------------------+
```

EXAMPLES:

1. Display non-global parameters:

```
   :DP,MESSAGE TO LOG DEVICE <---Message to be sent to log device.

     MESSAGE TO LOG DEVICE <------Message displayed on log device.

   :DP,WORD,10B,-1,-32768,50 <---Parameters to be displayed on log
                                  device.
     WORD,10B,-1,-32768,50 <---Parameters displayed on log device.
```

2. Display values of globals 1G and 2G passed in a TR command:

```
   :TR,TEST,PROG,1 <-----Transfer to procedure file, TEST, passing
                          parameters PROG  and 1 through 1G and 2G.
   :DP,1G,2G <-----------Display  contents  of  globals 1G  and 2G
                          (command is within procedure file, TEST).
     PROG,1 <------------------Parameters displayed on log device.
```

3. Assume 1G and 2G have values set in previous example and that value of 3G is null. Display global types of 1G, 2G and 3G.

```
   :DP,-36P,-32P,-28P <----Display contents of P globals (refer to
                            Table 3-3 for global equivalence).
     3,1,0 <---------------Global types for 1G, 2G and 3G displayed
                            (3=ASCII, 1=numeric, 0=null).
```

COMMENTS:

This command is commonly used to display global values. It may, however, be used to display any parameter. Numeric values are printed as decimal values with no leading sign when they are positive, as decimal values with a leading minus sign when negative.

Null globals are displayed as adjacent commas (,,) unless they are P globals which are never null.

The P globals can be used to display the global type (refer to the Global Parameters section in this chapter).

The display is not inhibited by a severity code greater than 0.

```
+--------+
| LEVEL  |
|  20    |
+--------+
```

DU (Transfer Data to Existing File)

Transfers data from an existing file or LU to another existing file or LU. For copying files from disc cartridges to or from tape devices, the FC program described in the RTE-6/VM Utilities Manual can also be used.

```
                    --                                      --
                    |                          --         --|
                    |,record format,eof control|     --   --||
 DUMP,namrl,namr2   |,eof control              |,file#|,#files|||
  --                |,record format            |     --   --||
                    |                          --         --|
                    --                                      --
```

namrl — File name of existing file or non-disc LU; data is transferred from namrl.

namr2 — Name of existing file or non-disc LU to which data is transferred.

record format — Format of data being transferred; default is derived from namrl if namrl is a disc file; default is ASCII if namrl is a non-disc device.

eof control — SA to transfer end-of-file or subfile marks from namrl to namr2; IH to inhibit end-of-file on namr2 with subfile marks not transferred. If omitted, end-of-file is written at end of data on namr2 (refer to EOF control description below).

file# — Positive integer indicating file (or subfiles) relative to beginning of namr2 to which the beginning of data is transferred; default is 1.

#files — Positive integer indicating number of non-disc files or disc subfiles to be transferred; default is 1, unless namrl is a disc file and file# is omitted in which case default is 9999.

DU (Transfer Data to Existing File) . . . cont'd

*** NOTE ***

Only one place-holding comma is required when both record format and EOF control are omitted.

Files are transferred record by record; records longer than 128 words are truncated.

EXAMPLES:

1. Dump contents of MYFILE to left cartridge tape unit:

```
:DU,MYFILE,4
```

2. Transfer three files from magnetic tape and one file from right cartridge tape unit to disc:

```
:ST,8,A1,AS,IH,1,3     <-----Write 3 files from LU 8 as one file
                             on A1; inhibit EOF mark.
:DU,5,A1,AS,2          <-----Append 1 file  from right cartridge
                             tape unit and write EOF on A1.
```

3. Transfer three files from right cartridge tape unit to disc file WXY as three subfiles and then transfer the first two subfiles to magnetic tape as two files following an existing file on the magnetic tape.

```
:ST,5,WXY,SA,1,3    <----Transfer  3 files  from right cartridge
                         tape unit to WXY as subfiles.
:DU,WXY,8,SA,2,2    <----Transfer first two subfiles from WXY to
                         magnetic tape following existing file.
```

4. To add the contents of file B to the end of file A when both are existing disc files:

```
:DU,B,A,,2          <----File B is transferred to follow file A.
```

COMMENTS:

For copying files to or from Cartridge Tape Drives integrated with CS80 discs, use the FC program described in the RTE-6/VM Utilities Manual. If RECORD FORMAT is omitted, the file type of namrl is used to derive the format; if namrl is a non-disc device, the default record format is ASCII. Refer to the RTE-6/VM Programmer's Reference Manual for the formats. The choices for record format are:

AS ASCII records are transferred.

BA Binary absolute records are transferred; checksum is performed.

BR Binary relocatable records are transferred; checksum is performed.

BN Binary relocatable records are transferred; without checksum.

MT Magnetic tape ASCII records are transferred (MT record format is identical to AS record format).

MS Standard record formats are expected on namrl, magnetic tape SIO (System I/O) records are written to namr2.

Record formats can be combined as follows:

MSBR Magnetic tape SIO binary relocatable records.

MSBA Magnetic tape SIO binary absolute records.

If EOF CONTROL is omitted, an end-of-file mark is written on namr2 following the last data transferred; on paper tape, leader is punched at the beginning of the file as well as at the end. Any zero-length records on disc or embedded end-of-file marks on non-disc files are not transferred to namr2.

If specified, eof control is one of the following:

IH Inhibit the terminating end-of-file; useful only if namr2 is a non-disc file. On paper tape punch, inhibits initial leader.

SA Transfer embedded end-of-file marks or subfile separators from namrl to namr2.

When SA is specified, the embedded end-of-file marks are converted to the form used by namr2.

The FILE# specifies at which file or subfile on namr2 transfer begins. This feature allows you to append one file to another or to replace one file with another. For example, specify file# as 3 to skip the first two files or subfiles on namr2 and append a third file or subfile. Transfer is always from the start of namrl.

The #FILES parameter applies to files on non-disc devices or subfiles on disc. It specifies the number of files or subfiles to transfer starting with file#. File# and #files are specified or omitted under the following circumstances:

For a description of end-of-file marks and subfile marks refer to the ST command.

To transfer data to/from CS80 Cartridge Tape Drives, refer to the RTE-6/VM Utility Programs Reference Manual.

```
+--------+
| LEVEL  |
|   1    |
+--------+
```

EX (Terminate Session)

Initiates log-off process when operating under control of the Session Monitor; terminates the program FMGR when operating in a non-session environment.

```
+-------------------------------------------------------------------+
|                                                                   |
|       ,SP                                                         |
| EXIT ,RP [,RG [,KI ]]    <-----Session.                           |
|   --                                                              |
|                                                                   |
| EXIT                     <-----Non-session.                       |
|   --                                                              |
|                                                                   |
|-------------------------------------------------------------------|
|                                                                   |
| SP   Save private cartridges.                                     |
|                                                                   |
| RP   Release private cartridges.                                  |
|                                                                   |
| RG   Release group cartridges; default is not to dismount         |
|      group cartridges.                                            |
|                                                                   |
| KI   Abort any active session programs.                           |
|                                                                   |
+-------------------------------------------------------------------+
```

EXAMPLES:

```
   :EX,SP <-----------Log off saving private and group cartridges.

   :EX,RP,,KI <------Log off aborting any active session programs,
                     releasing private cartridges and saving group
                     cartridges.

   :EX,RP,RG <-----Log off releasing private and group cartridges.
```

COMMENTS:

When logging off a session, either SP or RP must be specified if the user has a private cartridge. If neither is specified, a FMGR 071 error will result. If the user does not have a private cartridge mounted, SP or RP is not required.

The system will log off the user if the session terminal times out five consecutive times (with time-out value greater than 20 seconds) without intervening input. A prompt is reissued to the terminal after each time out. If the user is logged off due to a time out, it will default to EX,SP,,KI. The time-out value of the terminal can be modified by the TO system command described in Chapter 4.

When logging off a session with the RP or RG option specified, FMGR will remove all ID segments of any restored programs (not restored segments) that reside on the cartridges being released.

Refer to Chapter 2 for a detailed description of the EX command when terminating a session.

```
+-------+
| LEVEL |
|   1   |
+-------+
```

HE (HELP Function)

Provides a detailed explanation of an error and guidance in possible corrective action.

```
+-----------------------------------------------------------------+
|                                                                 |
| HELP [,keyword [,lu ]]   <---Session                            |
|   --                                                            |
|                                                                 |
| HELP,keyword [,lu ]      <---Non-session                        |
|   --                                                            |
|                                                                 |
|-----------------------------------------------------------------|
|                                                                 |
| keyword   A select group of eight or less characters identify-  |
|           ing the error for which explanation is requested.    |
|           Under session its default is the last error posted   |
|           to the user's Session Control Block (SCB).  When not |
|           under session the keyword must be specified.         |
|                                                                 |
| lu        LU of device where explanation will be output.       |
|           Default is the user's terminal.                      |
|                                                                 |
+-----------------------------------------------------------------+
```

EXAMPLES:

```
   :HE <-------Displays explanation of last error posted to user's
               SCB on user's session terminal.

   :HE,FMGR-006,6 <----Display explanation of FMGR error code -006
                       on LU 6 (line printer).
```

COMMENTS:

All keywords and their corresponding explanations are contained in a disc-resident HELP file. A standard list of keywords provided with RTE-6/VM includes error messages of several systems and subsystems, as well as other useful information. A user can generate additional keywords and explanations for any interactive program using the GENIX and CMD utilities. For more details on these utilities, refer to the RTE-6/VM Utility Programs Reference Manual.

One of the features of the HELP function is that the System Manager can modify existing entries in the help file and/or create additional keywords. This feature is described in the RTE-6/VM System Manager's Manual.

```
+-------+
| LEVEL |
|  40   |
+-------+
```

IF (Conditional Skip)

Compares two values (usually globals) and skips a specified number of commands depending on the result of the comparison. This command cannot be executed from an interactive device; it must be within a procedure file or a batch job.

```
+---------------------------------------------------------------+
|                                                               |
| IF,pl,operator,p2 [,skip ]                                    |
|                                                               |
|---------------------------------------------------------------|
|                                                               |
| pl, p2     Values  to be compared;  one  or  both may be global |
|            parameters.                                        |
|                                                               |
| operator   Relative operator  used to compare values  of pl and |
|            p2;  entered  as one of the following  two-character |
|            operator abbreviations:                            |
|                                                               |
|            operator  operation                                |
|                                                               |
|              EQ      pl equals p2                             |
|              NE      pl does not equal p2                     |
|              LT      pl is less than p2                       |
|              GT      pl is greater than p2                    |
|              GE      pl is greater than or equal to p2        |
|              LE      pl is less than or equal to p2           |
|                                                               |
| skip       Skip count;  positive or negative integer specifying |
|            the number  of commands  to skip  when the result of |
|            the operation  between  pl and p2 is true.   Forward |
|            skip  if positive,  backward  if  negative.   If the |
|            result  is not true,  the next sequential command is |
|            executed.  If omitted, one command is skipped.      |
|                                                               |
+---------------------------------------------------------------+
```

EXAMPLES:

```
   :IF,1G,GE,2G,2 <---If the contents of global 1G is greater than
                      or equal to the contents  of global 2G, then
                      skip  the next  2 commands  in the procedure
                      file; otherwise, execute next command.

   :IF,1G,LT,5,-3 <---If the contents of global 1G is less than 5,
                      then  go back  2 commands  in the  procedure
                      file  and  execute that command;  otherwise,
                      execute next command.
```

COMMENTS:

The specified relation between p1 and p2 is examined, and if it is true then commands are skipped. One command is skipped if "skip" is not specified. If skip is specified, then that number of commands are skipped. A skip of -1 causes the IF command to be repeated. To skip back to the preceding command, specify a skip of -2. Note that comment lines in a procedure file are skipped as commands.

IF will not skip past the beginning or the end of the procedure file. Such an attempt will cause a skip to the beginning or end of file mark; no error message is issued.

When a negative skip is used, the file must be on a device that recognizes a backspace. For example, it is useless to attempt a negative skip on a paper tape reader. Jobs that are spooled recognize backspace commands.

The following relations hold for mixed types:

```
   null < numeric < ASCII
```

This corresponds to the type codes: null=0, numeric=1, ASCII=3.

If p1 and p2 are both ASCII, the comparison is based on the ASCII collating sequence.

For a discussion on procedure files refer to the Procedure Files section in this chapter. For a discussion on batch job command files refer to the RTE-6/VM Batch and Spooling Reference Manual.

LEVEL 60

IN (Initialize Cartridge)

Initializes a cartridge by defining an entry for the cartridge in the file directory maintained for each cartridge. The command can also be used to change this cartridge entry or to assign a new master security code to the system. For information on formatting a disc pack, refer to the RTE-6/VM Utility Programs Reference Manual.

```
IN,master security,cartridge,label,id[,first track[,#dir tracks
    [,sec/track ]]]

IN,master security--new security
```

*** NOTE ***

The first form initializes a cartridge or changes the description of an initialized cartridge. The second form changes the master security code of the cartridge.

master security — Security code that governs access to the FMP cartridge directory and to all file security codes; must be 2 ASCII characters; if omitted, file security codes can be accessed with any code or none. If using the disc cartridge initialization format under Session Monitor, the master security code is ignored. Refer to the comments below. Note that if the character "control E" is used in the master security code, the code cannot be changed until a "SWITCH" to a new system occurs.

cartridge — Cartridge identifier; specifies CRN, if ASCII or positive. Specifies LU, if negative. Must be negative the first time cartridge is initialized.

label — CRN that identifies the cartridge; must be positive integer from 1 through 32767 or two ASCII characters.

id — ASCII identifier assigned to cartridge; up to 6 ASCII characters specified exactly like an FMP file name.

IN (Initialize Cartridge) . . . cont'd

first track First FMP track on cartridge; a postive integer. For the system disc cartridge (LU 2), it must be at least 8 greater than the last system track; if omitted, 0 is assumed.

#dir tracks Number of directory tracks used by file directory on cartridge; positive integer from 1 through 48 for a 96 sector/track LU, or 1 through 36 for a 128 sector/track LU; if omitted, one track is assumed.

sec/track Number of 64-word sectors per track. If cartridge is on same channel (that is, the same hardware select code) as LU 2 or LU 3, this parameter is ignored. On any other channel, it is the minimum number of sectors/track which will be required on the cartridge. If greater than the actual sectors/track available, the cartridge will be initialized using all of the sectors on a track.

EXAMPLES:

1. New Cartridge Initialization:

```
:IN,,-14,9600,CLASYS <-----Initialize  cartridge  defined at
                           generation time as LU 14; and define
                           CRN  to  be  9600,   and   cartridge
                           identifier, CLASYS.
```

2. Re-initializing a Cartridge:

```
:IN,,9600,9700,NEWSYS <--Re-initialize  cartridge from previous
                         example changing CRN from 9600 to 9700
                         and  cartridge identifier  from CLASYS
                         to NEWSYS.
```

3. Change the system master security code:

```
:IN,SC--RT <---------Change master security code from SC to RT.
```

COMMENTS:

The system and system auxiliary disc cartridges (LUs 2 and 3) must be initialized the first time the RTE system is run after system generation. This process is described in the RTE-6/VM System Manager's Manual.

The master security code entered for the system is the code that controls all access to FMP files and cartridges. If specified, then that code must be used when initializing cartridges (except if under Session Monitor). Once specified it is important to remember the master security code since it is never printed or displayed by the system.

The master security code is ignored if initializing a disc cartridge in a session environment. A user may initialize any cartridge within his session capabilities. Thus, a user may initialize a private cartridge belonging to his session; a group manager (capability level of 63) may initialize a cartridge belonging to his group; the System Manager may initialize any cartridge. If the user does not have sufficient access privilege to the cartridge being initialized, a FMGR 046 error is displayed and the IN command is aborted.

For other cartridges (other than LUs 2 and 3) the MC command can also be used to initialize cartridges. One difference between the two commands is that the IN command requires that the cartridge be mounted, whereas, the MC command mounts the cartridge at the same time that it initializes it. Refer to the MC command for more information on its format and use.

Whenever an FMP cartridge is initialized the first time, "cartridge" must be a negative number specifying the LU with which the cartridge is associated.

Any cartridge, including LUs 2 and 3, can be re-initialized in order to change the initialization parameters. Before attempting to re-initialize a cartridge, however, all files on the cartridge must be closed. If any files are open, the FMGR-008 error message is issued, and the IN command is not executed. If any ID segments point to that cartridge, the FMGR 011 message is issued and the IN command is not executed.

If the first FMP track is lowered on the system cartridge (LU 2), it may lower the FMP area into the RTE system area. FMGR checks and if there is no conflict, assigns the tracks to FMP. But if the tracks requested for FMP conflict with the RTE system tracks, error message FMGR 059 is issued and the highest assigned track is reported. The IN command is not executed for this case. You may re-enter IN using the next highest track number as the first track.

Whenever the first FMP track is lowered, you must pack the disc cartridge with the PK command in order to recover the area. When the first track assigned to FMP is higher than the previous first track for the cartridge, the extra tracks are returned to RTE.

If the new parameters in a re-initialization raise the first track or lower the directory into an existing file, FMGR 060 is issued. This is a caution message that allows you to abort the initialization by entering either ?? or NO in response to the message. If you enter YES all files on the disc cartridge are purged.

Anyone knowing the master security code has access to all the file security codes on the cartridge. For this reason, it is never printed and if you know the code, you must remember it.

If the current master security code is zero (default if omitted), you must still enter any valid code in order to assign a new master security code. If the master security code is other than zero, you must enter the exact code in order to change it.

The new security code can be any two ASCII characters except:

colon (:)   comma (,)   a leading blank.

Non-printing characters are acceptable; in fact, such characters provide greater security since they are never printed or displayed.

To remove an existing master security code, the new security code can be set to two blanks. The blanks need not be specified; they are supplied by the parameter parsing routine.

```
+--------+
| LEVEL  |
|  10    |
+--------+
```

LI (List File Contents)

Lists the contents of a file, file directory information, or data stored on an LU to the list device.

```
+------------------------------------------------------------------+
|                                                                  |
| LIST,namr [,format [,L1 [,L2 ]]]                                 |
|   --                                                             |
|------------------------------------------------------------------|
|                                                                  |
| namr   File name or LU. If the file is protected by a negative   |
|        security code,  it must be specified.   If cartridge is   |
|        specified, that cartridge is searched for the file name,|
|        otherwise, the first found with that name is listed.      |
|                                                                  |
| format Specifies list format:                                    |
|                                                                  |
|          S     ASCII source format                              |
|          B     binary format                                    |
|          D     directory information only                       |
|                                                                  |
|        If omitted,  file type determines format:  S if file is   |
|        type 0, 3, or 4;  B for all other types.                  |
|                                                                  |
| L1, L2 Starting  and  ending line numbers of file being listed   |
|        in the specified format.  If neither L1 (starting line)   |
|        nor L2  (ending line)  are given,  the  entire file  is   |
|        listed.   If L1 is specified,  but not L2,  one line is   |
|        listed.  If L1 is greater than L2, no lines are listed.   |
|        If L1 is not specified,  but L2 is, L1 defaults to line   |
|        1 of the file.                                            |
|                                                                  |
+------------------------------------------------------------------+
```

EXAMPLES:

```
   :ST,1,AA <-----------------------Create file AA from terminal.
   FIRST RECORD FILE AA <-----------------------File AA contents.
   CNTL/D <-------------End input for file AA (CNTL/D) is input by
                        simultaneously  depressing  the CNTL and D
                        keys on the terminal).
```

```
:ST,1,BB <------------------------Create file BB from terminal.
FIRST RECORD FILE BB <------------------------File BB contents.
CNTL/D <--------------------------------End input for file BB.

:DU,BB,AA,,2 <------------------------------Dump file BB to AA.
```

1. Source Listing:

```
:LI,AA <--------------------S is default for type 3 or 4 files.

AA      T=00003 IS ON CR00002 USING 00001 BLKS R=0000

0001  FIRST RECORD FILE AA
0002  FIRST RECORD FILE BB
```

2. Binary Listing:

```
   :LI,AA,B

   AA      T=00003 IS ON CR00002 USING 00001 BLKS R=0000

REC# 00001

043111 051123 052040 051105 041517 051104 020106 044514*FIRST RECORD FIL
042440 040501                                           *E AA

REC# 00002

043111 051123 052040 051105 041517 051104 020106 044514*FIRST RECORD FIL
042440 041102                                           *E BB
```

3. Directory Listing:

```
:LI,AA,D

AA     T=00003 IS ON CR00002 USING 00001 BLKS R=0000
```

COMMENTS:

If a cartridge is locked, LI will skip that cartridge when searching for the file.

LI lists the specified file record by record. Any binary records longer than 128 words are truncated. Source records are truncated to 72 characters. On a teleprinter, the list starts in column one, on other list devices two blanks precede the list line.

HEADINGS

If namr is a file, the listing is headed by:

```
<file name> T= <file type> IS ON CR <cartridge> USING <file size>*
BLKS R= <record size>
```

where the lower-case words are replaced by the actual values in the file directory for the file (see examples). The asterisk following the file size is printed if the size of the file is in 128-block multiples. Note that this is the size of the main only, the extents are not shown by the LI command.

If namr is a LU, then a brief heading is printed with asterisks replacing the file name.

```
   ******T=00000 IS ON LU nn
```

where nn is the LU number.

DIRECTORY FORMAT

When D is specified, one of the headings shown above is all that is listed.

SOURCE FORMAT

When S is specified, each line number (1-9999) followed by a line of text is printed. Lines may not exceed 72 characters; longer lines are truncated.

BINARY FORMAT

When B is specified, the record number is printed followed by each word of the record. Words are printed in octal followed by an ASCII equivalent if a legal ASCII character corresponds to the octal. The ASCII is separated from the octal by an asterisk. Lines are truncated after the last non-blank character (asterisks are treated as blank characters in this case). Binary format prints eight words per line, using as many lines as are needed to print the record up to the maximum of 128 words.

For zero-length records, only the record number is printed.

```
+--------+
| LEVEL  |
|  20    |
+--------+
```

LL (Change List Device)

Changes current list device assignment.

```
+-----------------------------------------------------------------+
|                                                                 |
| LL, namr                                                        |
|                                                                 |
+-----------------------------------------------------------------+
|                                                                 |
| namr     New list device; may be either a file or an LU.        |
|                                                                 |
+-----------------------------------------------------------------+
```

EXAMPLES:

1. Change list device from user's terminal to line printer and back.

```
   :LL,6 <------Change list device from default of user's terminal
                to LU 6, the line printer.

   :LL,0G <-----Change list device back to user's terminal (global
                0G equals LU of terminal).
```

2. Change list device from user's terminal to file named LISTF and back.

```
   :LL,LISTF::1000 <---Change list  device from default  of user's
                       terminal to disc file,  LISTF, on cartridge
                       1000.

   :LL,0G <-----Change list device back to user's terminal (global
                0G equals LU of terminal).
```

COMMENTS:

The namr parameter may refer to any existing device or LU, however, it should be a device allowing output.

Certain FMGR commands (AN, LI, CL, and DL) direct their output to the list device. By default this is the user's terminal. If a printed copy is desired, the list device can be changed to the line printer as shown in the first example above.

LEVEL
50

LO (Change Log Device)

Changes current log device assignment.

```
LO,lu
```

lu	Specifies the LU of the new log device. Note that a file name cannot be used as a log device.

EXAMPLE:

To send a message from your terminal (system LU 17) to another terminal (system LU 18) the following commands can be entered:

```
:SL,18,18 <-------Place terminal,  where message is to  be sent,
                  into your SST so that it may be accessed.

:LO,18 <----------Change  log  device  from   your  terminal  to
                  terminal where message is to be sent.

:DP,MESSAGE <-----Display message on new log device.

:LO,0G <----------Change  log  device  back   to  your  terminal
                  (global 0G).
```

COMMENTS:

The LU specified in the LO command must be a two-way device such as a teleprinter or CRT terminal since it is used both to log messages and to correct errors.

All error messages are printed or displayed on the log device. When the log device is not the input device and an error occurs that requires operator correction, control is transferred from the input to the log device and corrective action must be taken at the log device. To transfer control back to the input device, simply type a colon after the colon prompt. The second colon is interpreted as a transfer command.

If the LO command is specified within a batch job, the default device (your terminal) is re-established at the end of the job.

LEVEL 10

MC (Mount Cartridge)

Makes an unmounted cartridge available to a user.

```
MC,lu [,P/G [,size [,id [,#dir tracks [,label ]]]]]
               \-------------v-------------/
              (Only used if there is not a valid
               directory on 1st directory track)
```

lu — LU of the cartridge to be mounted (can be positive or negative). If operating under session control, cartridge must first be in user's Session Switch Table (SST).

P/G — Private, group or non-session cartridge designation. When operating under session control, a private cartridge can be designated by typing a "P" or a group by typing "G" (default is private). When operating non-session this designation is meaningless, but its place must be provided for when specifying other optional parameters (refer to the example below).

size — The number of tracks to be used on the cartridge is specified with the size parameter. First track is always 0 and the last track is equal to size -1.

id — ASCII identifier assigned to the cartridge; up to 6 ASCII characters. When defaulted, the id will be DC00XX, where XX is the system LU of the terminal from which the MC command is input.

#dir tracks — Number of directory tracks used by the file directory on the cartridge (if omitted, 1 track is assumed); positive integer from 1 through 48 for a 96 sector/track LU, or 1 through 36 for a 128 sector/track LU.

MC (Mount Cartridge) . . . cont'd

label	CRN to be assigned to the cartridge being mounted (only applies to cartridges which do not have a valid directory at the specified track). If the cartridge already has a CRN associated with it, specification of a label will not be used (to change the CRN of a disc LU, refer to the IN command description).

EXAMPLES:

1. Operating under session, mount cartridge with LU 39 as a group cartridge.

```
:MC,-39,G
```

2. Operating non-session, mount cartridge with LU 40.

```
:MC,40
```

3. Operating non-session, mount cartridge with LU 41. LU 41 does not have a valid directory on track 201.

```
:MC,-41,,202,DATA,2,1250
```

(Note that even though it is unnecessary to specify the "P/G" parameter, its position must be maintained.)

4. Operating under session, attempt to mount LU 42 as a private cartridge.

```
:MC,42,P
CRN=XY
FMGR 012
```

The resulting FMGR 012 error (duplicate disc label) indicates that LU 42 has a CRN of XY, and a cartridge with that CRN is already mounted.

COMMENTS:

The MC command will initialize the cartridge only if there is not a valid directory on the first directory track. This depends on the size parameter if it is specified. For example, for a cartridge (LU 41) that is defined in the Track Map Table to have 203 tracks, an MC, -41 requires a valid directory on track 202. An MC, -41,,100 requires a valid directory on track 99. If the cartridge has a valid directory, specification of id, #dir tracks and label will not be used. Even though these parameters are ignored they must be valid parameters or an error will occur.

If the cartridge does not have a valid directory, the label parameter must be specified before the cartridge can be mounted. If it is not, an error is issued and MC terminates. The other parameters can be defaulted to their default values.

In the session environment, there are certain restrictions that the user should be aware of before mounting a cartridge. First, each user is restricted to having mounted at one time to his Session Control Block (SCB) only a specified number of private and group cartridges. The number is assigned by the System Manager and put into the user's account file entry. If the user tries to mount more than the specified number of cartridges, an FMGR 063 error results.

Second, there must be room in the user's Session Switch Table (SST) to post the LU number of the cartridge being mounted. Also, there cannot be another session LU currently in the SST that has the same LU as the cartridge being posted. These two conditions can cause FMGR 066 and FMGR 065 errors, respectively.

Third, if a user has a particular CRN on a cartridge he has mounted to his session as a private cartridge, he cannot mount a group cartridge to his session that also has that CRN assigned to it or vice versa. An attempt to do this will cause an FMGR 012 error. Also, a user cannot mount to his session a private or group cartridge with the same CRN as a system cartridge.

A system cartridge can be mounted using the MC command by the System Manager.

NOTE

> If an account in one group is linked to an account in another group, all users in both groups must be careful not to use duplicate CRNs when allocating (AC) or mounting (MC) cartridges.

```
+--------+
| LEVEL  |
|  10    |
+--------+
```

ME (Display Messages)

Displays contents of user's message file.

```
+-----------------------------------------------------------------+
| ME [,namr [,clear ]]                                            |
+-----------------------------------------------------------------+
| namr   File name  or  LU of non-disc device where messages are  |
|        to be sent (default is user's session terminal).         |
|                                                                 |
| clear  Optional parameter.  If set equal to 1 the message file  |
|        is cleared after listing it; if set to 0 does not clear  |
|        (default case).                                          |
+-----------------------------------------------------------------+
```

EXAMPLES:

```
1) :ME <---Displays contents  of message file to  user's terminal.
           File not cleared.

2) :ME,6,1 <---Displays contents  of message file on  line printer
               (LU 6) and clears file.

3) :ME,MYFILE,1 <---Save the  contents of  message file  in MYFILE
                    (file  created  by  ME  command),  then  clear
                    message file.
```

COMMENTS:

When a user successfully logs on, a message of "MESSAGES WAITING" will be displayed if there are any entries in the user's message file.

If a file name is specified for the namr parameter, the file will be created if it does not currently exist and the message file contents will be stored in it. If the file does exist, the message file contents will be dumped into it.

The ME command, like the SM command which sends messages, is valid only in the session mode.

```
+-------+
| LEVEL |
| 30/60 |
+-------+
```

OF (Terminate Program)

Immediately terminates a specified program. If the user has a capability level of 30 or higher, any program within the caller's current session can be terminated. If the user has a capability level of 60 or higher, any program in the system can be terminated. If the program is a temporary program loaded on-line, its ID segment is cleared and returned to the system.

```
+-----------------------------------------------------------------+
|                                                                 |
|  OFF, program                                                   |
|    -                                                            |
|-----------------------------------------------------------------+
|                                                                 |
|  program   Name of the program to be terminated.                |
|                                                                 |
+-----------------------------------------------------------------+
```

EXAMPLES:

```
1. :OF,APROG <------- remove APROG and its ID segment from system.

2. :OF,MAIN
   :OF,SEG1 <----------- remove program MAIN and its two segments.
   :OF,SEG2
```

COMMENTS:

The OF command clears the program's ID segment, if it was a temporary loaded program, and returns any disc tracks used by it to the system. The ID segment and tracks become available for use by another program. If executing, the program is also terminated. If the program is segmented, all segments must be removed with separate OF commands; an OF naming the main program will not remove its segments automatically.

The File Manager OF command is the same as the RTE command OF,program,8 (refer to Chapter 4). The message 'PROGRAM ABORTED' generated by OF,program,8 is also generated by OF,program.

Removal of a program with OF causes an abort message to be displayed at the system console unless the program is a background disc-resident segment.

Note that permanent programs created at generation or by the loader maintain their ID segments on the disc and can be removed only with the loaders. (Refer to the RTE-6/VM Loader Reference Manual for more details.)

```
+-------+
| LEVEL |
|  40   |
+-------+
```

PA (Pause and Send Message)

Suspends execution of the current job or procedure file, transfers control to the log device or some other specified device, and optionally displays a message. PA is used to send messages only during non-interactive processing.

```
+------------------------------------------------------------------+
|                                                                  |
| PAUSE [,lu [,message ]]                                          |
|   ---                                                            |
|                                                                  |
|------------------------------------------------------------------+
|                                                                  |
| lu         LU of device  to  which control transfers and where  |
|            message is displayed; default is the log device.     |
|                                                                  |
| message    Message  to  be  displayed  on lu;  must conform to  |
|            parameter  syntax  rules  for  privileged  commands  |
|            (refer to the Parameter Syntax Rules section).        |
|                                                                  |
+------------------------------------------------------------------+
```

EXAMPLE:

Use PA to issue a request to the session terminal to load a source program into the left cartridge tape unit. PA is part of the procedure file PROC that compiles, loads and runs the program.

```
  :TR,PROC <--------------------transfer to procedure file, PROC.

     :PA,,PUT MINICARTRIDGE CONTAINING SOURCE PROGRAM INTO LEFT CTU
     :ST,4,PROG
file :RU,FTN7X,PROG,0G,%PROG
PROC :RU,LOADR,,%PROG,0G
     :RU,10G
     :TR
```

When PA is executed, the following message is printed at the console:

```
:PA,,PUT MINICARTRIDGE CONTAINING SOURCE PROGRAM INTO LEFT CTU
```

The user then loads the minicartridge and enters TR to return to PROC at the command following PA.

COMMENTS:

The PA command causes a transfer to the log device or specified LU where the entire commmand line is printed. The message, if any, must conform to the syntax rules for any parameter.

When job processing is suspended with PA, it may be continued by entering a TR command on the device to which control transferred. Control returns to the command following PA.

```
+-------+
| LEVEL |
|  20   |
+-------+
```

PK (Pack Cartridge)

Recovers the tracks assigned to purged files and closes any gaps between files.

```
+--------------------------------------------------------------+
|                                                              |
| PK [,cartridge]                                              |
|                                                              |
|--------------------------------------------------------------+
|                                                              |
| cartridge  Cartridge identifier; a positive or ASCII CRN or, |
|            if negative, the LU of the FMP cartridge to be    |
|            packed.  If omitted, all private, group and global|
|            cartridges mounted to a user are packed.          |
|                                                              |
+--------------------------------------------------------------+
```

EXAMPLES:

```
   :PK,1000 <------- Pack cartridge mounted to user with CRN 1000.

   :PK <------------ Pack all cartridges mounted to user as a
                     private, group or global cartridge.
```

COMMENTS:

When operating under the Session Monitor, only the System Manager can pack LU 2 or 3. Any cartridges which the user specifies to be packed must be mounted as either a private or group cartridge to the user, or as a global cartridge (i.e., a system cartridge other than LU 2 or 3).

When PK executes, it moves files into empty spaces left from purging, if possible, and updates the file addresses in the file directory. When all files are packed, it then packs the directory removing any entries for purged files.

PK will purge all files that contain bad tracks reported by the IN command. If you do not want the file with bad tracks purged, you must save it on another cartridge using the ST command.

Since the PK command locks the cartridge being packed, you must be sure that there are no open files on the cartridge before requesting PK. If files are open, a FMP -008 error message is issued followed by the CRN or LU of the cartridge. If the cartridge being packed contains any ID segments pointing to type 6 files on that cartridge by the RP command, an FMGR 011 error is issued, followed by a list of the programs corresponding to these segments. ID segments must be removed with the OF command before packing can take place. If the PK command is executed from a transfer file, the cartridge on which the transfer file resides should not be packed from that transfer file. FMGR may abort if the transfer file is moved during the packing process.

NOTE

To pack a cartridge containing spool files, you must first shut down the spool system (refer to the RTE-6/VM Batch and Spooling Reference Manual).

If the system goes down during execution of PK, it is possible to lose, at most, one file. In order to determine which file, if any, has been lost, a copy of the file directory listing, before PK was entered, is needed. This list should be the complete list requested with the master security code so that is shows length, first track and sector addresses for each file. When power is returned, get another such list and compare the two lists. Look for the first file with an old disc address preceded by a file with a new address. Since the directory is updated after each file is successfully moved, the unchanged entry may indicate that the file has been lost. For example, the directory list before packing shows:

NAME	TYPE	#BLKS/LU	SCODE	TRACK	SEC	OPEN TO	
A	00003	00001			0100	000	
B	00004	00003			0100	002	
							← 1 block gap follows B
C	00004	00002			0100	010	
D	00003	00003			0100	014	

If the directory after packing shows C in sector 8 and D in sector 12, no files have been lost; PK has completed its operation. If, however, either file C or D is listed in the same sector (10 or 14, respectively), there is a good chance that it was being moved when the system went down and is now lost. If the second directory shows duplicate entries this simply means the directory was being re-written when the failure occurred and no files were lost. This can be corrected by storing the affected files in new files, purging the affected files, and changing the names of the new files back to their original names.

```
+-------+
| LEVEL |
|  20   |
+-------+
```

PU (Purge File)

Removes a file and its extents from the system. Type 0 files can be purged only with this command; other file types can also be purged with the programmatic PURGE call (refer to the RTE-6/VM Programmer's Reference Manual for details on the PURGE FMP call).

```
+----------------------------------------------------------------+
|                                                                |
| PURGE, file descriptor                                         |
|   ---                                                          |
|                                                                |
|----------------------------------------------------------------+
|                                                                |
| file           File name and, optionally, security code and CRN |
| descriptor     of existing file.                               |
|                                                                |
+----------------------------------------------------------------+
```

EXAMPLES:

```
1. :PU,AA <------------------ purge the first file found named AA.

2. :PU,CC:55:100 <---- purge file CC protected by security code 55
                       on cartridge 100.
```

3. Assume files A, B, and C are the last files in the directory:

```
   :PU,A
   :PU,B
   :PU,C
```

When C, the last file is purged, FMGR releases its disc space and then checks the next to last file, B. Since B has also been purged, its disc space is released and A is checked. It too was purged and FMGR releases its disc space. This procedure continues until FMGR finds a file that has not been purged.

COMMENTS:

When operating under the Session Monitor, any programs saved on LU 2 or 3 by the SP command can only be directly purged by either the person who saved the program, the System Manager, or a non-session user.

If a file is protected by a security code it cannot be purged unless the correct code is entered. If a label is specified, that cartridge is searched for the file to be purged; otherwise, cartridges are searched and the first file found with the correct file name is purged.

When a file is purged, the first word in its file directory entry is set to -1. Tracks assigned to the file are returned to the system only if the file was the last on the cartridge. When a type 6 program is purged, its tracks are released to the system in the same manner as any other file. All purged files except the last can be returned to the system only by packing the cartridge with the PK command, or by creating a file or a file extent that is the same size as a purged file.

A purged file cannot be accessed and its name will not appear on the file directory list requested with the DL command. A new file with the same name can now be created on the cartridge.

A type 6 file cannot be purged if an ID segment pointing to the disc space occupied by the type 6 file exists (the file has been RP'ed).

```
+--------+
| LEVEL  |
|  20    |
+--------+
```

RN (Rename File)

Renames an existing, but closed, disc file. The name, security code, and/or the file type may be changed with this command.

```
+-------------------------------------------------------------------+
|                                                                   |
|  RN,file descriptor,new file descriptor                           |
|                                                                   |
+-------------------------------------------------------------------+
|                                                                   |
|  file           File name and, optionally, security code and CRN  |
|  descriptor     of existing file.                                 |
|                                                                   |
|  new file       New file name, security code, and/or file type    |
|  descriptor     replace existing subparameters.                   |
|                                                                   |
|                 Note: No other subparameters can be changed. Any  |
|                       subparameters not specified will be left    |
|                       unchanged.                                  |
+-------------------------------------------------------------------+
```

EXAMPLES:

```
1. :RN,MYFILE:-25:100,MF <--- change the name MYFILE on CRN 100 to
                              MF; security code must be correct.

2. :RN,URFILE,FILEA <--- search cartridges for first file named
                         URFILE, and change its name to FILEA; the
                         file must not protected by a security
                         code.

3. :RN,FILEA:XX:50,FILEB:YY::4 <--- for FILEA on CRN 50, change
                                    the name to FILEB, the
                                    security code to YY, and the
                                    type to 4.

4. :RN,&TEST::RT,:GE <---- add security code GE to the file &TEST
                           on CRN RT; note the file name and type
                           remain unchanged.
```

COMMENTS:

The new name must be unique to the disc cartridge to which the existing file is allocated. If the file was created with a non-zero security code, then the file descriptor in this command must include that code. If a cartridge is included in the file descriptor, then only that disc cartridge is searched. If the cartridge is omitted, all mounted cartridges are searched and the first file found with the corresponding file descriptor is renamed. If operating with the Session Monitor, the search starts with the user's private cartridges, then group, then system. In a non-session environment, the search starts with the first non-session or system cartridge in the system cartridge directory.

If the new file name is the same as the existing file name, a duplicate file error (FMGR-002) will NOT occur as long as either the security code or file type (or both) is different.

The changing of a file's type is only possible if the internal file structure implied by the new type is the same as the existing file structure. This means that there are some restrictions in changing file types 0,1,2, and 6. These restrictions are:

Types 0 and 2 - may not be changed and no types may be changed to type 0 or 2 (restricted because these two types have unique internal structures matched by no other type).

Types 1 and 6 - may be interchanged, but no other types may be changed to type 1 or 6 (the internal structure of these two types match each other, but no other type).

All other types (3,4,5, and 7 & up) may be changed among themselves with no restrictions.

LEVEL
30

RP (Restore Program)

Restore a program by assigning an ID segment to a type 6 file, or release an ID segment previously assigned.

```
RP,file descriptor[,,pname] <--------Assigns new ID segment to
                                     "file descriptor".

RP,file descriptor,program[,pname] <-----Assigns ID segment of
                                         "program"  to  "file
                                         descriptor".
RP,,program <---------------Release disc tracks and ID segment
                            assigned by previous RP.
```

file descriptor   Identifies type 6 files that were saved with SP; security code, and cartridge may be specified. File descriptor is only omitted in format 3 to release the ID segment of the named program.

program   1-5 character name of program whose ID segment is assigned to file descriptor (format 2) or released (format 3). Its ID segment must have been created by a previous RP command.

pname   Optional 1-5 character name which the restored file descriptor is to have. If not specified, the program defaults to the first 5 characters of file descriptor.

EXAMPLES:

1. Restore program file APROG1 as program APROG:

```
:RP,APROG1 <--- file APROG1  is restored  as program APROG; its
                ID segment is stored in memory.
```

2. Restore program file TEST01 as program TEST0 and assign it APROG's ID segment:

```
:RP,TEST01,APROG,TEST0 <--APROG must be inactive.
```

3. Release ID segment and tracks previously assigned to TESTO:

```
:RP,,TESTO
```

4. Restore program file APROG1 as program F00:

```
:RP,APROG1,,F00
```

COMMENTS:

To execute a program saved as a type 6 file, the user can either restore the program with the RP command and then execute it or allow the FMGR RU command to automatically restore the program before execution. When the FMGR RU command restores the program, the ID segment and resources are released after the program terminates or aborts. The RP command can be used to restore program segments. However, only the segments that reside on LU 2 and LU 3 can be restored with the FMGR RU command. Refer to the description of the FMGR RU command for a further explanation.

Programs may be restored from type 6 files saved on any disc cartridge. If a CRN is not specified in the file descriptor, cartridges are searched in the order they appear in the user's cartridge list, and the first file that matches the file descriptor will be RP'ed.

Format 1

A program file, file descriptor, is restored as a program that can be accessed by the RTE system commands. It is restored with the same time parameters, priority, and partition assignment it had when saved. The name of the restored program will be "pname", if it was specified, otherwise the first five characters of the program file descriptor define the program name. If the file descriptor has been renamed since it was saved, then the new name is used.

If a program with the same name is already in the system, error message FMGR 023 is issued. To avoid this error, it is a good idea to delete any program saved by the SP command by using the FMGR OF, command or rename the type 6 file before it is restored.

All main programs restored with the RP command are temporary and are released when the user logs off. However, all segments restored with the RP command will remain after the user logs off. These programs are not recorded in the system area of disc and will not be restored automatically when the system is restarted with the bootstrap loader or when the user logs on in a later session.

If a blank ID segment cannot be found for the type 6 file, FMGR 014 is issued. An ID segment may be freed by deleting a program with the FMGR OF command or by using the "program" option of RP (formats 2 or 3).

Format 2

The restored program can be assigned the ID segment of another previously restored program, "program", by using the second format. The parameter "program" must identify a currently inactive program that was restored with RP.

RP first releases the ID segment assigned to "program" and then allocates a blank ID segment to file descriptor. RP then looks for a blank ID segment from a temporary program. If none are available, it uses a blank ID segment from a permanent program; such ID segments occur when a permanent program is purged by the interactive loaders. If the file descriptor is a program segment, short ID segments are used when possible. (Refer to the RTE-6/VM Loader Reference Manual.)

Format 3

The third format releases the ID segment and any assigned tracks of the program. The ID segment is returned to the system as a blank ID segment that can be used by another program.

If "program" does not exist and thus does not have an ID segment, FMGR 009 is issued but does not cause transfer to the log device. If "program" is currently active, FMGR 018 results. The user can enter "OF,program" from break-mode and try RP again in order to release the ID segment.

The restored program file cannot be purged while an ID segment points to it and while it is still running; a FMGR-037 error occurs. The cartridge where the type 6 file resides cannot be packed with the FMGR PK command until the ID segment pointing to the type 6 file is released. Failure to return ID segments before packing results in a FMGR 011 message.

```
+-------+
| LEVEL |
|  30   |
+-------+
```

RT (Release Tracks)

Releases disc tracks (on LU 2 and LU 3) locally assigned to a dormant program back to the system.

```
+------------------------------------------------------------------+
|                                                                  |
| RT,program                                                       |
|                                                                  |
|------------------------------------------------------------------+
|                                                                  |
| program     Name of program  whose assigned  tracks  are  to be  |
|             released; must be a dormant program.                 |
|                                                                  |
+------------------------------------------------------------------+
```

EXAMPLE:

```
  :RT,EDIT <--------- all LS tracks assigned to EDIT are released.
```

COMMENTS:

If the named program is dormant, all local tracks assigned to that program are released. Any released tracks become available to the system and all programs suspended and waiting for disc track allocation are rescheduled. If the named program is not dormant, the request is illegal. The error message 'ILLEGAL STATUS' is issued.

This command can be used to release any LS tracks assigned to the EDIT program. These tracks may accumulate through use of MS commands and also may be left when EDIT terminates leaving an LS area assigned. Refer to Appendix B for a discussion of LS tracks.

Disc tracks on LU 2 and LU 3 can be assigned locally to a program programmatically through an EXEC 4 command (refer to the RTE-6/VM Programmer's Reference Manual for details on EXEC calls). The RT command is equivalent to the EXEC 5 command except that it is interactive rather than programmatic.

```
+--------+
| LEVEL  |
|  30    |
+--------+
```

RU (Run Program)

Searches for and executes a named program.

```
+-----------------------------------------------------------------+
|                                                                 |
|       ,program                                                  |
| RUN   ,file descriptor [,parameters] <--- passes command string |
|  -                                                              |
|               or                                                |
|                                                                 |
|       ,program                                                  |
| RUIH  ,file descriptor [,parameters] <--- inhibits  passing  of |
|                                           command string        |
|               or                                                |
|                                                                 |
|       ,program:IH                                               |
| RUN   ,file descriptor:IH [,parameters] <-- inhibits  automatic |
|  -                                          renaming feature of |
|                                             RTE-6/VM            |
|                                                                 |
|-----------------------------------------------------------------|
|                                                                 |
| program    Program name;   5-character name  of program  to be |
|            executed.                                            |
|                                                                 |
| file       Identifies  type 6 file  containing  program  to be |
| descriptor executed.                                            |
|                                                                 |
| parameters Parameters  or  string  to be  passed  to  program; |
|            omitted if no parameters or string is to be passed.  |
|                                                                 |
+-----------------------------------------------------------------+
```

EXAMPLES:

1. Run program PROGA, inhibiting the automatic renaming feature of RTE-6/VM and passing the values 10, 20, 30, 40, 50 to PROGA.

   ```
   :RU,PROGA:IH,10,20,30,40,50
   ```

   The above command allows the user to run the actual program PROGA (not a copy of PROGA) and pass parameters to the program which can be picked up by calling the subroutine RMPAR (refer to the RTE-6/VM Library Manual for details on RMPAR).

2. Run program PROGB, passing a string to PROGB and not inhibiting the automatic renaming feature of RTE-6/VM.

   ```
   :RU,PROGB, I AM SENDING THIS MESSAGE TO PROGB
   ```

   The above command will run a copy of PROGB, and pass the command string, which is everything to the right of the FMGR colon prompt, to a buffer in System Available Memory (SAM). The string, I AM SENDING THIS MESSAGE TO PROGB, can then be picked up by PROGB by calling the RTE library subroutine GETST (refer to the RTE-6/VM Library Manual for details on GETST).

3. Run program PROGB, inhibiting the passing of the command string and inhibiting the RTE-6/VM automatic renaming feature.

   ```
   :RUIH,PROGB:IH, I AM SENDING THIS MESSAGE TO PROGB
   ```

   The above command allows the user to run the actual program PROGB (not a copy of PROGB) and not pass the command string so that no buffer in SAM need be created. No parameters are passed in this example.

COMMENTS:

When RU is executed, a search is made of all ID segments in the system for the named program. If found, the program is executed. If not found, a search is made for a type 6 file in which the named program is stored. If such a file is found, it is restored, its ID segment is built in memory, and the program is renamed and executed. In the case, it is unnecessary to specify RP; RU will insure that RP is performed. Following execution of a program restored by the system from a type 6 file, the program's ID segment is released and any tracks used by the program also are released. A segment can be restored automatically by the RU command (implicit RP) only if it resides on LU 2 or LU 3. If the segment resides on a peripheral cartridge, it must be restored with the RP command prior to running.

When parameters are specified for a program to be executed, the particular parameters depend on the program. For instance, the Macroassembler (MACRO) and the FORTRAN compilers have a standard set of parameters that are used when these programs are executed with RU. A program may pass back up to five one-word parameters with the routine PRTN. The five program parameters are passed back to FMGR in global parameters 1P through 5P. The first three parameters are also returned to FMGR as the global parameter 10G, a three-word ASCII parameter that usually contains a file name (refer to section Global Parameters).

Global parameters may be used within any of the RU command parameters. The global parameters are interpreted and correct values are passed through to the scheduled program.

When a program is scheduled using the RU command form, a section of System Available Memory (SAM) is allocated for storage of a command string. A command string consists of every item following the prompt character in a scheduling command entry.

When a program scheduled by FMGR passes a command string back to FMGR via an EXEC 14, FMGR will execute the string as the next command following termination of the program.

Passing of the command string to a buffer in System Available Memory allows the passing of strings to programs via the RU command. The RU command is a privileged command and conforms to the syntax rules for privileged commands. The string passage capability is described in detail in the RTE-6/VM Programmer's Reference Manual.

The following descriptions define system action when the program restored (implicit RP) via the FMGR RU command terminates:

1. If the running program terminates, is aborted, or terminates serially reusable (refer to the RTE-6/VM Programmer's Reference Manual for the description of serially reusable):

   a. The system releases program owned tracks, e.g., EDIT, LS, system scratch and compiler scratch tracks.

   b. The system releases the program's ID segment.

   c. The system releases program owned resource numbers, LUs, and program owned memory.

2. If the running program calls EXEC to place itself into the time list and/or terminates saving resources:

   a. The system does not release any resources.

   b. The program's ID segment may not be released by RP (the OF command can be used). An attempt to release the program's ID segment with the RP command will result in an error (FMGR 018).

   c. The program remains within the system.

3. If the running program calls EXEC to place itself into the time list and then terminates, or terminates serially reusable:

   a. The system releases program owned resource numbers, LUs, and program owned memory.

   b. The program's ID segment cannot be released by RP (the OF command can be used). An attempt to release the program's ID segment with the RP command will result in an error (FMGR 018).

   c. The program remains in the system.

In the RTE-6/VM environment, ID segments are managed so that each user can have his own copy of a program. If the user wishes to run a program with FMGXX as the father (i.e., :RU,PROGX but not :SYRU,PROGX), then in certain circumstances a copy of the program will be created belonging to the particular terminal and run for the user at the terminal. RTE-6/VM will perform this action whenever the program to be run is a permanently loaded or SP'd program and is a son of FMGXX. A copy of the program will be created with the last two characters being XX and scheduled for execution from the terminal with system LU XX. Temporary loaded programs are not automatically renamed.

For example, if the EDIT program is loaded on-line as a temporary load and saved as a type 6 file, the command:

```
   :RU,EDIT
```

will create a program named EDIXX and schedule it to terminal XX. When EDIXX is finished the ID segment will automatically be returned to the system.

The advantage of processing the ID segments in this way is that all terminals can run the same program so that each user gets his own copy of the program. Therefore, the user does not have to wait for other users to finish with a program before he can use it himself.

Even if the program to be run has been previously restored using the RP command, the above procedure will still work properly. In fact, the program will be created more quickly since there would be no disc search time before the program could be run.

The program renaming feature may be inhibited when the user runs a program. The following form of the RU command can be used:

```
   :RU,PROGX:IH
```

In this case, the actual program named PROGX will be run, and not a copy.

This ability is especially useful when loading permanent programs. The programs LOADR or MLLDR are the only programs that can load programs permanently into the system; a copy of the LOADR or MLLDR cannot do this. (Refer to the RTE-6/VM Loader Reference Manual for a description of these loaders.) Therefore, if the user is operating from FMGXX at terminal XX, the following command can be used to load a permanent program:

```
   :RU,LOADR:IH, .......
```

For a type 6 file with a positive security code, the security code and cartridge should be specified when using IH. If the type 6 file has a negative security code the inhibit renaming feature cannot be used.

```
   :RU,PROGX:1:2:IH
```

```
+-------+
| LEVEL |
|  40   |
+-------+
```

SE (Set Global Parameters)

Assigns values to the global parameters 1G through 9G.

```
+------------------------------------------------------------------+
|                                                                  |
|  SET [,pl [,p2 [... [,p9 ]]]]                                    |
|    -                                                             |
|                                                                  |
|------------------------------------------------------------------|
|                                                                  |
| pl through p9  Values  assigned  to globals 1G and 9G;  if all   |
|                parameters are omitted,  globals are nulled; if   |
|                any one parameter is omitted, the corresponding   |
|                global is unchanged.                              |
|                                                                  |
+------------------------------------------------------------------+
```

EXAMPLES:

```
1. :SE,,,,,FIVE <-------------------global 5G assigned ASCII FIVE.
   :SE,256,NEWFIL,AA,0 <------globals 1G  through 4G  are assigned
                               values; 5G through 9G are unchanged.
   :DP,1G,2G,3G,4G,5G <-------------display globals 1G through 5G.
     256,NEWFIL,AA,0,FIVE

2. :SE,4G <--global 1G is set to the value of 4G; in this case, 0.
   :DP,1G <-------------------------------------display global 1G.
     0
```

COMMENTS:

The value in pl is assigned to 1G, that in p2 to 2G, and so forth. By using SE alone with no parameters, all the globals 1G through 9G can be cleared (nulled). Individual globals can be nulled or set with the CA command. Any integer value between -32767 and 32767 or an ASCII value up to six characters can be assigned to pl through p9. If, however, you enter a global name (0G - 10G, 1P - 5P, 0S or 1S) as a parameter, the value of the specified global is assigned to the global corresponding to the parameter position.

```
+--------+
| LEVEL  |
|  10    |
+--------+
```

SL (Display Session LU Information)

Displays the corresponding system LU, Equipment Table (EQT) entry number and subchannel number for either a specified session LU or all session LUs in a user's Session Switch Table (SST).

```
+---------------------------------------------------------------+
|                                                               |
| SL [,lu]                                                      |
|                                                               |
|---------------------------------------------------------------|
|                                                               |
| lu  Session LU  for  which  linkage  information  is desired; |
|     default is to list information for all session LUs in the |
|     user's SST.                                               |
|                                                               |
+---------------------------------------------------------------+
```

EXAMPLES:

```
 :SL,14<--Display linkage information for session LU 14.

    SLU  14=LU  # 14 = E  1 S 5 D
     ^      ^          ^    ^   ^
     |      |          |    |   |
     |      |          |    |   --device status
     |      |          |    |
     |      |          |    --subchannel #
     |      |          |
     |      |          --EQT #
     |      |
     |      --System LU
     |
     --Session LU
```

```
:SL  <-----  Display linkage information for all session LUs in
             user's SST.

  SLU   1=LU  #  7 = E  7
  SLU   2=LU  #  2 = E  1
  SLU   3=LU  #  3 = E  1 S 6
  SLU   4=LU  # 29 = E  7 S 1
  SLU   5=LU  # 30 = E  7 S 2
  SLU   6=LU  #  6 = E  6
  SLU   7=LU  # 10 = E 10
  SLU   8=LU  #  8 = E  8
  SLU  14=LU  # 14 = E  1 S 5 D
```

COMMENTS:

The session LU information is always displayed on the user's terminal. Use of the File Manager LL Command will not switch the SL output to another device such as the line printer.

When operating in a non-session environment, the SL Command will provide EQT, Subchannel and device status information for any specified LU in the system. If the LU parameter is defaulted, information for all LUs defined in the system will be provided on the user's terminal.

```
+--------+
| LEVEL  |
| 30/50  |
+--------+
```

SL (Set Up Spool File for I/O Device)

Allows a user to interactively assign or allocate a spool file for spooling to an I/O device. Users with command capability levels of 50 or greater may also automatically schedule a specified program (PROG) when the spool file is closed with the 16-word string consisting of the spool set up buffer passed to it. For more details on Spooling, refer to the RTE-6/VM Batch and Spooling Reference Manual.

```
+---------------------------------------------------------------+
|                                                               |
| SL,lu,[file descriptor[,attribute[,outlu[,priority[,prog]]]]] |
|                                                               |
|---------------------------------------------------------------|
| lu          The LU which the program doing I/O, references. If |
|             the LU is not currently a session LU in the user's |
|             SST, this command will make an entry for it        |
|             providing that "outlu" is currently a session LU.  |
|             Disc LUs are not permitted when specifying this    |
|             parameter.                                         |
|                                                                |
| file        Name of existing file to be used as a spool file.  |
| descriptor  If omitted, system assigns a file from the spool   |
|             pool defined at spool set-up.                      |
|                                                                |
| attribute   Defines characteristics of the spool access for    |
|             the current session or job.                        |
|                                                                |
|             Any three of the following attributes codes can be |
|             combined in any order, without delimiters in order |
|             to override defaults. Refer to the comments below  |
|             for defaults.                                      |
|                                                                |
| NO  NOW; queue the file for outspooling now. If not given, the |
|     file will be queued (providing that an outlu is present)   |
|     when the CS,lu command is given or an SL,lu,[...] command  |
|     is given or in any case at the end of the job or session.  |
|                                                                |
| RE  READ; file is read only.                                   |
|                                                                |
| WR  WRITE; file is write only.  An EOF will be written at the  |
|     start of the file as part of set up (except see BO).       |
+----------------------------------------------------------------+
```

SL (Spooling) . . . cont'd

BO BOTH; same as RE WR except that EOF is not written. Note that if both RE and WR are coded, it is the same as BO and the EOF is not written.

WN WRITE NOW; same as WR plus NO. Writes EOF at start of file, except if RE also coded (see BO).

BU BUFFERED; file is buffered.

PU PURGE; file is to be purged on completion of outspooling (if file is not to be outspooled, it is purged when it is closed). This parameter is ignored if the file descriptor is not specified.

SH WRITE SPOOL HEADERS; allows reproduction when the file is outspooled to an outspool LU.

ST STANDARD FILE FORMAT; specifies that headers are not to be written. This is required if the file is to be positioned and re-read.

outlu Session LU number mapped into the corresponding system LU to which the spool file will be outspooled. This is the system LU associated with outlu and must have been specified as an outspool destination LU during GASP initialization. If omitted, the outspool file will not be output to a device when the spool file is closed .

priority This is the outspool priority of the program's output. The default is 99 if the program is scheduled interactively from a session terminal.

prog Program "prog" will be immediately scheduled by the spool system when the spool LU is closed. The 16-word Spool Setup buffer will be passed to the program when it is scheduled. Note the spool file will not be outspooled, and it is the program's responsibility to properly dispose of the file.

EXAMPLES:

1. While in session, spool all output to the line-printer:

```
:SL,6,,,6
```

When the user logs off, the spool file will be closed and the output will the spooled to the line printer (LU 6).

2. In a session, dump file ASC to the line printer via spooling:

```
:SL,20,ASC,,6 <----------------------Set up spool equivalence.
:CS,20 <----------Close spool LU (refer to the FMGR CS command.
```

3. Use spooling to perform formatted writes to a disc file:

```
   PROGRAM WRSPL
   WRITE (45,10) <----------------Program does formatted write.
10 FORMAT ("HI THERE")
   END

:SL,45,"TEXT,WR <----Equivalence session LU 45 to existing file
                     "TEXT.
:RU,WRSPL <----------------------Run program to write message.
```

The file "TEXT now contains the ASCII string "HI THERE" as its first record.

4. Use spooling to perform a formatted read from a disc file:

```
   PROGRAM RESPL
   DIMENSION IST(39)
   READ (45,10) IST <----FORTRAN program to read a 78 character
                         message into array IST.
10 FORMAT(39A2)
   END

:SL,45,"TEXT <-------Equivalence session LU 45 to existing file
                     "TEXT.
:RU,RESPL <--------------------------------------Run program.
```

Array IST in program RESPL now contains the ASCII string "HI THERE".

5. Set up a spool file in a session. When the spool LU is closed, have the spool system schedule program GETIT to process the file with subroutine PROC:

```
PROGRAM GETIT
INTEGER SETUP(16)
CALL EXEC(14,1,SETUP,16) <-------------Retrieve 16-word buffer.
CALL PROC(SETUP)
END

:SL,20,TEXT,,6,,GETIT <---Set up spool equivalence  of existing
                          file TEXT and session LU 20. Schedule
                          program GETIT  when the spool file is
                          closed.
:LL,20 <--------------------------Change list device to LU 20.
:DL <----------------------------------Do a directory listing.
:CS,20 <--------Close spool LU.  Program GETIT is automatically
                scheduled and processes the file.
```

COMMENTS:

Logical Unit Assignment

The system associates a spool LU with the specified LU. The spool LU associated with the specified LU can be retrieved by displaying the global parameter 0S with the FMGR DP command.

The LU specified should correspond to the device type of the actual device for which it is intended (outlu). This is used to set up the device type to ensure that control functions and control requests are issued to the device as expected. For example, if output is to be sent to the line printer, LU 6 associated with the line printer should be specified. If the LU specified for the parameter lu has not been associated with a particular device, magnetic tape is assumed.

The LU must not be any of the following:

* LU 2 (system disc), or LU 3 (if assigned as auxiliary disc).

* any LU associated with a disc driver (DVR30, 31, 32 or 33).

* if in a job, system LU 5 (standard spool input device). If you want to use the true system LU 5, use an SL switch command (refer to the version of SL command describing modification of a user's SST).

* a spool LU.

Spool File Assignment

If file descriptor is not specified, an available file from the spool pool files (SPOL01-SPOL80) is associated with the LU. If a particular file is specified, then that file is associated with the parameter lu and will be used as a spool file during spooling. The specified file must be an existing user file; SL does not create the file. The name of the most recently assigned spool file can be retrieved by displaying the global parameter 1S with the FMGR DP command.

Attributes

Defaults for the attribute parameter are as follows:

	SPOOL POOL FILE	USER FILE
OUTLU	WR HO SH SP	WR HO SH SA
NO OUTLU	BO HO ST SP	RE HO ST SA

WR=WRITE ONLY
RE=READ ONLY
BO=BOTH READ & WRITE
ST=STANDARD FILE
SH=OUTSPOOL HEADERS
SP=SPOOL POOL FILE
HO=HOLD TILL CLOSE
SA=SAVE(DON'T PURGE)

Note that in all cases no buffering and hold (until close) are defaulted.

If the specified LU is to be used for output only or for output as well as input, then its attributes must be specified.

Unbuffered requests save System Available Memory (SAM) space. Generally, there is no need to request buffering with BU. Spool file headings are used rather than standard format (ST) if you want to read back the file either within the job or after the job. If the file is to be outspooled only, it is better to use the spool headers format. If ST is specified, no control is passed through to the outspool routine (SPOUT). In this case a default EOF action based on the actual device to which the file is being outspooled is supplied.

If you do not want to save the user-defined spool file namr, you must specify PU. Spool pool files are never saved. The normal case is to hold the spool file for outspooling until the spool is closed. If you specify NO, the file will be placed in the outspool queue immediately.

The SL hold attribute is not the same as the GASP hold, established with the GASP CS command. The SL hold prevents the file from being placed in the outspool queue (an entry is not made for it in SPLCON) until a CS command removes the hold or the session or job terminates. The GASP hold places a hold on a file already entered into the outspool queue (already listed in SPLCON), and may only be removed with a subsequent GASP command.

Outspool Logical Unit Assignment

The device to which output from spooling is to be directed must be specified with the outlu parameter, or the file will not be outspooled. The specified LU is not associated with a spool file or spool LU. Any outspool LU specified at GASP initialization may be used. An octal function code that specifies an I/O function code in bits 6-10 and an LU in bits 0-5 can be used when a file with standard format is output; files with spool header format use the function code in the second header word. For example, the code 104B indicates punch binary on LU 4. The function code is equivalent to the CONWD parameter in the standard I/O EXEC calls (refer to the RTE-6/VM Programmer's Reference Manual).

For spool header file format, if the file is to be outspooled, be careful that the proper attribute is used to indicate the file type SPOUT will actually see.

If outspool format is specified for a spool file, the validity test performed by SPOUT will usually find the error and change it to standard format. However, one or more records may be transmitted before the test fails.

```
+--------+
| LEVEL  |
| 30/50  |
+--------+
```

SL (Modify Session Switch Table)

Defines an LU mapping which is to be entered or deleted from a user's Session Switch Table (SST). With a command capability level of 30 or greater, a user may map a new session LU to a system LU currently within the user's SST. With a command capability level of 50 or greater, a user may add a system LU to his SST with this command. A user may only delete LU mappings which have been created during his session.

```
+------------------------------------------------------------------+
|                                                                  |
| SL,session LU,system LU  <-----  defines LU mapping in SST       |
|                                                                  |
| SL,session LU,-          <-----  deletes LU mapping in SST       |
|                                                                  |
|------------------------------------------------------------------+
|                                                                  |
| session LU  LU by which session users address a system LU. Can   |
|             be the same as the system LU, but is only required   |
|             to be so for disc cartridges.  Session LUs must be   |
|             unique within any user's SST.                        |
|                                                                  |
| system LU   LU by which  the  system  addresses  a physical or   |
|             logical device.  System LUs  are defined at system   |
|             generation time.                                     |
|                                                                  |
+------------------------------------------------------------------+
```

EXAMPLES:

```
   :SL,7,10 <--- Adds system LU 10  to the user's SST  and assigns
                 it session LU 7.

   :SL,7,- <--- Deletes entry for session LU 7 for the user's SST.
```

COMMENTS:

The System Manager defines each user's SST configuration at account set-up. The SST configuration resides in the Account File on disc and can only be changed by running the account program.

At account set-up time the System Manager also defines the number of additions (new LU mappings) a user can make to his SST with the SL command. An attempt to exceed this limit will result in an FMGR 066 error.

See Chapter 2 for a description and purpose of the Session Switch Table (SST).

```
+-------+
| LEVEL |
|  10   |
+-------+
```

SM (Send Message to Session User)

Sends a file and/or message to a specified session user's message file.

```
+------------------------------------------------------------------+
|                                                                  |
| SM,user,namr,message                                             |
|                                                                  |
+------------------------------------------------------------------+
|                                                                  |
| user      Log-on identification  of user to whom message is to  |
|           be sent.                                               |
|                                                                  |
| namr      Name of existing file  or LU of a non-disc device (a  |
|           positive integer) where data to be sent is stored.     |
|                                                                  |
| message   Message string entered from your terminal,  directed  |
|           to and stored in the specific user's message file.     |
|                                                                  |
+------------------------------------------------------------------+
```

EXAMPLES:

1.
```
  :SM,SMITH.ACCTG,INVENT::1000 <--passes inventory data to another
                                  user in another group.
```

2.
```
  :SM,SMITH.ACCTG,,WHERE ARE YOUR COST ESTIMATES <--passes literal
                                                    message to an-
                                                    other user.
```

3.
```
  :SM,SMITH.ACCTG,8,HERE IS THE DATA THAT YOU REQUESTED
```

(Sends contents (first file) on LU 8, magnetic tape, to another user and precedes it by a message.)

COMMENTS:

When the user who is to receive the message logs on, he will be informed of any messages pending in his message file ("MESSAGES WAITING").

When an LU is the source of a message file (e.g., magnetic tape or paper tape) the first file will be read, terminating at an EOF. When a file is entered from a terminal, EOF is acknowledged by entering a control D. A message string is terminated by a carriage return.

Stored with the file is a header, indicating the sender of the message, and the time it was sent.

There is no limit to the size of the message, other than the available room on the cartridge containing the message file.

If both a file and a message string are sent to a user, the message string will precede the file in the user's message file.

LEVEL 30

SP (Save Program)

Saves a disc-resident program as a type 6 FMP file.

```
                  /,PR
SP,file descriptor[     [,cap]]
                  \,GR
```

file descriptor File name of the type 6 file which is created by the SP command. The first five characters of the file name must be identical to the name of the saved program. File descriptor subparameters default to:

security code	default to 0
cartridge	default to first cartridge in cartridge list
file type	forced to type 6
file size	forced to size of program
record size	forced to 128

PR If specified, only the user who issued the SP command (or users linked to this account) will be allowed to run or RP the program from the type 6 file.

GR If specified, only users belonging to the same group as the user who issued the SP command will be allowed to run or RP the program from the type 6 file.

If PR or GR is not specified, any user in the system with a minimum capability of 30 with access to the type 6 file, can run or RP the program.

cap An integer representing the minimum capability level required to run or RP the program from the type 6 file. Note that the user must be able to invoke the RU or RP command (30), regardless of the capability specified in the SP command.

EXAMPLES:

1. Save a temporary disc-resident program APROG as a type 6 file named APROG1:

```
:SP,APROG1::CS <---save APROG on cartridge CS and its ID segment
                   as a file named APROG1.

:OF,APROG <---------------delete the original program from disc.
```

2. Save a permanent program PROG as a type 6 file named PROG, then purge the permanent program.

```
:SP,PROG <----save PROG and its ID segment as a file named PROG.

:RU,LOADR:IH <---Run interactive  loader;  renaming  feature  is
                 inhibited because  only the original version of
                 LOADR or MLLDR can  create  or purge  permanent
                 programs.    (Refer  to  the  RTE-6/VM   Loader
                 Reference Manual for more details.)

/LOADR: OP,PU<--purge option specified.

  /LOADR: PNAME ?PROG <----------------------------- purge PROG.
```

COMMENTS:

A program can be saved with the SP command or restored with the RP command to and from any disc cartridge.

The system interprets only the first five characters of a program name. A file name can be six characters. This means a type 6 file, for more than one version of the same program, can be created by adding one character to a five character program name. If, however, the program name is less than five characters, the file name must be identical.

The SP command is usually used in conjunction with the RP command. A program that has been edited, compiled or assembled, has been loaded and then run successfully can be saved in its loaded form for future use with the SP command. To run this program through FMGR, issue either the RU command alone or RP and RU. In either case, the file is restored and then executed.

Segmented programs can be SP'ed to any disc cartridge, but can only be restored automatically by the FMGR RU command (implicit RP) if saved on LU 2 or LU 3. When a segment is saved on a peripheral disc cartridge, it must be restored with the RP command prior to running.

One reason to save a program with SP is to release its ID segment for use by another program. The two examples above demonstrate the procedure used to SP either a temporary or permanent program and release its ID segment by using either the FMGR OF command or by purging it with the loaders.

Temporary programs created during a session are automatically OF'ed by the Session Monitor when the user logs off and the program's ID segment is released back to the System (if the program has segments appended to it, these will not be purged and their ID segments will not be released at log off). If the user wishes to save temporary programs when logging off, an SP command must be issued first.

```
+--------+
| LEVEL  |
|  20    |
+--------+
```

ST (Transfer Data and Create File)

Transfers data from a file or LU to a disc file or LU; the receiving file is created by the command unless it is an LU. To transfer data to an existing file use the DU command. For copying files from disc cartridge to or from tape devices, the FC program, described in the RTE-6/VM Utilities Manual, can also be used.

```
+------------------------------------------------------------------+
|                     --                                        -- |
|                     |                            --         --| |
|                     |,record format,eof control|    --     --|| |
| STORE,namrl,namr2|,eof control             |,file#|,#files||| |
|   ---               |,record format           |    --     --|| |
|                     |                            --         --| |
|                     --                                        -- |
|------------------------------------------------------------------|
| namrl    File name of existing file or non-disc LU; data is     |
|          transferred from namrl.                                 |
|                                                                  |
| namr2    File name or LU to which data is transferred from      |
|          namrl; if a file name, the file is created using the   |
|          last three namr subparameters if supplied.              |
|                                                                  |
| record   Format of data in namrl; default is derived from file  |
| format   type of namrl or is ASCII; refer to record format      |
|          description below.                                      |
|                                                                  |
| eof      SA to transfer embedded subfile marks in namrl to      |
| control  namr2 and write end-of-file mark at end of data on     |
|          namr2.  IH to inhibit end-of-file on namr2; subfile    |
|          marks are not transferred.  If omitted, end-of-file    |
|          marks are saved, subfile marks purged.                  |
|                                                                  |
| file#    Positive integer indicating file (or subfile)          |
|          relative to beginning of namrl at which to start       |
|          transfer; default is 1.                                 |
|                                                                  |
| #files   Positive integer indicating number of non-disc files   |
|          or disc subfiles to be transferred; default is 1       |
|          unless namrl is a disc file and file# is omitted, in   |
|          which case default is 9999.                             |
+------------------------------------------------------------------+
```

ST (Transfer Data and Create File) . . . cont'd.

*** NOTE ***

Files are transferred record by record; records longer than 128 words are truncated.

Only one place-holding comma is required when both record format and eof control are omitted.

EXAMPLES:

1. Create file AA and copy contents of disc file XYZ into it.

```
:ST,XYZ,AA
```

2. Copy a file from cartridge 2 to cartridge 17.

```
:ST,XYZ::2,XYZ::17
```

3. Create and store ASCII data in a file, FILEX, from a terminal.

```
:ST,1,FILEX

     .  <--- enter lines of ASCII data; terminate each line
     .       with a carriage return.
     .

CNTL/D <---------- press CNTL and D keys simultaneously to
                   terminate input (Control D).
```

4. Copy first five subfiles on disc file DFILE to magnetic tape as five separate files.

```
:ST,DFILE,8,MT,SA,,5
```

5. Transfer a binary relocatable file from minicartridge to disc. Data is input from cartridge tape unit LU 4 and stored in a file created by the ST command called BINRE.

```
:ST,4,BINRE,BR
```

COMMENTS:

namrl can be a created file (disc or non-disc) or an LU defined for the system.

namr2 can be an LU or a disc file name; it may not name a type 0 file. The file type subparameter for namr2 defaults to the file type of namrl if namrl is a disc file.

If namrl is not a disc file, the file type of namr2 is based on the record format of namrl:

type 3	if record format is MT, MS, or AS (ASCII or System I/O) or is omitted.
type 5	if record format is MSBR or BR (binary relocatable).
type 7	if record format is MSBA or BA (binary absolute).

Type 6 files are normally created with the SP command; type 4 and 5 files with the EDIT and FORTRAN Compiler or the Macroassembler. These files can be stored to new files with the ST command.

The file size subparameter for namr2 defaults to 24 blocks (4096 bytes) when namrl is a device, otherwise file size defaults to size of namrl without extents. The size of namr2 will be truncated to the actual number of records used in namrl, when namrl is a device or a disc file. If the file is empty, namr2 will not be created.

Any records greater than 128 words are truncated in namr2.

Record Format

If record format is omitted, the file type of namrl is used to derive the format:

if file type is 0 (non-disc files) 3 or 4 (disc files) record format is AS (ASCII).
if file type is 5, then record format is BR (binary relocatable).
if file type is 7, then record format is BA (binary absolute).

The choices for record format are:

AS	ASCII records are transferred.
BA	binary absolute records are transferred; checksum is performed.
BR	binary relocatable records are transferred; checksum is performed.

MT magnetic tape ASCII records are transferred (MT record format is identical to AS record format).

MS magnetic tape SIO (System I/O) records are expected on namrl, standard records are written to namr2.

Record Formats can be combined as follows:

MSBR magnetic tape SIO binary relocatable records

MSBA magnetic tape SIO binary absolute records.

Subfiles

Before discussing the remaining parameters, it is necessary to understand subfiles. On non-disc devices, files are separated by end-of-file marks that depend on the device:

magnetic tape	EOF mark
paper tape	leader
line printer	top-of-form indicator (page eject)
terminal	two spaces (two carriage return/line feeds) on output; CTRL/D on input.
minicartridge tape	carriage return/line feed.

If namrl is a non-disc device, it may contain many such subfiles: the physical end-of-file for the particular LU (device) terminates the collection of files or information. A disc file is terminated by a special end-of-file, -1, in the first length word of the last record. Disc files may be divided into subfiles by zero length records, two length words set to zero. These are logical divisions that usually result from transferring non-disc files to disc. (Refer to the RTE-6/VM Programmer's Reference Manual for record formats.)

Subfiles are useful when you want to save more than one tape file or relocatable program file (such as a library) on a single disc file and retain the files as separate entities. Subfiles can subsequently be stored (ST command) or dumped (DU command) to another file separately or as a single file (see Figure 3-2).

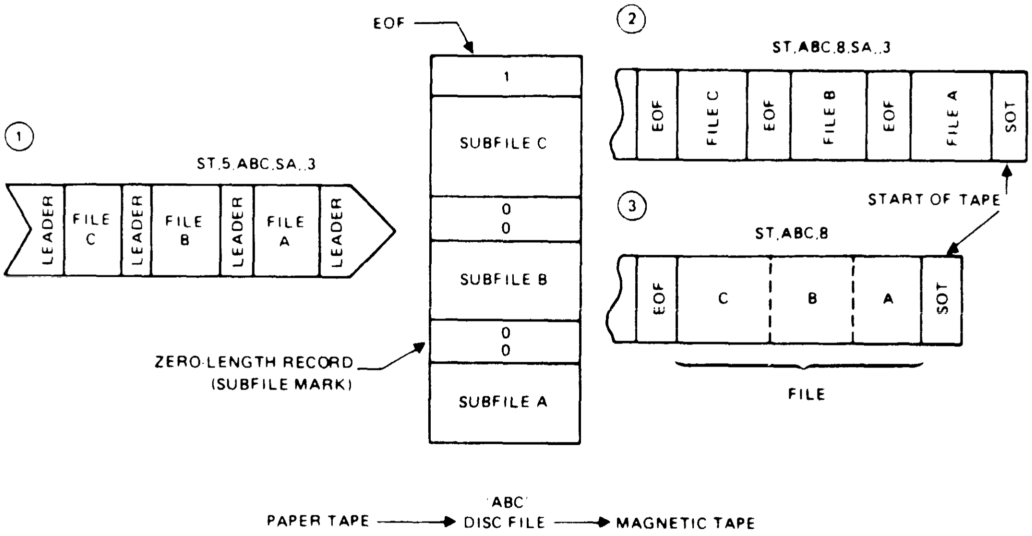


Figure 3-2. Relation of Files to Subfiles

EOF Control

If EOF control is omitted, an end-of-file mark is written on namr2 following the last data transferred; on paper tape, leader is punched at the beginning of the file as well as at the end. Any zero-length records on disc or embedded end-of-file marks on non-disc files are not transferred to namr2 (see (3) in Figure 3-2).

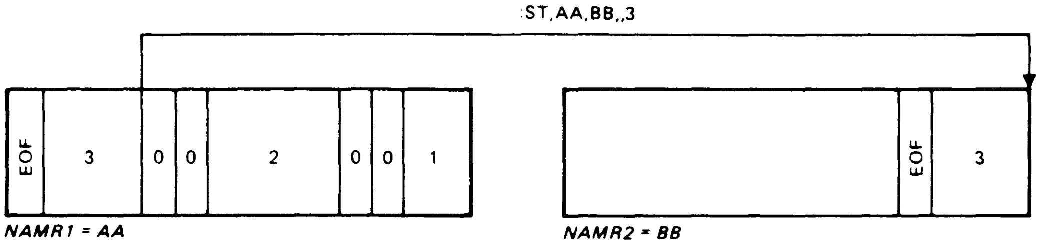
If specified, EOF control is one of the following:

IH	inhibit the terminating end-of-file; useful only if namr2 is a non-disc file. On paper tape punch, inhibits initial leader.
SA	transfer embedded end-of-file marks or subfile separators from namr1 to namr2

When SA is specified, the embedded end-of-file marks are converted to the form used by namr2. For instance, if a paper tape file is being stored on disc, the leader is converted to zero-length records (see (1) in Figure 3-2) and zero-length records to EOF marks on magnetic tape (see (2) in Figure 3-2).

File Number

The file# parameter applies to files on non-disc devices or subfiles on disc. It specifies which file or subfile relative to the first with which to start the transfer. For instance, if file# = 3, transfer starts with the third file or subfile. If omitted, transfer starts at the beginning of namrl. Transfer is always to the beginning of namr2.



FOR ST, *FILE#* SPECIFIES FILE OR SUBFILE IN *NAMR1*.

Number of Files

The #files parameter applies to files on non-disc devices or subfiles on disc. It specifies the number of files or subfiles to transfer starting with file#. File# and #files are specified or omitted under the following circumstances:

Transfer:	file#	#files
1 particular non-disc file or the rest of a disc file	yes	no
1 non-disc file or all of a disc file	no	no
a number of non-disc files or disc subfiles from beginning of file	no	yes
a number of non-disc files or disc subfiles from particular file or subfile	yes	yes

To transfer data to/from the Cartridge Tape Drives integrated with the CS80 discs, use the FC program described in the RTE-6/VM Utility Reference Manual.

```
+-------+
| LEVEL |
|  20   |
+-------+
```

SV (Change Severity Code)

Allows the user to change the severity code currently being used by FMGR.

```
+-----------------------------------------------------------+
|                                                           |
| SV,severity [,global# [,IH ]]                             |
|                                                           |
|-----------------------------------------------------------|
|                                                           |
| severity  New severity code; it may be:                   |
|                                                           |
|           0 Display  error  codes  and  echo  commands on log |
|             device (default).                             |
|           1 Display  error  codes  on  log  device,  inhibit |
|             command echo.                                 |
|           2 Error  code  displayed  only  if  error  requires |
|             transfer of control to log device for correction, |
|             in this case, any active batch job terminates; no |
|             command echo.                                 |
|           3 Same as 2  except  active job not terminated when |
|             an error causes transfer  to log device.  You can |
|             transfer back to the job. No command echo.    |
|           4 If  an  FMGR  command  error  is  encounterd, job |
|             continues  automatically;  no  command  echo,  no |
|             transfer to log device,  and no job abort occurs. |
|             The BR command will not stop a job.           |
|                                                           |
| global #  Optional  G global number in the range 1-9, in which |
|           the original severity code is placed.           |
|                                                           |
| IH        Optional parameter to inhibit echo of command entry. |
|                                                           |
+-----------------------------------------------------------+
```

EXAMPLE:

Create a transfer file which displays a message to the log device; changes the severity code to a 2 storing the original severity code in global 9G; inhibits the echo of command entries; and resets the severity code to the original severity code at the end of the transfer file.

```
:RU,EDIT <--------------------------------------- run EDIT/1000.
EDI62 : Use ? for help
FI, namr specifies file to edit.
EOF
/ :SV,2,9,IH
/ :DP,MESSAGE FROM TRANSFER FILE
/ :SV,9G,,IH
/ :TR
/EC/EXMPL <------------ end edit; create transfer file /EXMPL.
END OF EDIT
```

Refer to the EDIT/1000 User's Guide for a description of the interactive editor. The original severity code is 0 (default case). Transferring to the transfer file:

```
:TR,/EXMPL <--------- transfer control to transfer file /EXMPL.
 MESSAGE FROM TRANSFER FILE <------------- output to terminal.
 :TR <--------------------------------------- command echo.
```

In the above example, echo of the two SV commands is inhibited by the "IH" option. The DP command echo has been inhibited by the severity code of 2. The TR command was echoed back to the terminal because the severity code was changed back to a 0, the original severity code.

COMMENTS:

The normal default mode is to echo command as it is entered on the input device and to log all errors on the same device. During interactive operation, the severity code should be this default value, zero. When there is no advantage to echoing commands at the console, for instance when commands are entered in a batch job, from a peripheral device, or through a file, the severity code can be set to 1. If, in addition, it is desired to suppress messages unless they require action, the code can be set to 2. This code terminates any currently executing job when an error occurs so that the job will not continue with errors. A severity code of 3 allows jobs to continue in spite of errors. This code would be useful when more than one batch job must be processed so that the job containing errors does not hinder subsequent job processing. If SV is specified within a batch job, the severity is reset to zero when the job is terminated with the EO command.

Whenever command echo is suppressed, any command causing an error is displayed preceding the error code unless the error display is also inhibited.

```
+-------+
| LEVEL |
|   1   |
+-------+
```

SY (Execute RTE System Command)

Allows execution of system commands from FMGR. The following commands: FL, HE, RS, SL, TE, and WH are not allowed; BR and OF require that a program name be supplied. Refer to Chapter 4 for a description of system and break-mode commands and their command capability level requirements.

```
+-----------------------------------------------------------------+
|                                                                 |
| SYcommand                                                       |
|                                                                 |
|-----------------------------------------------------------------|
|                                                                 |
| command    The system command mnemonic code.   No delimiter is |
|            permitted between SY and the command.                |
|                                                                 |
+-----------------------------------------------------------------+
```

EXAMPLES:

```
  :SYUP,6 <---------------------Make LU 6 available to the system.

  :SYTI <-------------- Request the time and date from the system.
```

COMMENTS:

FMGR strips off the SY prefix and passes the remaining characters to the system for execution. Any messages resulting from execution of the system command are passed back through FMGR to the log device, but do not force a transfer to the log device.

If you do not specify an LU in an SY command request for the system command RU or ON, the SY command will assume the log device as the default LU.

The SY command is a privileged command and subject to privileged command syntax rules.

```
+--------+
| LEVEL  |
|  10    |
+--------+
```

TE (Send Message to System Console)

Allows user to send a message to the system console.

```
+------------------------------------------------------------------+
|                                                                  |
|  TELL,message                                                    |
|    --                                                            |
|------------------------------------------------------------------|
|                                                                  |
|  message     Message to be sent to system console; must conform  |
|              to parameter syntax rules for privileged commands.  |
|                                                                  |
+------------------------------------------------------------------+
```

EXAMPLE:

```
   :TE,MESSAGE TO CONSOLE <-- command typed in at user's terminal.

    TE,MESSAGE TO CONSOLE <--- message displayed on system console
                               CRT.
```

COMMENTS:

The message may consist of any printable upper-case ASCII characters. It is limited by the length and number restrictions placed on any FMGR command parameters (refer to the Parameter Syntax Rules section). Note that any commas divide the message into separate parameters. A one-parameter message could be as long as 60 characters.

```
                                                  +-------+
                                                  | LEVEL |
                                                  |   1   |
                                                  +-------+
```

TR (Transfer Control)

Allows transfer of control to a file or LU, optionally passing values to the globals 1G through 9G.

```
+--------------------------------------------------------------+
|                                                              |
|              --                      --                      |
|             | ,namr                   |                      |
| TRANSFER   | ,-integer [,parameters] |                      |
|   ------   |                         |                      |
|              --                      --                      |
|                                                              |
|--------------------------------------------------------------|
|                                                              |
| namr         Identifies  file  or LU to which TR transfers.  If |
|              omitted,  TR  returns  control  to  the  namr of a |
|              previous  transfer.   Up  to  10  transfers can be |
|              nested, and a stack of return pointers is saved.  |
|                                                              |
| -integer     Indicates  a transfer back the specified number of |
|              files in the nested stack.   The transfer stack is |
|              flushed  if the integer  exceeds the current level |
|              of nesting.                                     |
|                                                              |
| parameters   The values to be set into the  globals  1G through |
|              9G.  The position  determines  to which global the |
|              value is passed; omitted globals are unchanged.  |
|                                                              |
|                            **NOTE**                          |
|                                                              |
|                A colon (:), a comma (,), or a space ( ) may  |
|                replace "TR," as the transfer command code.   |
|                                                              |
+--------------------------------------------------------------+
```

EXAMPLES:

1. Set up procedure file RP to restore a main program and two segments or three main programs or two main programs and a segment:

```
            :RP,1G
    file    :RP,2G
    "RP"    :RP,3G
            ::
```

   Transfer to RP to restore program MAIN and two segments SEG1 and SEG2:

```
    ::RP,MAIN,SEG1,SEG2
```

2. Assume three procedure files, FILEA, FILEB, and FILEC stacked so that FILEA transfers to FILEB to FILEC:

```
    :TR,FILEA
    :

   (FILEA)    :TR,FILEB
   (FILEB)    :TR,FILEC
   (FILEC)    :TR,-3     <---- returns to command after original
                               transfer.
```

COMMENTS:

Whenever a TR command is executed, the "namr" associated with TR is saved in a stack that can contain up to ten file names or LU numbers of these ten. The first file name or LU stacked is the input device; the last entry is reserved in case an error necessitates a transfer to the log device. This allows the system to transfer back when TR or "TR, -integer" is specified. If the "namr" identifies a disc file, the current record number is also saved. The transfer returns to the point immediately following the specified TR command.

When transferring back to a control file on disc, it is possible to backspace and re-execute one or more commands within that file. To accomplish this, specify a negative integer (-integer) as the security code of a null namr. For example:

```
        :TR,:-2
    or,
        :TR::-2
    or,
        :::-2
```

will result in a backspace of two commands while transferring back to the previous control file.

NOTE

> For non-disc devices on which a backspace is legal, e.g., LU 5 in spooled jobs, the backspace record function of the CN command can be used (:CN,5,BR).

If an error requiring operator intervention occurs and the severity code is less than 4, the system transfers control to the log device. The TR command (TR or :) is used, in this situation, o return control to the procedure file or input device where the error occurred.

When parameters are specified, the parameter values are passed to the global parameters 1G through 9G. Position determines which globals receive values. For example:

```
 :TR,PROG2A, , , ,ABC,3270, ,XYZ <---sets 4G to "ABC", 5G to 3270,
            ^ ^ ^          ^          7G to "XYZ".
            | | |          |
            | | |          |
            | | |          +--6G
            | | +--3G
            | +--2G
            +--1G
```

```
+-------+
| LEVEL |
|  60   |
+-------+
```

VL (Assign System Scratch and VMA Backing Store File LU)

Allows the user to define the disc LU where scratch and virtual memory backing store files (scratch files for virtual data on disc) are to reside.

```
+--------------------------------------------------------------+
|                                                              |
| VL,cartridge                                                 |
|                                                              |
+--------------------------------------------------------------+
|                                                              |
| Negative LU or  CRN of disc cartridge  where VMA backing store |
| files are to reside.  The cartridge must be a system cartridge |
| other than LUs 2 or 3.   If the command is not entered   or the |
| cartridge   is 0  the  backing  store file  will default to the |
| cartridge in the user's cartridge list.                      |
|                                                              |
+--------------------------------------------------------------+
```

EXAMPLE:

```
 :VL,SP        Set backing store cartridge
 :RU,VMAPR     When a VMA program is run the default backing store
               file will be created on cartridge SP.
```

COMMENTS:

The virtual memory system creates a backing store file (scratch file) for the virtual data. By default the file is created on the first available cartridge in the user's cartridge list; whether private, group, or system.

The VL File Manager command sets the cartridge of the virtual memory backing store file to the specified cartridge where all backing store and system scratch files can reside.

Certain other system utilities (including EDIT) and languages also use the cartridge specified in the VL command for scratch files rather than defaulting to the top of the user's cartridge list. Refer to the RTE-6/VM System Managers Manual for information on utilities that use this cartridge and on the definition of a system scratch cartridge.

```
+--------+
| LEVEL  |
|  10    |
+--------+
```

WH (Run WHZAT Program)

Schedules the WHZAT program to display the current system status. Both program and partition status may be selected.

```
+----------------------------------------------------------------+
|                                                                |
| WH [,lu[,option[,prog]]] <---displays status on specified LU.  |
|    or                                                          |
| WH [,option[,prog]]      <---displays status on input terminal.|
|                                                                |
|----------------------------------------------------------------|
|                                                                |
| lu      The session LU  of the device on which the information |
|         is to be displayed. Defaults to the LU of the terminal |
|         from which command is input.                           |
|                                                                |
| option  Indicates type of information to be displayed. Default |
|         is to display  only those programs associated with the |
|         user's session.                                        |
|                                                                |
|         AL  Status of all scheduled and suspended programs.    |
|                                                                |
|         SM  Same as AL option except that all state-3 programs |
|             not having  "father-son"  relationship will not be |
|             displayed.                                         |
|                                                                |
|         PA  Status of all partitions being used.               |
|                                                                |
|         PL  Status of all programs  (or a specific program) in |
|       or    the system.  This gives a list of all ID segments. |
|         PR                                                     |
|                                                                |
| prog    Program name. Only the status of the specified program |
|         is displayed. This parameter can only be used with the |
|         PL or PR option.                                       |
|                                                                |
+----------------------------------------------------------------+
```

EXAMPLES:

```
:WH <---- Displays status  of scheduled  and suspended programs
          associated with the user's session on input terminal.

:WH,PA <--Displays status, on input terminal, of all partitions
          being used.
```

COMMENTS:

The WHZAT program can also be run outside the session environment:

```
:RU,WHZAT[,lu[,option[,prog]]]
```

For more details, refer to WHZAT description in the RTE-6/VM Utility Programs Reference Manual.

Procedure Files

A procedure file is a set of FMGR commands stored in a disc file. Instead of entering each FMGR command interactively at the terminal, a group of consecutive commands can be saved as a procedure file to which control can be transferred using the transfer command (TR or :). This feature is especially useful when the same set of commands is to be used over and over.

Interactive FMGR commands can be upper or lower case characters, but FMGR commands in a procedure file must be upper case characters only.

A procedure file can be generalized with global parameters (refer to the Global Parameters section). The TR command allows the user to pass values to global parameters 1G through 9G when control is transferred to the procedure file (refer to the TR COMMAND in this chapter).

A set of FMGR commands that manipulate parameters may be useful within a procedure file. These commands are:

Command	Purpose
SE	Assigns values to the global parameters 1G through 9G.
CA	Can assign values to most P and G globals from results of arithmetic or logical calculations, and can null most P and G globals. (Globals -36P through -1P, 1P through 6P, and 1G through 9G.)
IF	Compares two values (usually globals) and skips a specified number of commands depending on the result of the calculation.

Other FMGR commands often used in procedure files include:

Command	Purpose
**	Allows inclusion of comment lines within command entry list to explain the command flow.
DP	Displays parameter values or a message string to the log device.
PA	Suspends execution of the procedure file and transfers control to the log device or some other specified device; also, can optionally send a message to the device where control is being transferred.
SV	Changes severity code; allows inhibiting of echo commands on log device.

Each procedure file is terminated with TR or : in order to return control to the point of origin. Up to ten procedure files can be nested. The return is normally to the command following the TR in the preceding procedure file. If files are nested, the return may skip to any of the preceding files.

Privileged FMGR commands in procedure files, located on disc cartridges LU 2 and LU 3, are not restricted by FMGR command capability level checking. This allows the System Manager to create procedure files on LUs 2 and 3 which can circumvent certain Session Monitor constraints for special applications. A procedure file cannot execute a pack command for the cartridge the procedure file resides on or FMGR will abort.

A typical example might be for a low capability level session user to have access indirectly to a higher capability level command by transferring to a procedure file which executes the command for him. Only the System Manager or a non-session user can create a procedure file on LU 2 or LU 3.

General Example Using Procedure File with Globals

This example illustrates all the commands discussed with procedure files. A procedure file MOVE is used to copy a number of files from one FMP cartridge to another without copying all the files. Four message files are used to request input specifying the cartridge to be copied, the cartridge to which files are copied, a security code for the copied files, and the names of the files themselves. If desired, the copied file can be given a new name.

When you transfer to MOVE with the TR command, you should enter the session LU of the terminal at which you want to enter input as the first parameter to be passed to MOVE. If omitted, a message is displayed at the log device and you must re-enter the command with the input parameter.

The messages asking for input are displayed at the terminal you specify. The procedure then pauses so that you can return the information as parameters in the : version of the TR command.

The message files are:

File #MESS1

```
  PRECEDE ANSWERS TO THE FOLLOWING QUESTIONS WITH A COLON AND A
  COMMA:
  :,ANSWER1,ANSWER2,...
```

File #MESS2

```
  ENTER THE CRN OF THE CARTRIDGE WITH THE FILES TO BE MOVED:
```

File #MESS3

```
  ENTER DESTINATION CARTRIDGE'S CRN, FILE'S NEW SC:
```

File #MESS4

NOW ENTER SOURCE AND DESTINATION FILE NAMES. IF NEW FILE IS TO HAVE THE SAME NAME AS OLD, ONLY ONE NAME IS REQUIRED. END LIST WITH /E.

The file MOVE:

```
:SV,1
:CA,4,0G <-------------------------------- save terminal LU in 4G.
:IF,,36P,EQ,1,2 <------- if not specified, display error message.
:DP,YOUR LU WAS NOT GIVEN
:: <-------------------------------------------------- and return.
:DU,#MESS1,4G <------------------- send first message to terminal.
:DP
:DU,#MESS2,4G
:PA,4G <-- at pause, operator enters TR followed by information.
:CA,6,1G <------ CR of cartridge containing files is saved in 6G.
:DU,#MESS3,4G
:PA,4G <------------ enter cartridge and security code at pause.
:CA,8,2G <------------------------------------ security (SC) in 8G.
:CA,7,1G <------------------------------- and cartridge (CR) in 7G.
:DU,#MESS4,4G
:DP <---------------------- note use of DP to generate blank line.
:SE,0,0,0 <-------- clear 1G, 2G, and 3G prior to entry of files.
:PA,4G,FILES: <--------------------------------------- enter files.
:IF,1G,EQ,/E,5
:IF,1G,EQ,,,3
:IF,,32P,GT,1 <--------------------------- test for new file name.
:CA,2,1G
:ST,1G::6G,2G:8G:7G::,1 <--------------------------- transfer file.
:IF,A,EQ,A,,8 <----------------------------- return for next file.
:SV,0
::
```

To use this procedure:

```
  :TR,MOVE
  :SV,1
    YOUR LU WAS NOT GIVEN
  :TR,MOVE,1


  PRECEDE ANSWERS TO THE FOLLOWING QUESTIONS WITH A COLON AND A COMMA:

      :,ANSWER1,ANSWER2,...


  ENTER THE CRN OF THE CARTRIDGE WITH THE FILES TO BE MOVED:

  :PA,4G
  ::,2


  ENTER DESTINATION CARTRIDGE'S CRN, FILE'S NEW SC:

  :PA,4G
  ::,13,JG


  NOW ENTER SOURCE AND DESTINATION FILE NAMES. IF NEW FILE IS TO HAVE
  THE SAME NAME AS OLD, ONLY ONE NAME IS REQUIRED.
  END LIST WITH /E.


  :PA,4G,FILES:
  ::,OLDFIL,NEWFIL <---enter first file to be moved and rename it.
  :PA,4G,FILES:
  ::,FILEA <-----------enter next file and use same name for copy.
  :PA,4G,FILES:
  ::,/E <-------------------------------------------no more files.
  ::
  :
```

Command Stacking

For FMGR commands which are input from a terminal, several typing aids are provided by the FMGR program. All FMGR commands entered from a terminal, whether they are correct and executed, or are incorrect and produce an error, are placed in a command stack resident within the FMGR program. These commands may be displayed, modified and executed later. The modify commands are a subset of the edit commands provided by the EDIT program. The following stack manipulation commands are provided:

:Ln "n" is the number of lines to list (default is to list the entire command stack).

:P Display or edit the pending line in the command stack. Edit options are CNTL/R, CNTL/I, CNTL/S, CNTL/T and CNTL/C. Refer to the Edit/1000 User's Guide for a description of these options. Also see the example below

:n Position pending line to the "n"th line in the command stack.

:^n or Rn Position "n" lines preceding pending line.

:/n Position "n" lines past pending line.

:-n Delete "n" lines from command stack from the pending line.

Once the pending line has been displayed, it may be executed by typing a carriage return. On 264x terminals the cursor keys can be used to position the cursor to the desired entry in the command stack. This command can be executed by hitting the enter key.

Whenever a command is executed it is placed at the bottom of the stack. An aging feature eliminates duplicate entries from the command stack. A command is placed on the stack, all other entries are checked for duplication and deleted if found.

To be assured of a command stack position, use "TR" for the transfer command. When a colon (:) or comma (,) is used, the command is not placed in the command stack.

Implied RUN: All input from a terminal that is not a legal FMGR command or a command stack command will have a :RU, placed in front of the input string and will be interpreted as a run command. This feature allows programs and FMGR commands to appear to be more homogeneous. It also reduces the amount of typing which is required.

When using the Implied RUN feature, caution should be taken in naming user programs. A user program should not be named the same as a valid alternate form of a FMGR command if the implied RUN command is to be used. Some FMGR commands allow extra characters (beyond 2) to be an alternate form of the command. For example, the ST command can also be specified as STORE. If a user program is named STO, STOR, or STORE and is executed with the implied RUN, the FMGR ST command will be executed instead of the user program. These programs will run with the RU command. The implied RUN feature cannot be used in a transfer file or a FMGR-056 error will occur.

With the implied RUN command installed, the pending line edit feature is slightly limited. As noted below, the first character of the pending line may not be edited since FMGR will recognize the input as an implied RUN command.

EXAMPLE	COMMENTS
:L RU,FTN7X,&TEST1,-,-	This is the listing of the command stack.
:PRU,FTN7X,&TEST2,-,-	User is attempting to edit the pending line.
FMGR 067	FMGR issues an error, recognizing "PRU" as an implicit RUN command of the program PRU.
:P/////////////2	One correct method is to replace the first character with "/".

Chapter 4
System and Break-Mode Commands

Introduction

System and break-mode commands allow the user to interactively communicate with the RTE-6/VM Operating System to obtain system status, modify system parameters and control the scheduling of programs. Functions performed by these commands include:

* Schedule and terminate a program.

* Suspend and restart a program.

* Schedule a program to execute at a specific time.

* Change the priority and size of a program.

* Examine the status of any partition, program or I/O device.

* Alter the system I/O structure.

* Examine and alter the time-out parameter of an I/O device.

* Declare I/O device up or down.

* Initialize the real-time clock and print the time.

* Set device buffering limits.

* Release disc tracks assigned to a dormant program.

The commands and their functions are summarized alphabetically in Table 4-1; complete descriptions are given later in this chapter.

Chapter 2 of this manual introduced the user to the operational differences between system and break-mode commands. In this chapter, the Processing Of System And Break-mode Commands section describes the fundamental differences of how the system processes the two modes.

Table 4-1. System and Break-Mode Command Summary

COMMAND	CAPABILITY LEVEL	DESCRIPTION
AB	--	Abort current executing batch job.
AG	60	Modify the partition priority.
AS	50	Assign program to partition.
BL	10	Examine current buffer limits.
BL	60	Change current buffer limits.
BR	10	Set break flag in program's ID segment (program within session).
BR	60	Set break flag in program's ID segment (any program in system).
DN	60	Down an I/O controller.
EN	--	Enable system console to be session terminal.
EQ	10	EQT description and status.
EQ	60	Change automatic buffering designation for an EQT.
FL	10	Flush terminal buffer.
GO	30	Reschedule operator suspended program (program within session).
GO	60	Reschedule operator suspended program (any program in system).
HE	1	Call HELP program.
IT	50	Put program in time list.
LU	60	Print EQT assignment for LU.
LU	60	Reassign EQT assignment for LU.
OF	30	Terminate a program (program within session).
OF	60	Terminate a program (any program in system).
ON	50	Schedule a program.
OP	--	Allow system level operator command.
PR	50	Change program priority.
QU	10	Examine current system timeslice parameters.
QU	60	Modify system timeslice quantum and/or fence.
RS	10	Abort and reschedule a session's copy of FMGR.
RT	30	Release disc tracks assigned to a dormant program.
RU	30	Run a program.

Table 4-1. System and Break-Mode Command Summary (continued)

COMMAND	CAPABILITY LEVEL	DESCRIPTION
SL	10	Display session LU information.
SS	30	Suspend a program (program within session).
SS	60	Suspend a program (any program in system).
ST	10	Display program or partition status.
SZ	30	Print current program size information.
SZ	30	Change program size requirement.
TE	10	Send message to system console.
TI	10	Print time.
TM	60	Set real time clock.
TO	10	Print time-out value for EQT.
TO	60	Set time-out value for EQT.
UL	60	Release shareable EMA partition.
UP	10	Up an I/O controller.
UR	50	Release a reserved partition.
VS	30	Print current virtual memory size.
VS	30	Change virtual memory size.
WH	10	Call the program WHZAT.
WS	30	Print the current working set size.
WS	30	Change the working set size.

Program States

Within the RTE-6/VM operating environment, programs can exist in 7 states. The state of a program can be changed by the user (RTE-6/VM System command), by another program (EXEC call), or by the system to reflect an environmental condition, i.e., program requested memory or disc space that was not available, requested I/O on a down device, etc. At any given time, the state of a program indicates its relationship to the RTE-6/VM operating environment.

State 0 - DORMANT. The program is not scheduled to execute. A program can be in this state if it was never scheduled or was placed in this state by an operator command or by an EXEC call. The program is known to system and can easily be scheduled; no file or directory search time is required to find the program.

State 1 - SCHEDULED. The program is scheduled to execute; it has been placed in the scheduled list. The program that is currently executing is generally at the head of the scheduled list and is therefore in State 1.

State 2 - I/O SUSPENDED. The program has requested I/O service and the system is currently performing the I/O operation. This condition occurs with any input operation, or with an output operation to an unbuffered device. If the class I/O technique discussed in the RTE-6/VM Programmer's Reference Manual is used, or if the output device is buffered, the program is allowed to continue executing while the I/O operation is being performed.

State 3 - GENERAL WAIT. The program has requested system services or resources that are temporarily unavailable. For example, a program would be placed in this state if it tried to talk directly to an I/O device but another program had already locked the device for its own use. Alternately, if one program wants to schedule a second program and waits until the second program terminates before resuming execution (schedule with wait), the first program would be put into state 3 for the duration of the second program's execution and then automatically be rescheduled upon termination.

State 4 - MEMORY SUSPENDED. The program has requested an operation that requires the use of System Available Memory (SAM) and an insufficient amount of SAM is available. This is a temporary state and when enough SAM becomes available, the program will by placed back into the scheduled list.

State 5 - DISC SUSPENDED. The program has requested disc space that is unavailable. This is a temporary condition and when disc space becomes available the program will be placed back into the scheduled list.

State 6 - OPERATOR SUSPENDED. The program is suspended by an operator command or by an EXEC call from within the program. Another operator command or EXEC call is necessary to remove the program from this state.

Program State Lists

RTE uses linked lists for keeping track of programs within a given state list. Using this technique, RTE moves programs from state to state by linking and unlinking the program's ID segment from the appropriate lists.

Each state list originates with a list header pointing to the ID segment of the first program in the list. The first word of each program's ID segment is then used as a linkage word pointing to the ID segment of the next program in the list. The order of the programs within the list is on a priority basis with the highest priority program first.

The list headers for program states 1, 3, 4, 5 and 6 are located within the system base page (refer to the Memory Management section of the RTE-6/VM Programmer's Reference Manual for a description of the system base page). The list header for program state 2 (I/O Suspend) is the first word of the Equipment Table (EQT) entry for the device to which I/O is being directed or requested. With RTE-6/VM there can be up to 255 of these I/O lists. Note that state 0 (dormant) has no list associated with it.

Figure 4-1 illustrates examples for a scheduled and an operator suspended state list. Both lists originate with the list header containing the address of the first program's linkage word. Since there is only one program in the operator suspended list, the program's linkage word contains a zero to indicate that it is the last program in the list. In the scheduled list, each program's linkage word contains the address of the next program's linkage word except the last which contains a zero.

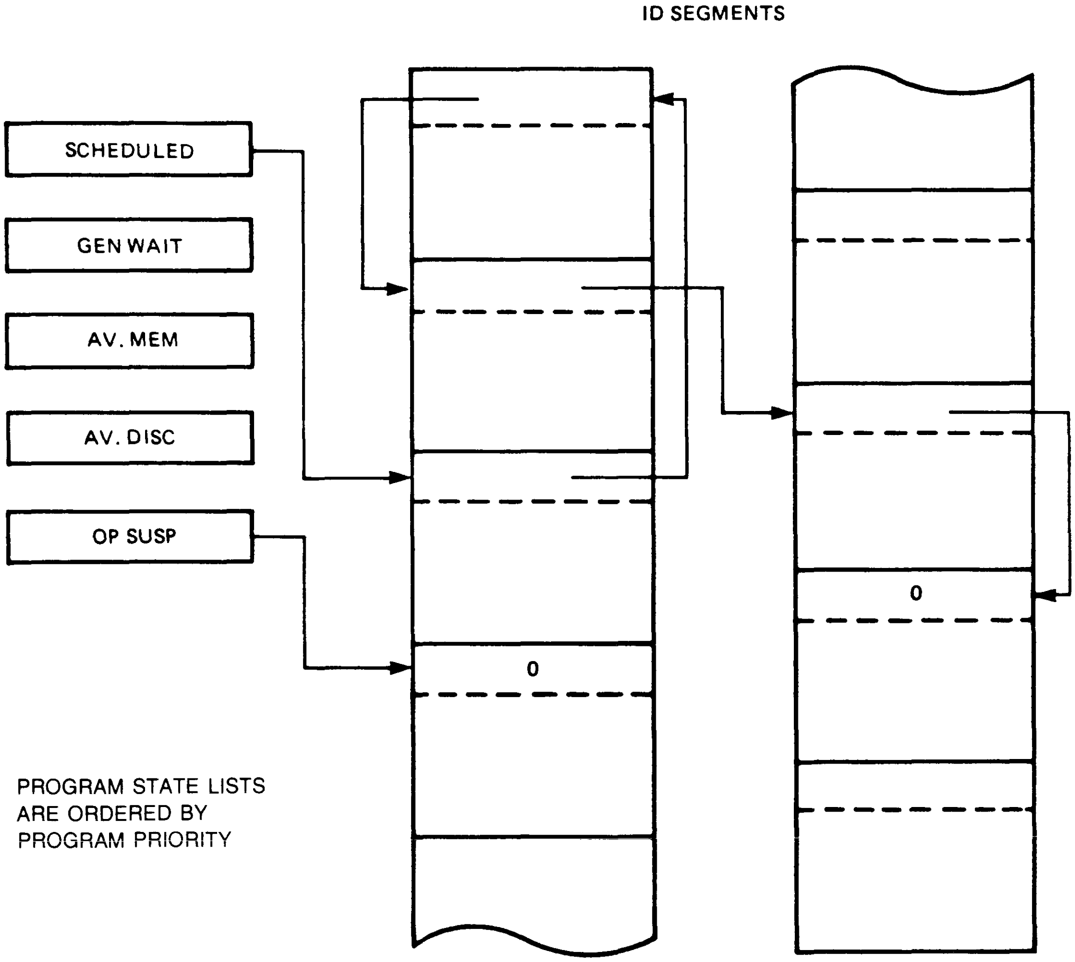


Figure 4-1. Program State List Examples

Buffered and Unbuffered Input/Output

The four ways in which I/O operations may be processed in RTE are:

1. Unbuffered I/O
2. Re-entrant I/O
3. Automatic Output Buffering
4. Class I/O

Unbuffered and re-entrant I/O require that the I/O process be completed before the program can continue executing (i.e., the program will be I/O suspended). For disc-resident programs, the main difference between the two I/O processes is that re-entrant I/O allows program swapping and unbuffered I/O does not. Terminal input requests should always be done using re-entrant I/O to allow program swapability while the input is being entered at the terminal.

Both unbuffered and re-entrant I/O link the program ID segment to the Equipment Table (EQT) entry of the device to which I/O is directed or requested. However, with re-entrant I/O, a temporary data block (TDB) is created in System Available Memory (SAM) to act as a buffer to the device. Since I/O is read from or to the buffer instead of the program, the program can then be swapped out if a higher priority program needs its partition. The TDB is linked to the program and EQT through the program's ID segment.

Automatic output buffering and class I/O allow programs to continue executing while I/O is being processed. Both processes create buffers in SAM to which data is temporarily stored. The SAM buffer is then linked to the device's EQT.

Figure 4-2 illustrates the basic difference between buffered and unbuffered output to a device (for an input operation, the arrows would point in the opposite direction). Figure 4-3 illustrates how I/O requests are linked to the EQT for buffered and unbuffered I/O.

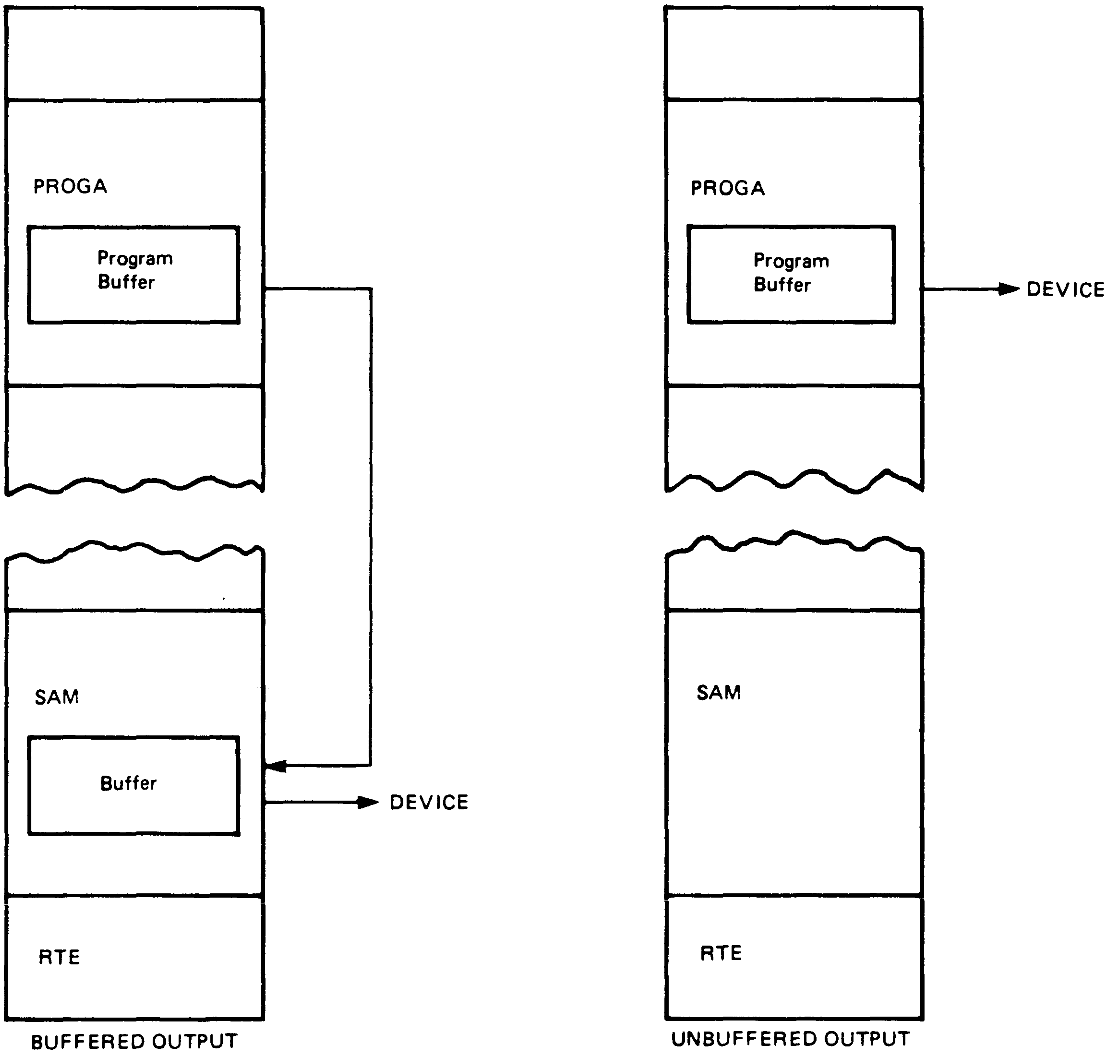


Figure 4-2. Buffered and Unbuffered Output to a Device

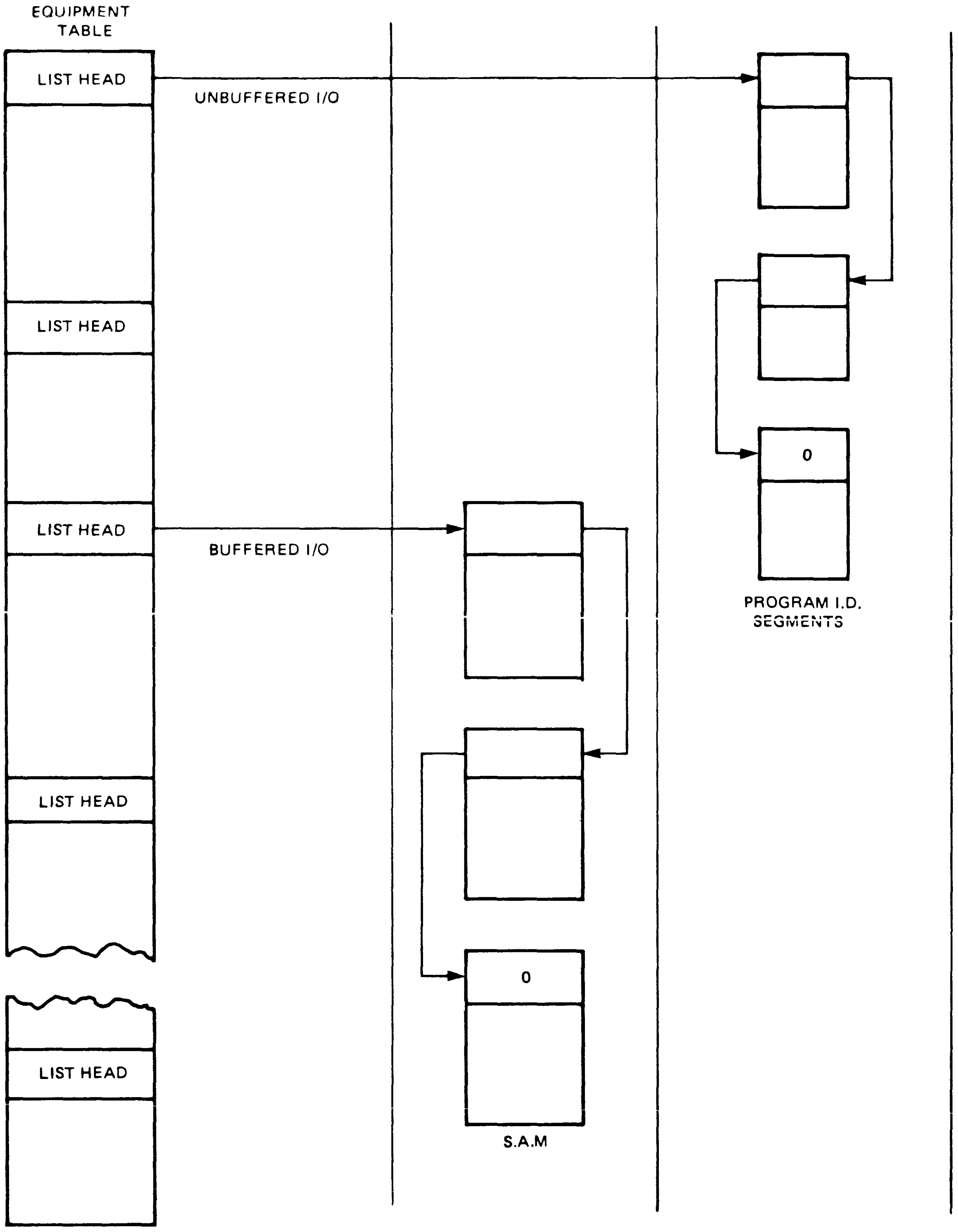


Figure 4-3. Buffered and Unbuffered I/O Linkage

A device can be designated as using automatic output buffering or being unbuffered by setting the appropriate bit in its associated Equipment Table (EQT) entry. This bit can be set at either system generation (refer to the RTE-6/VM On-Line Generator Manual) or interactively through the break-mode EQ command described later in this section.

A read or write request (EXEC 1 or 2) contains a bit which, when set, forces the request to be unbuffered. Refer to the RTE-6/VM Programmer's Reference Manual for further details.

There are two situations where an EQT can have both program ID segments and SAM buffers directly linked to it:

1. The EQT is unbuffered and Class I/O requests are linked to it.

2. The EQT uses automatic output buffering and unbuffered and/or re-entrant I/O read requests are linked to it.

Figure 4-4 illustrates an example where both SAM buffers and program ID segments are linked to the same EQT.

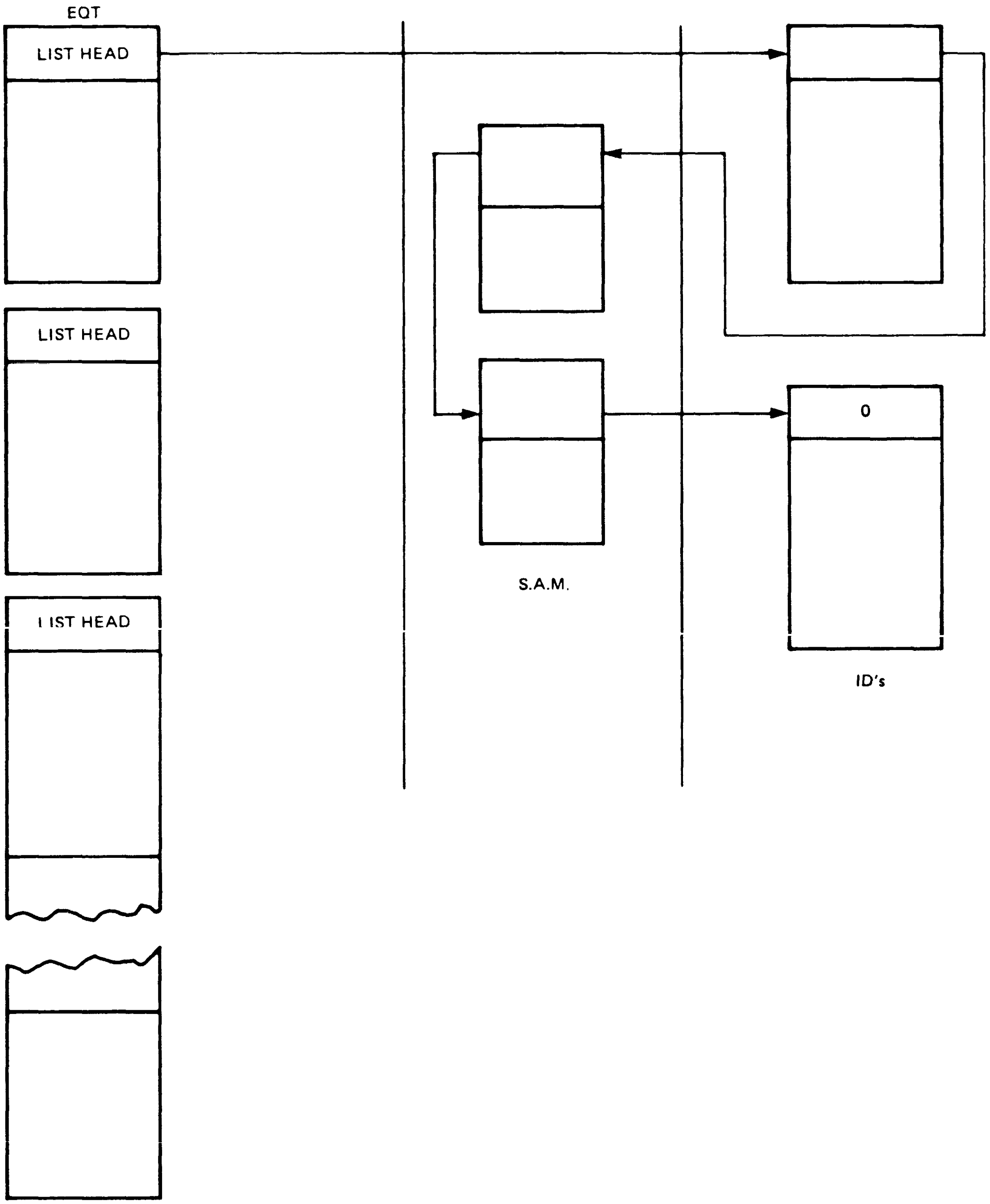


Figure 4-4. I/O Request Linkage

Time List

RTE allows users to schedule programs to execute at specific times by placing them in the time list. When the specified time for the program to execute arrives, it is placed in the scheduled list by RTE.

Programs in the time list are linked together through word 17 of their respective ID segments. The ordering of the programs is based upon a first-in, first-out (FIFO) basis. Also stored within the ID segment is the next execution time and execution time interval for the program.

When the next execution time for the program arrives, the program is placed in the scheduled list by RTE (refer to Program State Lists in this chapter). Figure 4-5 illustrates an example of a time list and a scheduled list. Note that programs PROGA and PROGC are in both the time list and scheduled list.

If a session program is in the time list, the following access restrictions are enforced:

1. EXEC schedule or program time value requests referencing a program in the time list may only be issued by another program of the same session. If an attempt is made to time schedule a program that is currently scheduled (or in the time list) for another session, the calling program will receive an SC11 error. If an attempt is made to schedule a program that is currently scheduled for your session, the system will ignore the attempt.

2. The session operator commands IT, RU and ON have the same access restrictions as described above. Attempts to reference a time scheduled program belonging to another session will result in an "ILLEGAL STATUS" error.

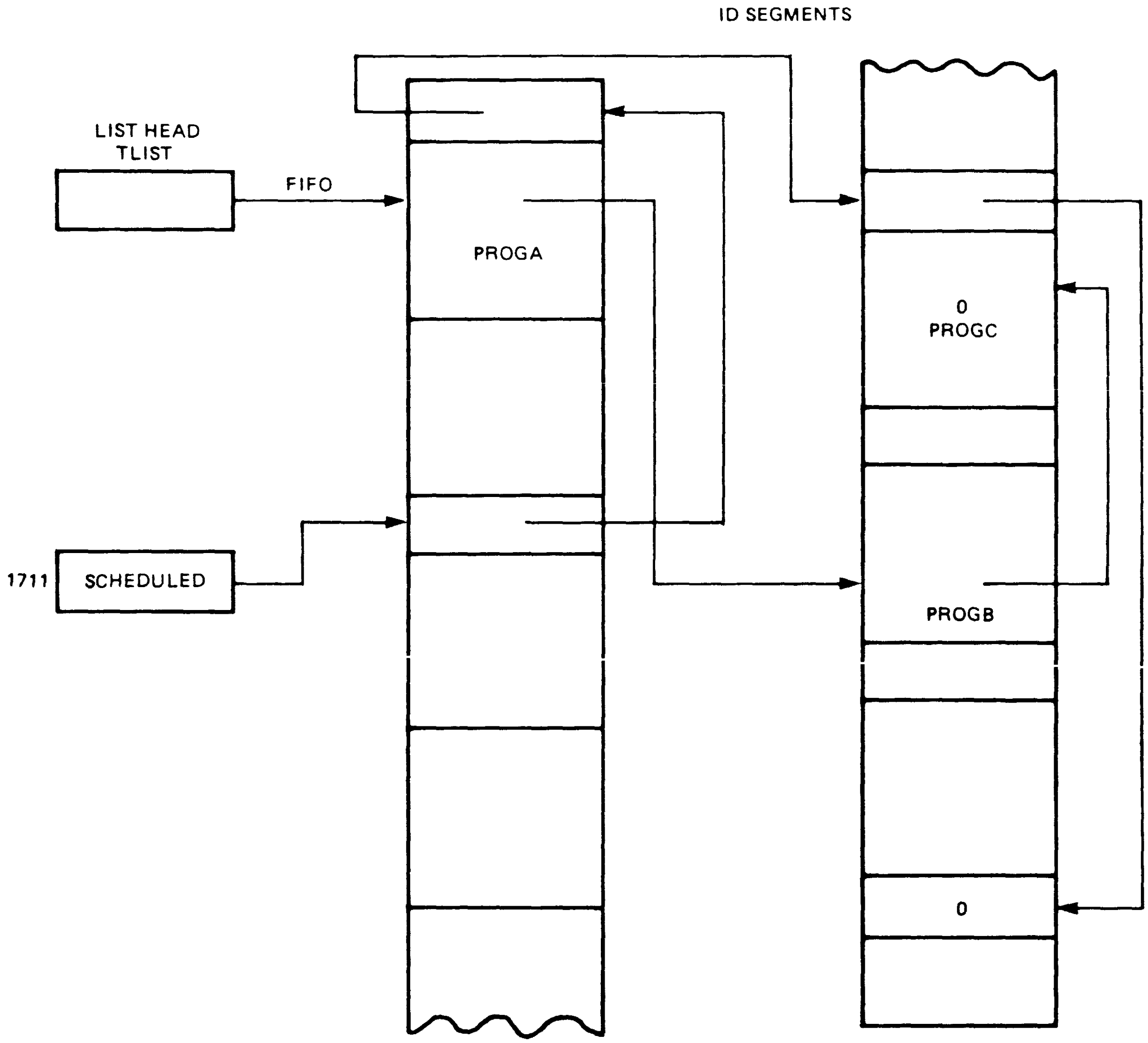


Figure 4-5. Time List Example

Processing of System and Break-Mode Commands

Processing of break-mode commands is done through the PRMPT and RPN software modules. Both the Session Monitor and the Multi-Terminal Monitor (MTM) have their own versions of these two modules. System commands are processed by the operating system.

Break-mode can only be entered from terminals which schedule the prompt processor (PRMPT) when the user makes an unsolicited interrupt at the terminal. When operating under Session Monitor, break-mode can be entered from any session terminal, including the system console if it has been enabled to operate in session. When operating under MTM, break-mode can be entered from any auxiliary terminal (i.e., any terminal other than the system console).

An unsolicited interrupt occurs when the user strikes a key while the system does not have a read request pending from the terminal (i.e., no prompt is issued to the CRT). After the interrupt is recognized by the system, the break-mode prompt:

```
     S=xx COMMAND?          (for a session terminal)

          or

     lu>                    (for an MTM terminal)
```

When xx is the session identification number or LU is the logical unit number of the MTM terminal, the above prompts are issued to the terminal CRT.

The response processor (RPN) is scheduled once by PRMPT and then suspends itself on a class "GET" request (refer to the RTE-6/VM Programmer's Reference Manual for a description of Class I/O). All operator requests input in break-mode are thus passed to the RPN module through the class "GET" request.

The RPN module passes most of the commands entered through break-mode directly to the operating system for processing, but some commands require special handling which the RPN module processes. The following commands are session-related and receive special handling by RPN:

FL - make a control request to flush all requests to this terminal's driver until all buffered requests have been cleared or a read request is found.

WH - schedule the WHZAT program for this session.

HE - schedule the HELP program for this session.

SL - examine contents of Session Switch Table (SST).

TE - send a message to the system console.

RS - abort and then restart this session's copy of FMGR.

OF - if no parameters, abort the current program running under the session's copy of FMGR (FMGxx). If FMGxx does not have a son running, the break flag is set in FMGxx's ID segment.

BR - if no parameters, set the break flag of the current session program.

If break-mode is entered while operating under MTM, the following commands receive special handling by RPN (the MTM Version):

FL - make a control request to flush all requests to this terminal's driver until all buffered requests have been cleared or a read request is found.

BR - if no parameters, set the break flag of the last son of FMGxx. If FMGxx does not have a son running, the break flag is set in FMGxx's ID segment.

AB - if no parameters, abort the last son of FMGxx (same as OF,program,1). If FMGxx does not have a son running, the break flag is set in FMGxx's ID segment.

Note that several commands available under Session Monitor are not available under MTM (i.e., WH, HE, SL, TE, and RS).

All other commands entered during break-mode are passed to the system for processing. See Table 4-1 for a complete list of the break-mode commands.

Three commands which are only valid when entered from the system console are:

AB - abort the currently executing batch job.

EN - enable system console to be session terminal.

OP - allow system level operator command.

For more information on these commands, refer to the System Console description in Chapter 2.

Commands which the operating system processes can also be entered from File Manager (FMGR) via the SY command (refer to chapter 3 on FMGR commands for more details). File Manager strips off the SY prefix and passes the remaining command string directly to the operating system.

Figures 4-6 and 4-7 summarize System and Break-mode command processing under Session Monitor and MTM, respectively. The prompts shown are:

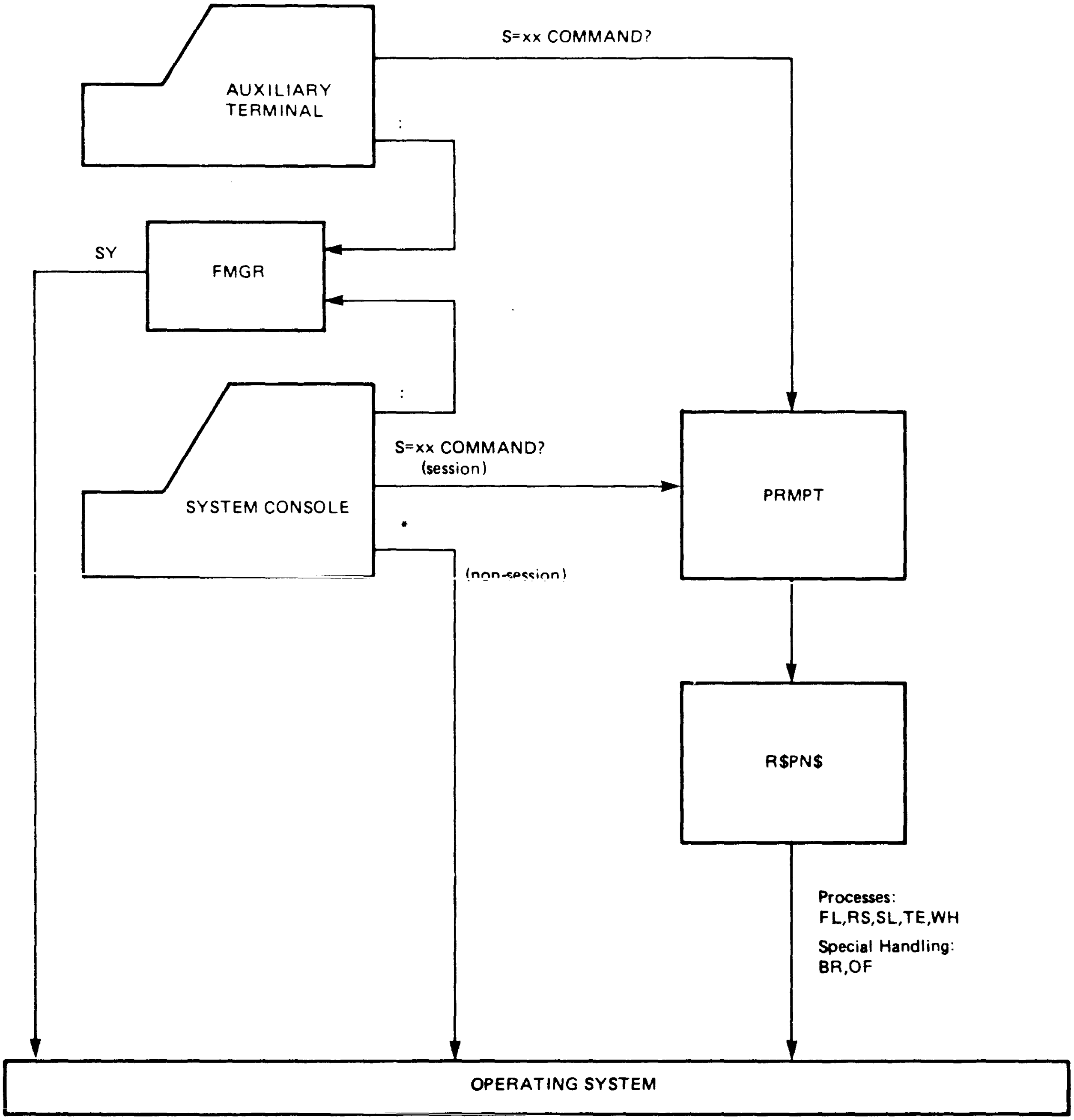
```
1)  *                 <--- System Prompt (System Console)

2)  :                 <--- FMGR Prompt

3)  S=xx COMMAND?     <--- Session Break-mode Prompt

4)  lu>               <--- MTM Break-mode Prompt
```

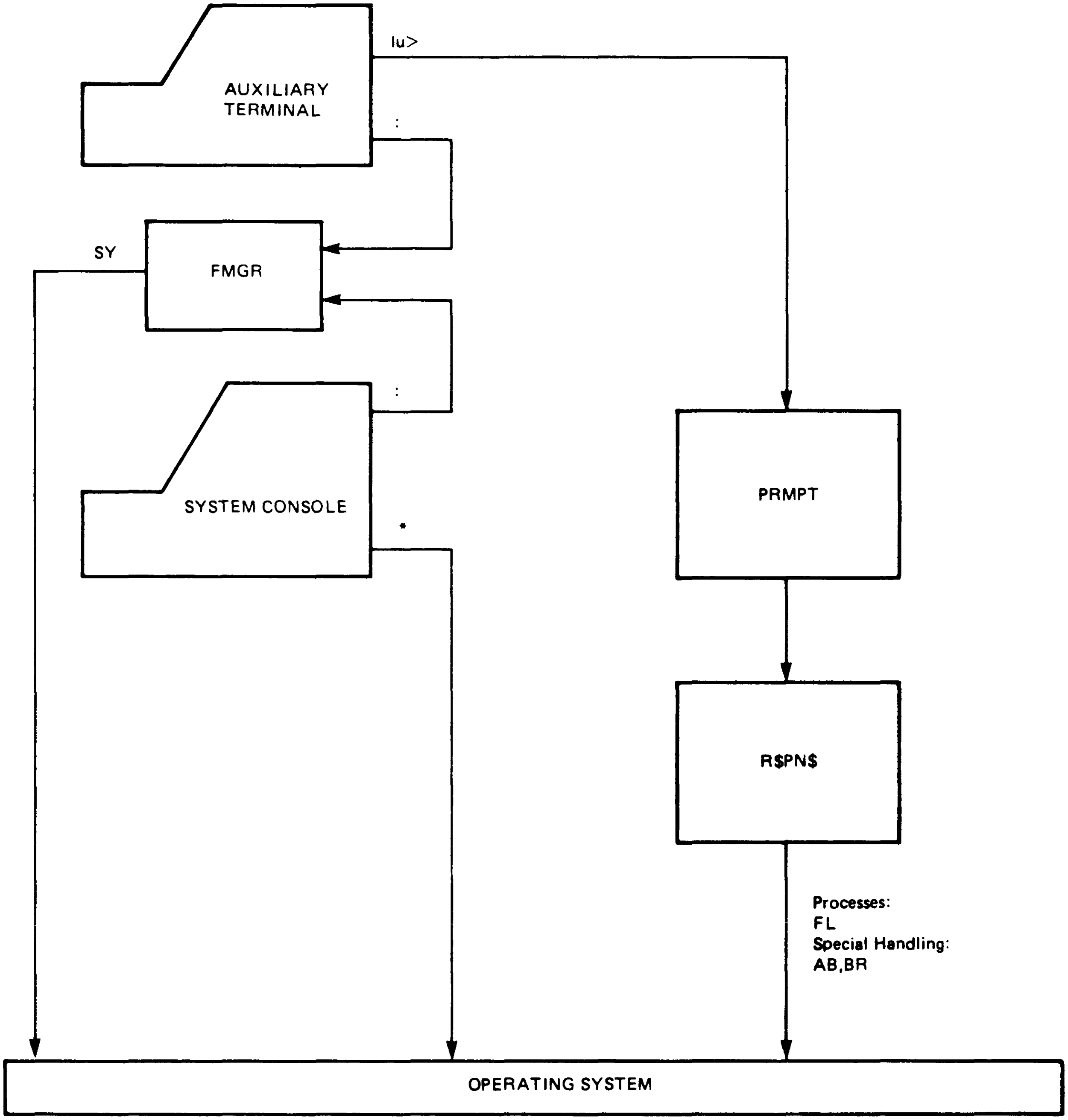
By following the arrowed paths starting with the appropriate prompt, the user can quickly see which commands are valid for each prompt.



Processes: AG,AS,BL,BR,DN,EQ,GO,IT,LU,OF,ON,PR,QU,RT,RU,
SS,ST,SZ,TI,TM,TO,UL,UP,UR,VS,WS

Special Handling: AB,EN,OP

Figure 4-6. Break-Mode Command Processing Under Session Monitor



Processes: AG,AS,BL,BR,DN,EQ,GO,IT,LU,OF,ON,PR,QU,RT,RU,
SS,ST,SZ,TI,TM,TO,UL,UP,UR,VS,WS

Special Handling: AB

Figure 4-7. Break-Mode Command Processing Under MTM

System and Break-Mode Commands

The following are detailed explanations of the interactive system and break-mode commands available with RTE-6/VM. System Commands are handled by the operating system; break-mode commands are handled by the program RPN (see Figures 4-6 and 4-7). The capability level required for each command is shown in a box in the upper right hand corner of the page.

Command Structure

Each command is specified by a mnemonic code of two letters to indicate the operation to be performed. Depending on the command, parameters may be entered to further specify the command operation.

Command Conventions

The conventions used in describing the following break-mode commands are summarized in Table 4-2. The following comments are worthwhile to note:

* The items outside the square brackets are required symbols. Items inside the brackets are optional.

* Two commas in sequence sets a parameter to its default value.

* Each command entered must be completed with an end-of-record mark (RETURN or ENTER key on a CRT or TTY system input device).

* An error made while entering a command parameter can be corrected by using the BACK SPACE key on a CRT system input device (the CONTROL and A keys struck simulataneously will delete the last character entered on TTY input devices). To delete an entire line, use the DEL key (RUBOUT key on TTY devices). Corrections to a command must be made before the RETURN key is pressed or the system will issue an error. Note that line feed is supplied by the system.

* Whenever the operating system is rebooted, parameters changed by user command will be restored to their original values established during system generation.

* Any system, break-mode, or FMGR command can be in lower case or upper case when entered from an interactive mode.

The required command capability levels for each of the break-mode commands are shown in the upper right hand corner of the command descriptions. Note that several commands have two levels associated with them, the higher level allowing additional capability.

Table 4-2. Command Syntax Conventions

Item	Meaning
UPPER CASE LETTERS	These words are literals and must be specified as shown.
lower case letters	Symbolic representations indicating what type of information is to be supplied.
[,item]	Items with brackets are optional. However, if item is not supplied, its position must be accounted for with a comma. This causes items to automatically default.
-- -- | ,item 1 | | ,item 2 | | ,item 3 | -- --	Indicates that exactly one item may be specified.
,item 1 ,item 2 ,item 3	Indicates that there is a choice of entries for the parameter, but one parameter must be specified.

```
+-------+
| LEVEL |
|  60   |
+-------+
```

AG (Modify Partition Priority Aging)

Modifies the rate a partition priority is increased and turns on or off partition priority aging.

```
+------------------------------------------------------------------+
|                                                                  |
| AG,numb           <--------- Turns on  partition priority aging |
|                              modifying the aging rate.           |
|   or                                                             |
|                                                                  |
| AG,OF             <--------- Turns off partition priority aging.|
|                                                                  |
|------------------------------------------------------------------|
|                                                                  |
| numb    Number of 10 millisecond  intervals  to be used as the  |
|         rate a partition priority is aged  (aging rate).  This  |
|         value must be in the range 10 <= numb <= 32767.          |
|                                                                  |
| OF      Turns off partition priority aging.                      |
|                                                                  |
+------------------------------------------------------------------+
```

EXAMPLES:

```
1.  S=07 COMMAND? AG,100 <-------Age the partition priority once a
                                 second.

2.  S=07 COMMAND? AG,OF <---Turn off all partition priority aging.
```

COMMENTS:

The partition priority aging feature allows high-priority suspended (state 3) programs to be swapped out in favor of lower priority programs that are scheduled but waiting for a partition. The AG command assigns partitions a priority equal to the occupying program. The partition priority is then increased at a rate specified in the AG command.

In RTE-6/VM, partition resources are allocated on a priority basis. This resource is allocated by swapping a lower priority program out of memory, making that partition available for a higher priority program. This assures that the higher priority program will have a partition to execute in when scheduled. In many applications, there are programs that spend considerable time suspended in state 3 (i.e., waiting for a class read to be satisfied, etc.). In some cases the disc overhead for swapping is too large to allow a fast enough response to a program that must execute immediately after leaving state 3. This program must always reside in a partition, enabling it to run when needed, without the overhead caused by swapping it to a partition. For other programs the delay caused by swapping may be insignificant. For this type of program, it is not necessary that it always reside in a partition, but it must have a high enough priority that it can be swapped into a partition immediately after it leaves state 3. In this situation, the AG command can be used to allow the high-priority suspended (state 3) program to be swapped out on behalf of a lower priority program.

When aging is in effect, for any particular partition the partition priority is initially set to equal the program's priority. When the program enters state 3, the partition priority is increased (aged) by 2 at intervals specified in the AG command. This interval is expressed in units of 10 milliseconds. While the program is suspended the partition priority continues to decrease. When the system allocates partitions to programs, this partition priority is checked, not the program priority. Thus, if a program with priority 30 remains in state 3 long enough (depending on the aging rate specified in the AG command) it could be swapped out in favor of a program with priority 50.

In order for the partition priority to be aged, a program must be currently suspended in state 3 and reside in a partition. A program can be placed in state 3 if no resource is available, a father program is waiting for the son program to finish executing, or if a device is downed during an I/O operation. When the suspended program is rescheduled (moves to state 1), the partition priority is reset to the value of the occupying program priority. Note that the program priority is not altered during this process.

Example 1:

PROGA has priority 30. It spends considerable time in state 3. PROGB has priority 33. This program never enters state 3. Only one partition is available in the system, all others have been reserved at bootup. The AG command can be used as follows:

```
S=07 COMMAND? AG,1000          (age the partition priority every 10
                                seconds.)
```

PROGA is scheduled and begins execution. PROGB is scheduled, but cannot run since PROGA has a higher program priority. (For scheduled programs, state 1, the partition priority is equal to the program priority.)

PROGA issues a class read, for example, and enters state 3; after 10 seconds, its partition priority becomes 32. The operating system checks if any program should be swapped. It checks the partition priority of PROGA. PROGB still cannot run, since its partition priority (33) is lower (greater priority number) than PROGA's partition priority (32).

PROGA remains in state 3, and after 10 more seconds; its partition priority becomes 34. The operating system checks if any programs can be swapped. The system compares PROGA's partition priority (34) to the partition priority of PROGB (33). Since PROGA has a lower partition priority than PROGB it is swapped out. PROGB is swapped in and begins execution.

When PROGA's class read is satisfied, it enters state 1 (scheduled). The operating system checks if any program should be swapped. It checks the partition priority of PROGA, which has been reset to the program priority (32). PROGB is swapped out, PROGA is swapped in and resumes execution.

Example 2:

Another example of where the aging command would be useful is in an RTE-6/VM and DS network environment where the system is being used for program development. In this application, DS has many high priority programs which run occasionally and suspend themselves in state 3 (general wait). If aging is not used, these high priority DS programs tend to stay suspended in memory using up resources. Due to their high priority, they generally will not be swapped by editors or compilers. Thus the programs that are being used (editors and compilers) must fight among themselves for the remaining partitions. By using the AG command, the least used DS programs will be swapped out, and the system will have more partitions available for the currently active programs.

```
+--------+
| LEVEL  |
|  50    |
+--------+
```

AS (Assign Partition)

Assigns a program to always be dispatched to same partition when scheduled for execution.

```
+------------------------------------------------------------------+
|                                                                  |
| AS,program,partition#                                            |
|                                                                  |
|------------------------------------------------------------------|
|                                                                  |
| program      Name  of  program  to  be  assigned  to  specified  |
|              partition.                                          |
|                                                                  |
| partition#   Partition number  which program  is to be assigned |
|              to   (since  a  maximum  of  64  partitions can be |
|              defined at  system generation,  this parameter can |
|              be any integer  from 1 to 64  depending  upon  the |
|              number of memory partitions defined at generation).|
|                                                                  |
+------------------------------------------------------------------+
```

EXAMPLE:

```
 S=07 COMMAND ?AS,PROGA,12
 \-----v------/     ^    ^
   Break-Mode       |    |
     Prompt         |    --partition number
                    |
                    --program name
```

COMMENTS:

Unless a program is assigned to a specific partition, when it is scheduled for execution it is dispatched to any partition of the proper type, large enough to run the program.

The AS command is only valid if the program specified is dormant and is not currently residing in a partition. A program may be dormant and reside in a partition if it ended saving resources, ended serially reusable or was operator suspended; otherwise, the error 'ILLEGAL STATUS' will be returned and the command input ignored by the system.

If the partition specified is not large enough to run the program, an 'ILLEGAL PART'N' error will be returned. Trying to assign a program to an undefined partition will also generate the same error message.

If the program specified cannot be found (i.e., its ID segment cannot be located), a 'NO SUCH PROG' error message will be issued.

If partition# is specified as 0, the program will be unassigned and can be dispatched to any partition of the proper type large enough to run the program.

A program cannot be assigned to a shareable EMA partition; otherwise an "ILLEGAL PART'N" error will be issued. In addition, if a subpartition is a shareable EMA partition, no program can be assigned to the mother partition. If a mother partition is a shareable EMA partition, no program may be assigned to any of its subpartitions.

```
+--------+
| LEVEL  |
| 10/60  |
+--------+
```

BL (Examine or Modify Buffer Limit)

Allows a low capability level user (10) to examine the current buffer limits, and a high capability level user (60) to change the current buffer limits.

```
+------------------------------------------------------------------+
|                                                                  |
| BL [,lower limit [,upper limit ]]                                |
|    \------------v----------------/                               |
|     Requires level 60 to specify                                 |
|                                                                  |
|------------------------------------------------------------------|
|                                                                  |
| lower limit   Lower buffer limit specified in number of words.  |
|               If upper limit is changed and lower limit is not  |
|               specified it defaults to 1.                        |
|                                                                  |
| upper limit   Upper buffer limit specified in number of words.  |
|               If lower limit is changed and upper limit is not  |
|               specified, it remains the same as the existing    |
|               upper buffer limit.                                |
|                                                                  |
+------------------------------------------------------------------+
```

EXAMPLE:

```
  S=07 COMMAND ?BL     <-------------Examine current buffer limits.
    100  400
     ^    ^
     |    |
     |    --current upper limit
     |
     --current lower limit

  S=07 COMMAND ?BL,200,500  <----------------Change buffer limits.

  S=07 COMMAND ?BL     <-------------Examine current buffer limits.
    200  500
     ^    ^
     |    |
     |    --new upper limit
     |
     --new lower limit
```

COMMENTS:

Each time a standard I/O request is made to a device with automatic output buffering enabled, the system adds up all the words in the I/O requests currently queued to the Equipment Table (EQT) entry pointing to the device where I/O is directed. If the sum is less than the upper limit, the new request is added to the queue. If the sum is larger than the upper limit, the requesting program is suspended and placed in the general wait (STATUS=3) list.

When a buffered I/O request completes, the system adds up the remaining words in the I/O requests queued to the EQT entry and compares the number to the lower limit. When the sum is less than the lower limit, any programs suspended for exceeding the buffer limits on the EQT are rescheduled.

Any program with a priority of 1 through 40 will not be suspended for buffer limit, so that alarm messages, etc., are not inhibited.

Setting the upper and lower buffer limits with this command can prevent an inoperative or slow I/O device from monopolizing System Available Memory (SAM).

For instance, by increasing the buffer limit, a program can be allowed to store its output in a buffer and continue executing rather than possibly be I/O suspended because the buffer is already full.

```
+--------+
| LEVEL  |
| 10/60  |
+--------+
```

BR (Set Break Flag)

Sets a break flag in a program's ID segment. With a capability level of 10 or greater, the user can set the break flag in a program currently running under his session. With a level of 60, the user can set the break flag in any program in the system.

```
+-------------------------------------------------------------------+
| BR [,program]                                                     |
|-------------------------------------------------------------------|
| program      Name of program  whose break flag  is  to  be set;   |
|              default is current session program.                  |
+-------------------------------------------------------------------+
```

EXAMPLES:

```
1. S=07 COMMAND ?BR,PROGC <--Set break flag in PROGC's ID segment.

2. S=07 COMMAND ?BR       <--Set  break  flag in  current  session
                             program's ID  segment (the  "current"
                             session program is found by searching
                             for  the  last  son  of  the  session
                             progenitor, FMGxx)

3. S=07 COMMAND ?BR,PROGC <--Session user with  command capability
                             level  of  60  sets   break  flag  in
                             program   currently   running   under
                             another user's session.
```

COMMENTS:

The BR command allows the user to break execution of a program if the program requests this via the IFBRK system subroutine. When BR is executed, a break flag in the named user program's ID segment is set. The user's program can call the HP-supplied subfunction that will test the break flag and then act accordingly.

The calling sequence is: `I=IFBRK(IDUMY)`

Where IDUMY is a dummy parameter to make the call appear as a function (IDUMY need not be supplied in Macro/1000). The returned value will be negative if the break flag is set; the break flag will be cleared by IFBRK.

LEVEL 60

CU (CPU Utilization)

Displays a bar graph in the S-Register representing percentage of CPU utilization.

```
    ON
CU,OFF
```

Turn the CU display on or off.

When the RTE-6/VM system is in the idle loop it approximates the CPU utilization and displays it in the S-Register on the front panel of the CPU when CU is on. CPU utilization is an approximation of the percentage of time spent outside of the system idle loop. CPU utilization is updated and displayed approximately every 160 milliseconds.

For example, if the operating system spends 30 out of 100 milliseconds in the system idle loop the system CPU is spending 70% of its time doing useful work, then the CPU utilization is 70%.

The utilization is displayed as a bar graph starting at bit 15 and going to bit 0. Each bit in the S-Register represents approximately 6% CPU utilization. For example, a CPU with 70% CPU utilization has bits 15 through 4 set.

```
+--------+
| LEVEL  |
|  60    |
+--------+
```

DN (Down a Device or I/O Controller)

Declares a device or I/O controller down (i.e., unavailable for use by the RTE system).

```
+-------------------------------------------------------------------+
|                                                                   |
| DN,,lu    <---------Downs an LU                                   |
|                                                                   |
|   or                                                              |
|                                                                   |
| DN,eqt    <---------Downs an EQT                                  |
|                                                                   |
|-------------------------------------------------------------------|
|                                                                   |
| lu      Specifies the system LU of the device to be set down.     |
|                                                                   |
| eqt     Specifies the Equipment Table (EQT) entry number of the   |
|         I/O controller to be set down.                            |
|                                                                   |
+-------------------------------------------------------------------+
```

EXAMPLE:

```
  S=07 COMMAND ?LU,9 <----Find out what EQT system LU 9 points to.

  LU  #  9 = E 9
               ^
               |--EQT 9


  S=07 COMMAND ?LU,32 <--Find out what EQT system LU 32 points to.

  LU  # 32 = E 9 S 1
               ^   ^
               |   |--subchannel 1
               |
               |--EQT 9
```

Both system LUs 9 and 32 point to EQT 9. If it is desired to selectively down one of these devices and not the other, the second format for the DN command can be used:

```
S=07 COMMAND ?DN,,32
```

System LU 9 will still be up (available for use) in this case. If all devices pointing to EQT 9 are desired to be downed the first format for the DN command can be used:

```
S=07 COMMAND ?DN,9
```

COMMENTS:

Setting the device (LU) down will make only the specific device unavailable. However, all other LUs pointing to the device will also be set down. Other devices using the device's I/O select code are unaffected.

Setting an I/O controller (EQT entry) down effectively sets all devices connected to the I/O channel down by blocking any I/O operations on the select code. The state of the devices (LUs) associated with the select code are unchanged.

The device or I/O controller remains unavailable until the I/O controller is set up by the UP command. The operator might set a device down because of equipment problems, tape change, etc.

```
+-------+
| LEVEL |
|  10   |
+-------+
```

EQ (Status)

Prints the description and status of an I/O controller, as recorded in the Equipment Table (EQT) entry.

```
+---------------------------------------------------------------+
|                                                               |
| EQ,eqt                                                        |
|                                                               |
|---------------------------------------------------------------|
|                                                               |
| eqt   Equipment Table (EQT) entry number of an I/O controller.|
|                                                               |
+---------------------------------------------------------------+
```

EXAMPLE:

```
S=07 COMMAND ?EQ,5

  21 DV.05 0 B U  1 2
  ^   ^    ^ ^    ^ ^
  |   |    | |    | |
  |   |    | |    | |--status (refer to the comments below).
  |   |    | |    |
  |   |    | |    |--last subchannel addressed.
  |   |    | |
  |   |    | |--automatic output buffering used.
  |   |    |
  |   |    |--DMA not required.
  |   |
  |   |--driver routine.
  |
  |--I/O select code.
```

COMMENTS:

The status information is printed as:

```
   select code  DV.nn  D  B  Unnn  status
```

where:

select code is the I/O select code number.

DV.nn is the driver routine.

D is D if DMA required; 0 if not.

B is B if automatic output buffering is used; 0 if not.

Unnn is the last subchannel addressed.

status is the logical status:

0 = available
1 = I/O controller unavailable (down)
2 = I/O controller unavailable (busy)
3 = waiting for DMA assignment

Note that if the specified EQT is 0, it is a bit bucket, as is any LU associated with it.

```
                                                        +-------+
                                                        | LEVEL |
                                                        |  60   |
                                                        +-------+
```

EQ (Buffering)

Changes the automatic buffering designation for a particular I/O controller.

```
+------------------------------------------------------------------+
|                                                                  |
|          ,UN                                                     |
| EQ,eqt <                                                         |
|          ,BU                                                     |
|                                                                  |
|------------------------------------------------------------------|
|                                                                  |
| eqt  Equipment Table (EQT) entry number of the I/O controller.   |
|                                                                  |
| UN   Turns off buffering (UNbuffer).                             |
|                                                                  |
| BU   Turns on buffering (BUffer).                                |
|                                                                  |
+------------------------------------------------------------------+
```

EXAMPLE:

```
  S=07 COMMAND ?EQ,5      <------------------Check status of EQT 5.

    21 DV.05 0 B U  1 2
               ^
               |------------------automatic output buffering used.

  S=07 COMMAND ?EQ,5,UN  <---------------------Turn off buffering.

  S=07 COMMAND ?EQ,5      <--------------Check new status of EQT 5.

    21 DV.05 0 0 U  1 2
               ^
               |------------automatic output buffering turned off.
```

COMMENTS:

When the system is rebooted from disc, all buffering designations are reset to the values originally specified during generation.

LEVEL
10

FL (Flush Terminal Buffer)

Eliminates buffered output to an auxiliary terminal.

FL
No parameters required.

EXAMPLE:

```
S=07 COMMAND ?FL
```

COMMENTS:

The FL command is only valid when in break-mode. It is illegal if entered from the system console unless it has been enabled to run under session control.

Another method for clearing the buffer is using EXEC calls (refer to the RTE-6/VM Programmer's Reference Manual for descriptions of the EXEC calls):

```
CALL EXEC(3,2300B+lu)
```

where lu is the LU of the terminal to be flushed (normally LU 1 when under session control). A high priority program could use this call prior to writing an important message to the terminal. Note that the program issuing the call must have a priority of 40 or higher, and must have a priority higher than the program that generated the data to be flushed.

```
+-------+
| LEVEL |
| 30/60 |
+-------+
```

GO (Reschedule Program)

Reschedules a program previously suspended by the SS command or a suspend EXEC call. Users with capability levels of 30 or higher can reschedule any program suspended within their session, and users with a capability level of 60 or higher can reschedule any suspended program in the system.

```
+-----------------------------------------------------------------+
|                                                                 |
| GO,program [,p1 [,... [,p5 ]]]]]     <---passes command string. |
|                                                                 |
|    or                                                           |
|                                                                 |
| GOIH,program [,p1 [,... [,p5 ]]]]]  <---inhibits passing   of  |
|                                          command string.        |
|    or                                                           |
|                                                                 |
| GO                                    <---reschedule the current|
|                                           session program.      |
|-----------------------------------------------------------------|
|                                                                 |
| program      Name  of suspended program  to be rescheduled  for |
|              execution; default is current session program.     |
|                                                                 |
| p1 ... p5    Parameters to be passed to program; only passed if |
|              program  has suspended itself  through  EXEC call, |
|              ignored if program was operator suspended with the |
|              SS command.                                        |
|                                                                 |
+-----------------------------------------------------------------+
```

EXAMPLES:

```
1. S=07 COMMAND ?SS,PROGA    <---User suspends program PROGA which
                                 was scheduled within his session.

   S=07 COMMAND ?GO,PROGA    <--User reschedules PROGA to execute.


2. S=07 COMMAND ?GO,PROGB,1,3 <--User reschedules PROGB  which had
                                 suspended itself via an EXEC call
                                 and passes the constants 1  and 3
                                 to the program.


3. S=07 COMMAND ?RU,PROGA <-----User A schedules PROGA to execute.

   S=07 COMMAND ?SS,PROGA <---User B, who has a command capability
                              level of 60, suspends PROGA.

   S=07 COMMAND ?GO,PROGA <---User B reschedules PROGA to execute.


4. S=07 COMMAND ?SS       <-------Suspend current session program.

   S=07 COMMAND ?GO       <----Reschedule current session program.
```

COMMENTS:

The GO command is illegal if the program has not been suspended previously by the operator or has not suspended itself.

Parameters pl through p5 can be entered in ASCII or numeric form. Octal numbers are designated by the "B" suffix and negative numbers by a leading minus sign. For example:

```
GO,program,FI,LE,31061B
```

Note that only two ASCII characters per parameter will be returned by a RMPAR subroutine call: if one is given, the second character is passed as a blank (blank = 40B). If the first parameter is ASCII "NO" it must then be repeated. The system interprets it as "NOW" in the GO command, and reads the next parameter as the first parameter to be passed. For example:

```
GO,program,NO,NO,FI,3,4,5
```

is interpreted as shown below.

```
NO
FI
3
4
5
```

After a program has suspended itself and is restarted with the GO command, the parameters passed by GO are stored in the TEMP area of the program's ID segment. An immediate call to library subroutine RMPAR (refer to RTE-6/VM Library Manual for details on RMPAR) retrieves the parameters. If the program has not suspended itself (via SS command), the parameters are ignored.

The program may also recover the ASCII command string (up to 80 characters typed after the prompt) that scheduled it by using the String Passage EXEC call (refer to the RTE-6/VM Programmer's Reference Manual for a description of this EXEC call). If the program was rescheduled with a GOIH (inhibit string passage) or if the program has not suspended itself, the command string is not passed.

The current session program is the last son of the session progenitor (FMGxx) or FMGxx itself.

```
+--------+
| LEVEL  |
|   1    |
+--------+
```

HE (HELP Function)

Provides a detailed explanation of an error and guidance in possible corrective action.

```
+------------------------------------------------------------------+
|                                                                  |
| HELP [,keyword [,lu ]]                                           |
|   --                                                             |
|                                                                  |
|------------------------------------------------------------------|
|                                                                  |
| keyword   A select group of eight or less characters identify-   |
|           ing the error  for which  explanation  is requested.   |
|           The keyword may be an error code itself. Its default   |
|           is the  last  error  posted  to the  user's  Session   |
|           Control Block (SCB).                                   |
|                                                                  |
| lu        LU  of  device  where  explanation  will  be output.   |
|           Default is session user's terminal.                    |
|                                                                  |
+------------------------------------------------------------------+
```

EXAMPLES:

```
  S=07 COMMAND ?HE <-----Displays explanation of last error posted
                         to user's Session Control Block (SCB).

  S=07 COMMAND ?HE,FMGR-006,6 <-Displays explanation of FMGR error
                                code -006 on line printer (LU 6).
```

COMMENTS:

All keywords and their corresponding explanations are contained in a disc-resident HELP file. A standard list of keywords provided with RTE-6/VM include the error codes of several systems, subsystems and utility programs.

One of the features of the HELP function is that the System Manager can modify existing entries in the HELP file and can create additional keywords and add them to the HELP file. This feature is further discussed in the RTE-6/VM System Manager's Manual. A user can also generate additional keywords and explanations for any interactive program using the CMD and GENIX utilities. For more details on the Help utilities, refer to the RTE-6/VM Utility Programs Reference Manual.

HELP may also be invoked by the File Manager HE command or by running the program HELP. For example:

```
   :RU,HELP[,keyword[,lu]]

     or

   S=xx COMMAND ?RU,HELP[,keyword[,lu]]

     or

   S=xx COMMAND ?ON,HELP[,keyword[,lu]]
```

where xx is the session identification number which the system uses to identify each session.

The break-mode HE Command is only available in the session environment.

```
+-------+
| LEVEL |
|  50   |
+-------+
```

IT (Interval Timer)

Sets time values for a program so that it automatically executes at selected times when scheduled with the ON command.

```
+------------------------------------------------------------------+
|                                                                  |
| 1. Put program into time list:                                   |
|                                                                  |
|    IT,program,res,mpt [,hr [,min [,sec [,ms ]]]]                 |
|                  \--v--/ \-----------v------------/              |
|                    set      set initial start time               |
|                  interval                                        |
|                   timer                                          |
|                                                                  |
| 2. Take program out of time list (program must be dormant):      |
|                                                                  |
|    IT,program                                                    |
|                                                                  |
|------------------------------------------------------------------|
|                                                                  |
| program          Name of the program to be placed in time list.  |
|                                                                  |
| res              Designates resolution code:                     |
|                                                                  |
|                     1 - tens of milliseconds                     |
|                     2 - seconds                                  |
|                     3 - minutes                                  |
|                     4 - hours                                    |
|                                                                  |
| mpt              Multiplier used in conjunction with resolution  |
|                  code is set time interval;  number can be from  |
|                  0 to 4095. If 0 is specified, the program runs  |
|                  only once.                                      |
|                                                                  |
| hr,min,sec,ms    Parameters  set initial start time in terms of  |
|                  hour, minute, second, and tens of milliseconds.|
|                  Default  for  any  parameter  not  specified is |
|                  zero (0).                                       |
|                                                                  |
+------------------------------------------------------------------+
```

EXAMPLES:

1. Schedule program PROGA to execute for the first time at 1:00 p.m. and run every hour after that.

```
S=07 COMMAND ?IT,PROGA,4,1,13
S=07 COMMAND ?ON,PROGA
```

2. Schedule program PROGB to execute now and run every 15 minutes.

```
S=07 COMMAND ?IT,PROGB,3,15
S=07 COMMAND ?ON,PROGB,NOW
```

3. Take program PROGB out of the time list (i.e., it will no longer run every 15 minutes). PROGB is dormant.

```
S=07 COMMAND ?IT,PROGB
```

COMMENTS:

The resolution code (res) is the units in time to be multiplied by the execution interval value (mpt) to get the total time interval. Thus, if res=2 and mpt=100, the program specified would be scheduled every 100 seconds. If hr,min,sec and ms are present, the first execution occurs at the initial start time specified by these parameters (the program must be initialized with the ON command).

When the system is rebooted from the disc, time values set by the IT command are lost, and the original time values set at original load time are reinstated.

Format 1 of the IT command puts the program into the time list. This is a list of all programs which are to be executed at specific times in the future. When the time for the program to execute arrives, the program is also linked to the scheduled list.

When the session user logs off, any programs placed in the time list during the session will be removed.

The IT command is similar to the Execution Time EXEC Call (refer to the RTE-6/VM Programmer's Reference Manual for a description of this EXEC call).

Refer to the Time List description in this chapter for access restrictions to time scheduled session programs.

```
+-------+
| LEVEL |
|  60   |
+-------+
```

LU (Assignment or Reassignment)

Can print the EQT entry number, device subchannel number, and I/O device status currently associated with a system LU, or can change the EQT and subchannel associated with a system LU.

```
+------------------------------------------------------------------+
|                                                                  |
| LU,lu                       <---Prints current LU assignment     |
|                                 and status.                      |
|                                                                  |
| LU,lu,eqt [,subchannel ]    <---Changes LU assignment.           |
|                                                                  |
|------------------------------------------------------------------|
|                                                                  |
| lu          System LU for which assignment information is        |
|             desired or for which reassignment is desired.        |
|                                                                  |
| eqt         Equipment Table (EQT) entry number to assign LU. If  |
|             zero (0) is specified, lu becomes the bit bucket.    |
|                                                                  |
| subchannel  Subchannel number (0 to 63) to assign lu.            |
|                                                                  |
+------------------------------------------------------------------+
```

EXAMPLES:

```
1. S=07 COMMAND ?LU,7 <---- Print current LU assignment and status
                            for system LU 7.
   LU #  7 = E 12 S  1 D
         ^      ^    ^ ^

         |      |    | |--Device status (down in this case).
         |      |    |
         |      |    |--Subchannel number
         |      |
         |      |--EQT number
         |
         |--System LU.
```

If the LUs device is unavailable (down), a D is printed as the status; otherwise the position is left blank.

```
2. S=07 COMMAND ?LU,8,0 <---------------Make LU 8 the bit bucket.
```

COMMENTS:

The SL command also gives EQT and device status for an LU. Information on down devices can also be obtained with the WH command.

The SL command is only available under session control. It uses session LUs rather than system LUs to specify a device for which information is desired. Refer to the SL command for more details.

Restrictions for changing system LU assignments are:

a. LU 1 (system console) must be an interactive console device.

b. LU 2 (system disc) and LU 3 (auxiliary disc) cannot be changed to another EQT entry number.

c. An LU cannot be changed to point at the same device as LU 2 or LU 3.

When an irrecoverable problem occurs on an I/O device, the operator can bypass the downed device for future requests by reassigning the LU to an operable device on another select code or essentially flushing the command by reassigning it to the bit bucket.

When the system is rebooted from the disc, all LU assignments are reset to those originally established during generation.

```
+-------+
| LEVEL |
| 30/60 |
+-------+
```

OF (Terminate Program)

Terminates a program. A user with a capability level of 30 or higher, can terminate any program within caller's current session. The user with a capability level of 60 or higher can terminate any program in the system.

```
+-------------------------------------------------------------------+
|                                                                   |
| OF,program,0        <--If I/O suspended, waits for completion of  |
|    or                  I/O before terminating; program's disc     |
|                        tracks are not released.                   |
|                                                                   |
| OF,program,1[,NP] <--Immediately terminates program; releases    |
|    or                  program's disc tracks.                     |
|                                                                   |
| OF,program,8[,NP] <--Immediately terminates program; releases    |
|                        program's disc tracks.  If the program is  |
|    or                  a temporary program loaded on-line, it is  |
|                        removed from the system.   (ID segment is  |
|                        released.)                                 |
|                                                                   |
| OF                  <--Performs an OF, program, 1 on the current  |
|                        session program.                           |
|-------------------------------------------------------------------|
|                                                                   |
| program   Name of the program to be terminated.                   |
|                                                                   |
| NP        No print option (refer to the comments below).          |
|                                                                   |
+-------------------------------------------------------------------+
```

EXAMPLES:

1. Purge a temporary program (i.e., one that was loaded on-line with the LOADR) from the system.

   ```
   S=07 COMMAND ?OF,PROG,8
   ```

   If an attempt is later made to schedule PROG, a FMGR-067 error (program not found) will result. To run the program, it will first have to be re-loaded into the system.

2. Immediately terminate a temporary program, but do not purge it from the system.

   ```
   S=07 COMMAND ?OF,PROG,1
   ```

   PROG can later be run without having to re-load it on the system.

3. Terminate a program after it finishes I/O processing.

   ```
   S=07 COMMAND ?OF,PROG
         or
   S=07 COMMAND ?OF,PROG,0
   ```

COMMENTS:

For options 1 and 8 of the OF command, the message "PROGRAM ABORTED" will be displayed for programs (but not segments) after the command is executed; unless the no print option is specified. If the no print option is specified, no program abortion message is printed. If the program is I/O suspended, a system-generated request to clear the device is issued to the driver and the program is immediately terminated.

For programs with segments, the OF,program,8 command must be used on the segments as well as the main to remove them from the system.

OF,program,8 will not remove permanently loaded programs, since their ID segments on the disc are not altered by this request. A permanently loaded program is defined as a program loaded during generation, or on-line with the LOADR or MLLDR as permanent. For temporary programs loaded on-line, the ID segment is blanked to make it available for use by another program loaded with the LOADR or MLLDR. A permanently loaded disc-resident program may only be removed permanently with either loader as described in the RTE-6/VM Loader Reference Manual.

If the program is I/O suspended, a system generated clear request is issued to the driver. The OF,program,8 command must then be entered a second time to permanently remove the program from the system.

The current session program is determined by finding the last son of the session progenitor (FMGxx). If a son cannot be found, the break flag is set in FMGxx's ID segment.

```
+-------+
| LEVEL |
|  50   |
+-------+
```

ON (Schedule a Program)

Schedules a program for execution. Up to five parameters and the command string may be passed to the program.

```
+-----------------------------------------------------------------+
| ON,program [,NOW] [,p1 [,... [,p5 ]]]]] <-------Passes command  |
|                  -                             string.          |
|       or                                                        |
|                                                                 |
| ONIH,program [,NOW] [,p1 [,... [,p5 ]]]]] <--Inhibits  passing  |
|                    -                        of command string.|
|-----------------------------------------------------------------|
| program     Name of program to be scheduled.                    |
|                                                                 |
| NOW         Schedules  a program immediately  that  is normally |
|             scheduled  by the  system  clock  (refer  to the IT |
|             command).   If program  is placed  in the time list |
|             (in  conjunction  with  the  IT  command),  but not |
|             scheduled  for immediate execution,  this parameter |
|             and its preceding comma are omitted.                |
|                                                                 |
| p1 ... p5   Parameters   passed  to  the  program  when  it  is |
|             scheduled.                                          |
+-----------------------------------------------------------------+
```

EXAMPLES:

1. Schedule program PROG to run every five minutes starting from now.

   ```
   S=07 COMMAND ?IT,PROG,3,5
   S=07 COMMAND ?ON,PROG,NOW
   ```

2. Schedule program PROG to run every hour starting at midnight. Pass the ASCII characters AA, BB, CC, DD and EE to the program.

   ```
   S=07 COMMAND ?IT,PROG,4,1
   S=07 COMMAND ?ON,PROG,AA,BB,CC,DD,EE
   ```

   Program PROG can pick-up the parameters by making a call to the subroutine RMPAR. Refer to the RTE-6/VM Library Manual for a description of RMPAR.

3. Schedule program PROG to run every hour starting at midnight. Pass the ASCII string MESSAGESTRING to the program.

```
S=07 COMMAND ?IT,PROG,4,1
S=07 COMMAND ?ON,PROG,MESSAGESTRING
```

   Program PROG can pick-up the ASCII string by making a call to the subroutine GETST. Refer to the RTE-6/VM Library Manual for a description of GETST.

COMMENTS:

Parameters pl through p5, which are retrieved by an immediate call to RMPAR, are stored in XTEMP words 1 through 5 in the program's ID segment (refer to Appendix C for the format of the ID segment). Note that under Session Monitor any parameters not entered as part of the ON command will be returned as zeroes by a call to RMPAR. If the first parameter is 0, the terminal LU from which the ON command was issued will be passed to the first parameter by RMPAR (always 1 if under session). If a blank is entered before the ON command, the terminal LU is not passed by RMPAR.

Parameters pl through p5 can be entered in ASCII or numeric form. Octal numbers are designated by the "B" suffix and negative numbers by a leading minus sign. For example:

```
ON,program,FI,LE,31061B
```

Note that only two ASCII characters per parameter will be returned by a RMPAR subroutine call; if only one is given, the second character is passed as a blank (blank=40B). If the first parameter is ASCII "NO" then it must be repeated (the system interprets it as "NOW" in the ON command). For example:

```
ON,program,NO,NO,FI,3,4,5
```

is interpreted as

```
NO
FI
3
4
5
```

The program can recover the ASCII command string (up to 80 characters typed after the prompt) by using the String Passage EXEC call (refer to the RTE-6/VM Programmer's Reference Manual for a description of this EXEC call). The ONIH command inhibits the passage of the command string. The string, if passed, is stored in a buffer in System Available Memory (SAM).

If the resolution code in the ID segment of the program is not zero, RTE places the program in the time list for execution at specified times (unless NOW appears; in which case, the program is scheduled immediately). The resolution may be non-zero as a result of:

a. Generation

   1. With a resolution code in the NAM record.
   2. Entry of a resolution code during parameter input phase.

b. The IT command.

c. Scheduling the program with absolute start time or offset by some program in the system.

Note that if there is no partition large enough to run the program, or if the program is assigned to a partition that is too small or does not exist, the error message 'SIZE ERROR' will be reported. A condition under which the error message could be output when attempting to run might be:

```
:SP,program
```

Reboot and reconfigure memory to remove partitions large enough for this program.

```
:RU,program
```

Note that a session user may not run a program while it is time scheduled in another session. Only that session which has it time scheduled may run the program. Refer to the Time List description in this chapter for more details.

LEVEL
50

PR (Change Program Priority)

Changes the priority of a program.

```
PR,program,new priority
```

program	Name of program whose priority is to be changed.
new priority	New priority number to be assigned to program; 1 is the highest priority and 32767 is the lowest.

EXAMPLE:

```
  S=07 COMMAND ?PR,PROGA,40
```

COMMENTS:

When the system is restarted from the disc, the priority of "program" resets to the value set by the generator, LOADR, or MLLDR.

```
+--------+
| LEVEL  |
| 10/60  |
+--------+
```

QU (Timeslice Quantum)

Allows a user with a command capability level of 10 or greater to examine the current system timeslice quantum and the priority level at which timeslicing begins. A user with the capability level of 60 may change the timeslice parameters with this command.

```
+-----------------------------------------------------------------+
|                                                                 |
| QU [,quantum [,limit ]]                                         |
|                                                                 |
|-----------------------------------------------------------------|
|                                                                 |
| quantum   New system slice quantum;  value  must lie between 0  |
|           and 32767 milliseconds (default is 1500).             |
|                                                                 |
| limit     Priority level  at which timeslicing begins; default  |
|           is  50.   All  programs  of equal   or lower priority |
|           (higher priority number) will be timesliced.          |
|                                                                 |
+-----------------------------------------------------------------+
```

EXAMPLES:

1. Examine current system timeslice quantum and limit.

```
   S=07 COMMAND ?QU
   Q=1500 P=50
        ^     ^
        |     |--Program priority level at which timeslicing starts.
        |
        |--System timeslice quantum (1.5 seconds).
```

2. Change system timeslice quantum to 2.0 seconds and timeslice limit to a priority level of 100.

```
   S=07 COMMAND ?QU,2000,100
```

COMMENTS:

A program's slice quantum is a function of the program's priority and the system slice quantum. The system slice quantum is the minimum slice allowed. A multiplier is formed from the program priority and is included in the following equation:

Program Quantum = System Quantum * (1 + Priority Multiplier)

The lower the priority (higher number), the larger the priority multiplier. This provides larger quantums for processes which execute infrequently.

The QU command allows high level users to increase or decrease the system slice quantum and/or the priority at which programs are timesliced. This command also allows the user to turn off timeslicing (QU,0 or QU,,32767).

For more details in selecting an efficient system timeslice quantum and limit, refer to the RTE-6/VM System Manager's Manual.

```
+--------+
| LEVEL  |
|  10    |
+--------+
```

RS (Restart Session Copy of FMGR)

Aborts and reschedules a session's copy of FMGR.

```
+---------------------------------------------------------------+
|                                                               |
| RS                                                            |
|                                                               |
|---------------------------------------------------------------|
|                                                               |
| No parameters required.                                       |
|                                                               |
+---------------------------------------------------------------+
```

EXAMPLES:

1. While a user is listing an ASCII file on the line printer, the printer goes down causing the user's copy of FMGR to be suspended.

```
   :LL,6   <---File Manager command  to change list device  to line
              printer (LU 6).

   :LI,FILEA <---File Manager  command to list  file FILEA  on the
                 list device.

  IONR L*  6 E  6 S 0 *** <---System error message indicating that
                              LU 6 is down (I/O not ready).

  S=07 COMMAND ?RS <--Abort and reschedule session's copy of FMGR.
```

2. A user offs his copy of FMGR and then wants it back (for instance, so that he can log off).

```
   :OF,FMG07 <-----File Manager command to off session 7's copy of
                   FMGR.

   S=07 COMMAND ?RS <---------------------------Reschedules FMG07.
```

COMMENTS:

This command is especially useful for situations such as in the first example above.

```
+--------+
| LEVEL  |
|  30    |
+--------+
```

RT (Release Disc Tracks)

Releases all disc tracks (on LU 2 and LU 3) assigned locally to a program.

```
+-----------------------------------------------------------------+
|                                                                 |
| RT,program                                                      |
|                                                                 |
|-----------------------------------------------------------------|
|                                                                 |
| program   Name of the program whose assigned local disc tracks |
|           are to be released.                                   |
|                                                                 |
+-----------------------------------------------------------------+
```

EXAMPLE:

```
  S=07 COMMAND ?RT,PROGA
```

COMMENTS:

A program can request local disc tracks on LU 2 or LU 3 through an EXEC 4 call. For more details on EXEC calls, refer to the RTE-6/VM Programmer's Reference Manual.

The RT command is illegal if the named program is not dormant. If the program is dormant, all local tracks assigned to the program are released.

Any tracks released as a result of this command cause all programs in disc track allocation suspension to be rescheduled. More information on disc tracks may be obtained from the utility program LGTAT, described in RTE-6/VM Utility Programs Reference Manual.

```
+--------+
| LEVEL  |
|  30    |
+--------+
```

RU (Run a Program)

Immediately schedules a program for execution. If the program is currently in the time list, its entry is not affected. Up to five parameters and the command string may be passed to the program.

```
+-------------------------------------------------------------------+
|                                                                   |
| RU,program [,pl [,... [,p5 ]]]]]     <---Passes command string.   |
|                                                                   |
|     or                                                            |
|                                                                   |
| RUIH,program [,pl [,... [,p5 ]]]]]   <---Inhibits  passing  of    |
|                                          command string.          |
|                                                                   |
|-------------------------------------------------------------------|
|                                                                   |
| program      Name of program to be run.                           |
|                                                                   |
| pl ... p5    Parameters  passed  to  the  program  when  it is    |
|              scheduled.                                           |
|                                                                   |
+-------------------------------------------------------------------+
```

EXAMPLES:

1. Immediately schedule program PROG passing it the ASCII characters AA, BB, CC, DD, and EE.

   S=07 COMMAND ?RU,PROG,AA,BB,CC,DD,EE

   Program PROG can pick-up the parameters by making a call to the subroutine RMPAR. Refer to the RTE-6/VM Library Manual for a description of RMPAR.

2. Immediately schedule program PROG passing it the ASCII string MESSAGESTRING.

   S=07 COMMAND ?RU,PROG,MESSAGESTRING

   Program PROG can pick-up the ASCII string by making a call to the subroutine GETST. Refer to the RTE-6/VM Library Manual for a description of GETST.

COMMENTS:

The parameters pl through p5, which are retrieved by an immediate call to RMPAR, are stored in XTEMP words 1 through 5 in the program's ID segment (refer to Appendix C for the format of the ID segment). Note that under Session Monitor any parameters not entered as part of the RU command will be returned as zeroes by a call to RMPAR. If the first parameter is 0, the terminal LU from which the RU command was issued will be passed to the first parameter by RMPAR (always 1 if under session). If a blank is entered before the RU command, the terminal LU is not passed by RMPAR.

Parameters pl through p5 can be entered in ASCII or numeric form. Octal numbers are designated by the "B" suffix and negative numbers by a leading minus sign. For example:

```
   RU,program,FI,LE,31061B
```

Note that only two ASCII characters per parameter will be returned by a RMPAR subroutine call; if only one is given, the second character is passed as a blank (blank=40B). If the first parameter is ASCII "NO" then it must be repeated (this is because the RU command is related to the "ON,program,NOW" command; refer to the ON command for details).

The program can recover the ASCII command string (up to 80 characters typed after the prompt) by using the String Passage EXEC call. The RUIH command inhibits the passage of the command string. If there are no characters passed to the program, the command string is not transmitted. The string, if passed, is stored in a buffer in System Available Memory (SAM).

The RU command is used when the operator desires to run a program without affecting its entry in the time list. It is similar to the File Manager RU command except that it does not do automatic program renaming (i.e., the break-mode RU command actually runs "program" not a copy of "program").

Note that a session user may not run a program while it is time scheduled in another session. Only that session which has the program time scheduled may run the program. Refer to the Time List description in this chapter for more details.

```
+--------+
| LEVEL  |
|  10    |
+--------+
```

SL (Display Session LU Information)

Displays the corresponding system LU, Equipment Table (EQT) entry number and subchannel number for either a specified session LU or all session LUs in a user's Session Switch Table (SST).

```
+-----------------------------------------------------------------+
|                                                                 |
| SL [,lu]                                                        |
|                                                                 |
|-----------------------------------------------------------------|
|                                                                 |
| lu   Session LU for which linkage information is desired.  The |
|      default is to list information for all session LUs in the |
|      caller's Session Switch Table (SST).                       |
|                                                                 |
+-----------------------------------------------------------------+
```

EXAMPLES:

```
S=07 COMMAND ?SL,14 <-----------Display linkage information for
                                session LU 14.

SLU  14=LU  # 14 = E  1 S 5 D
 ^       ^         ^    ^   ^--device status.
 |       |         |    |
 |       |         |    |--subchannel number.
 |       |         |
 |       |         |--EQT number.
 |       |
 |       |--System LU.
 |
 |--Session LU.
```

```
S=07 COMMAND ?SL <----------Display linkage information for all
                            session LUs in user's SST.

SLU   1=LU  #  1 = E  7
SLU   2=LU  #  2 = E  1
SLU   3=LU  #  3 = E  1 S 6
SLU   4=LU  # 29 = E  7 S 1
SLU   5=LU  # 30 = E  7 S 2
SLU   6=LU  #  6 = E  6
SLU   7=LU  # 10 = E 10
SLU   8=LU  #  8 = E  8
SLU  14=LU  # 14 = E  1 S 5 D
```

COMMENTS:

The session LU information is always displayed on the user's terminal. Use of the File Manager LL command will not switch the output to another device such as the line printer.

```
+--------+
| LEVEL  |
| 30/60  |
+--------+
```

SS (Operator Suspend)

Allows user to suspend a non-dormant program. Users with a command capability level of 30 or higher can suspend any program which was scheduled within their session. Users with a capability level of 60 can suspend any program in the system.

```
+-----------------------------------------------------------------+
|                                                                 |
| SS[,program]                                                    |
|                                                                 |
|-----------------------------------------------------------------|
|                                                                 |
| program   Name of the program to suspended. The default is the |
|           current session program.                              |
|                                                                 |
+-----------------------------------------------------------------+
```

EXAMPLES:

1. User A schedules a program, PROG, to execute. While the program is executing, user A enters break-mode and suspends the program.

   ```
   S=07 COMMAND ?RU,PROG

   S=07 COMMAND ?SS,PROG
   ```

   To reschedule the program, user A would use the GO command. To abort the program, user A would use the OF command.

2. User B schedules a program, PROGB, to execute. While it is executing, user A who has a command capability level of 60 enters break-mode and suspends PROGB.

   ```
   S=09 COMMAND ?RU,PROGB  <---User B runs PROGB from his session.

   S=07 COMMAND ?SS,PROGB  <--------User A suspends PROGB from his
                                    session.
   ```

COMMENTS:

The SS request is illegal if the program is dormant or already operator suspended. If the program is suspended for I/O, or is in the time list, RTE waits until the current state is ended and then operator-suspends the program. For any other state the program is immediately operator suspended.

The SS command is similar to the Program Suspend EXEC call (refer to the RTE-6/VM Programmer's Reference Manual for a description).

The current session program is the last son of the session progenitor (FMGxx) or FMGxx itself.

```
+--------+
| LEVEL  |
|  10    |
+--------+
```

ST (Status)

Depending upon format, can request the status (priority, current list, time values) of a named program, determine the name and partition number of the program currently executing, or print the name of the program occupying a specified partition.

```
+------------------------------------------------------------------+
|                                                                  |
| ST[,program]        <---Program status.                          |
|                                                                  |
| ST,0                <---Name of currently executing program and  |
|                         its partition number.  If no program is  |
|                         currently executing, 0 is printed.       |
|                                                                  |
| ST,partition#       <---Name of program in specified partition;  |
|                         if empty, 0 is printed.                  |
|                                                                  |
|------------------------------------------------------------------|
|                                                                  |
| program      Name of the program whose status is to be printed.  |
|                                                                  |
| partition#   Partition number defined at system generation; can  |
|              be from 1 to 64 depending upon how many partitions  |
|              are defined at system generation.  If an undefined  |
|              partition number is specified, the error NO SUCH    |
|              PROG is printed.   If  partition  is empty,  0  is  |
|              printed.                                            |
+------------------------------------------------------------------+
```

EXAMPLES:

1. Check program status of WHZAT which was placed in the time list and schedule it to run every hour with first execution scheduled for 1:00 p.m. (Refer to the IT command for placing program into time list.)

```
S=07 COMMAND ?ST,WHZAT
   41 0 4  1   1 0 0 0 T
    ^  ^ \-v-/ \--v--/ ^
    |  |   |      |    |--Program is in time list; blank if not.
    |  |   |      |--Next scheduled execution time (hr,min,sec,
    |  |   |--Time interval.                                ms).
    |  |--Current state of the program (0=dormant).
    |--Program priority.
```

2. Find name and partition number of currently executing program.

```
   S=07 COMMAND ?ST,0
   R$PN$ 2
    ^    ^
    |    |--Partition number in which program is currently
    |                                                residing.
    |--Currently executing program.
```

3. Find name of program in partition number 3.

```
   S=07 COMMAND ?ST,3
   LGOFF
    ^
    |--Name of program currently residing in partition 3.
```

COMMENTS:

The status of a program is printed on one line in a fixed format:

```
   pr s res mpt hr min sec ms T
```

where:

pr is the priority (a decimal value from 1 to 32767).

s is the current state of the program:

```
    0 = dormant
    1 = scheduled
    2 = I/O suspended
    3 = general wait
    4 = unavailable memory suspend
    5 = disc allocation suspend
    6 = operator suspend or programmed suspend
    9 = background segment
```

res, mpt, hr, min, sec, and ms are all zero (0) unless the program is scheduled by the clock (refer to the IT command in this chapter for the meaning of these terms).

The letter "T" appears when the program is currently in the time list as a result of the IT and ON commands.

The ST,0 command is only useful from the system console since from break-mode the currently executing program will always be RPN.

LEVEL 30

SZ (Program Size)

Depending on format, either prints current program size information or changes program size.

```
SZ,program <---------- Print program size information.

SZ,program,P1 <------- Change partition size requirement
                       for non-EMA program.

SZ,program,P1,P2 <---- Change EMA and MSEG size for EMA
                       program.
```

program Name of program for which size information is to be displayed or changed.

P1 New required partition size in pages for non-EMA program. For EMA programs, P1 is the new EMA size. For MLS programs (EMA or non-EMA), P1 is the maximum path length. The maximum path length is defined as the base page to the end of the longest path plus dynamic buffer space.

P2 New MSEG size for the EMA program specified.

EXAMPLES:

1. Print program size information for EMA program PROGE.

```
S=07 COMMAND ?SZ,PROGE
65211 22 200  180   2
  ^    ^  ^    ^    ^

  |    |  |    |    |--Program's MSEG size (pages).
  |    |  |    |--Program's EMA or working set size (pages).
  |    |  |--Minimum required partition size for program (pages).
  |    |--Program size (pages)
  |--Address of last word plus 1 of the program.
```

2. Change required program size for program PROGA, which is a non-EMA program, to 12 pages.

```
S=07 COMMAND ?SZ,PROGA,12
```

3. Change required EMA size for EMA program PROGE to 190 pages and MSEG size to 3 pages.

```
S=07 COMMAND ?SZ,PROGE,190,3
```

COMMENTS:

The program size information printed by the SZ command is formatted as:

```
AAAAA BB CCCC DDDD EE FFFFF
```

where:

AAAAA = the logical address of the last word plus 1 of the program. If the program is segmented, AAAAA is the address of the last word plus 1 of the largest segment (for single level segmentation programs) or the longest path (for MLS programs). If dynamic buffer space is used, this is the address of the word following the dynamic buffer space.

BB = the program size. The program size is defined as the base page to the end of code or dynamic buffer area (if used) plus two pages if VMA/EMA is used.

CCCC = minimum required partition size of the program. If the program uses EMA, BB equals the program code size plus its EMA size. For MLS programs, CCCC is the path length plus the desired dynamic buffer space.

DDDD = the program's EMA or working set size. Printed only for VMA programs.

EE = the program's MSEG size. This will only be printed for VMA/EMA programs.

FFFF = the last VMA page used by the program. This will only be printed for VMA programs.

The "SZ,program,P1" and "SZ,program,P1,P2" forms allow the user to increase or decrease the page requirements of a program. Many programs use memory after the end of the program for symbol table or data space. This space is called the "dynamic buffer area". The SZ command modifies the size of the additional memory available to the program.

Before a program's size requirements can be changed, the program must be dormant and not currently residing in a partition (i.e., it must not have terminated saving-resources or in a serially-reusable condition), and there must be at least one partition large enough to run the program at its new size.

EMA size changes are only allowed for those programs where no EMA size was specified within the program itself; that is, the default was used. An attempt to increase or decrease the EMA size in a program where the EMA size was specified within the program causes a 'SIZE ERROR n' message to be issued. All FORTRAN programs have specified EMA sizes. MSEG changes may be made for any EMA type program. The maximum MSEG size allowed is the maximum address space minus program size minus 1.

The SZ command cannot be used to modify the size parameters of a virtual memory program. However, the size of virtual memory programs can be modified by the system commands WS and VS described later in this chapter, or by MLLDR and LOADR commands. The SZ command can be used to display size information of virtual memory programs. Refer to the RTE-6/VM Loader Reference Manual for a description of the loader commands.

The following conditions will be flagged as errors and the error 'SIZE ERROR n' reported:

FOR NON-EMA PROGRAMS:

1. An attempt is made to make P1 smaller than the actual code of the program. If the program is a multilevel segmented (MLS) program, P1 is smaller than the longest path of the program. (SIZE ERROR 0)

2. An attempt is made to make P1 larger than the 32K word program address space. (SIZE ERROR 3)

3. An attempt is made to make P1 larger than the largest currently existing partition. If the program is a multilevel segmented (MLS) program, P1 is such that the resulting new partition page requirements are larger than any currently existing partition. (SIZE ERROR 3)

4. If the program is currently assigned to a partition, an attempt is made to make P1 larger than the partition size. If the assigned program is a MLS program, the resulting new partition page requirements are greater than the assigned partition size. (SIZE ERROR 3)

FOR EMA PROGRAMS:

1. An attempt is made to set P2 less than 1. (SIZE ERROR 1)

2. An attempt is made to set P2 such that the:

   program size + P2 + 1

   exceeds the maximum program address space in pages (32). If the program is an MLS program, P2 is such that P2 plus the longest path of the program exceeds the maximum program address space. (SIZE ERROR 2)

3. An attempt is made to set P1 such that the program size plus the EMA size is larger than the largest partition in the system. If the program is using shared EMA, an attempt is made to make P1 larger than the shareable EMA partition size. (SIZE ERROR 3)

4. If the program is assigned to a partition, an attempt is made to set P1 such that the program size plus the EMA size is larger than that partition. (SIZE ERROR 3)

5. An attempt is made to change MSEG size of SHEMA program. (SIZE ERROR 4)

An attempt to set P1 to a value less than 1 is reported as an input error to the SZ command.

Refer to the RTE-6/VM Loader Reference Manual for more details concerning MLS programs.

```
+--------+
| LEVEL  |
|  10    |
+--------+
```

TE (Send Message to System Console)

Allows user to send a message to the system console.

```
+-------------------------------------------------------------------+
|                                                                   |
| TE,message                                                        |
|                                                                   |
|-------------------------------------------------------------------|
|                                                                   |
| message  Message to be sent to system console; must conform to    |
|          parameter syntax rules for privileged commands.          |
|                                                                   |
+-------------------------------------------------------------------+
```

EXAMPLE:

```
   S=07 COMMAND ?TE,MESSAGE TO CONSOLE
```

COMMENTS:

The message may consist of any printable ASCII characters. It is limited by the length and number restrictions placed on any FMGR command parameters (refer to Parameter Syntax rules, Chapter 3). Note that any commas divide the message into separate parameters. A one-parameter string could be as long as 60 characters.

```
+--------+
| LEVEL  |
|  10    |
+--------+
```

TI (Print Time)

Prints the current year, day and time, as recorded in the real-time clock.

```
+-----------------------------------------------------------------+
|                                                                 |
| TI                                                              |
|                                                                 |
|-----------------------------------------------------------------|
|                                                                 |
| No parameters required.                                         |
|                                                                 |
+-----------------------------------------------------------------+
```

EXAMPLE:

```
   S=07 COMMAND ?TI
   1978 345  13  17  43
     ^   ^    ^   ^   ^
     |   |    |   |   |
     |   |    |   |   |--Seconds.
     |   |    |   |
     |   |    |   |--Minutes.
     |   |    |
     |   |    |--Hour of day.
     |   |
     |   |--Day of year.
     |
     |--Year.
```

COMMENTS:

The TI command is similar to the Time Request EXEC call (Refer to the RTE-6/VM Programmer's Reference Manual for a description of this EXEC call).

```
+--------+
| LEVEL  |
|  60    |
+--------+
```

TM (Set Real-Time Clock)

Allows user to set or reset the real-time clock.

```
+------------------------------------------------------------------+
| TM,year,day [,hr [,min [,sec ]]]                                  |
|------------------------------------------------------------------|
| year      Four-digit year.                                        |
|                                                                   |
| day       Three-digit day of the year.                            |
|                                                                   |
| hr        Two-digit hour of the day; default is 0.                |
|                                                                   |
| min       Two-digit minute of the hour; default is 0.             |
|                                                                   |
| sec       Two-digit second of the minute; default is 0.           |
+------------------------------------------------------------------+
```

EXAMPLE:

```
  S=07 COMMAND ?TM,1978,150,14,50,36   <-----------------Set time.
  S=07 COMMAND ?TI                     <---------------Print time.
  1978  150  14  50  36
    ^    ^    ^   ^   ^
    |    |    |   |   |--Seconds.
    |    |    |   |--minutes.
    |    |    |--Hour.
    |    |--Day of year.
    |--Year.
```

COMMENTS:

The TM command is entered in response to the message:

```
   SET TIME
```

which is displayed when the RTE system is booted from disc.

Enter a time value ahead of real-time. When real-time equals the entered value, press the RETURN key. The system will then be synchronized with the time of day.

When resetting the time, do not enter a time value less than the Log-on Time of the user. The TM command should not be used to reset the time if programs are in the time list.

LEVEL
10/60

TO (Device Time-Out)

Allows users with a command capability level of 10 or more to examine the current time-out value of an I/O controller. Users with a capability level of 60 can change the time-out value.

```
TO,eqt ←————————————— Print  time-out value for EQT (level 10
                      command).

TO,eqt,numb ←———————— Change time-out value for EQT (level 60
                      command).
```

eqt   Equipment Table (EQT) entry number of the I/O controller.

numb   Number of 10 millisecond intervals to be used as the time-out value; cannot be less than 500 (5 sec) for terminals driven by drivers DVR00 or DVR05. The maximum time-out value which can be specified is 32767 (5.5 min).

EXAMPLES:

1. Examine the current time-out value for a terminal with EQT#7.

```
S=07 COMMAND ?TO,7
TO#   7=12000
      ↑   ↑
      |   └—Time-out value (120 sec)
      |
      └—EQT number
```

2. Change the time-out value for the above terminal to 60 seconds.

```
S=07 COMMAND ?TO,7,6000
```

To check to see whether the time-out value has changed, the TO, eqt command must be entered to print-out the current value.

COMMENT:

The time-out value is calculated using "numb" time-base generator interrupts (the time-base generator interrupts once every 10 ms). For example, numb = 100 sets a time-out value of one second (100*10 ms =1 second). When the system is rebooted from the disc, time-out values set by TO are reset to the values originally set during generation.

When a controller times-out, a driver has the option of performing its own time-out processing or of letting the system handle it entirely.

A time-out value of zero is equivalent to not using the time-out feature for a particular controller. If a time-out parameter is not entered, its value remains zero and time-out is disabled for the controller.

If the time-out value is set to approximately 20 seconds or above (TO,eqt,2000) for a session terminal, the session will be terminated after five timeouts. If the time-out value is set below approximately 20 seconds, the session will not be terminated.

For more details on time-out processing, refer to the RTE Operating System Driver Writing Manual.

```
+--------+
| LEVEL  |
|  60    |
+--------+
```

UL (Unlock a Shareable EMA Partition)

Unlocks a shareable EMA partition which has been locked by the LKEMA subroutine, releasing it for use by other programs. A shareable EMA partition can be locked programmatically by the LKEMA library routine described in the RTE-6/VM Programmer's Reference Manual.

```
+-----------------------------------------------------------------+
|                                                                 |
|  UL, label                                                      |
|                                                                 |
|-----------------------------------------------------------------|
|                                                                 |
|  label     Label of the shareable EMA partition.                |
|                                                                 |
+-----------------------------------------------------------------+
```

EXAMPLE:

```
   S=07 COMMAND ?UL,DATA1 <--- Unlocks the shareable EMA partition
                               labeled DATA1.
```

COMMENTS:

If the number of programs actively using a locked shareable EMA partition is 0, then unlocking the partition will release it for use by other programs. If the partition with the specified label is not locked or the number of programs actively using the locked partition is greater than 0, no action is taken as a result of this command.

If there is no shareable EMA partitions defined by the label specified in the command, the following error message is displayed:

```
                  INPUT ERROR
```

A shareable EMA partition can also be unlocked programmatically by the ULEMA library routine. Refer to the RTE-6/VM Programmer's Reference Manual for a description of this routine.

```
+--------+
| LEVEL  |
|  10    |
+--------+
```

UP (Make Available)

Declares an I/O controller and all associated devices as up (i.e., available for use by the RTE system).

```
+---------------------------------------------------------------------+
|                                                                     |
| UP[,eqt]                                                            |
|                                                                     |
|---------------------------------------------------------------------|
|                                                                     |
| eqt  Equipment Table (EQT) entry number  of the I/O controller      |
|      to be re-enabled.  If not specified, re-enable the device      |
|      (EQT number) that FMGxx or one of its sons is waiting for.     |
|                                                                     |
+---------------------------------------------------------------------+
```

EXAMPLE:

```
   S=07 COMMAND ?UP,8
```

COMMENTS:

When the operator has previously set an I/O controller or device down for some reason, the condition should be corrected before using the UP command to declare the item available again. If the problem is irrecoverable, the LU command can be used to switch the assigned LU to another device for further requests (refer to the LU command in this chapter). Previous requests made to this device are switched to the new device. To prevent indefinite I/O suspension on a downed device, time-out is used. Refer to the TO command in this chapter.

The UP command places all downed devices (LUs) and the I/O controller (EQT entry) in the available state. Any I/O operations associated with downed devices are queued on the EQT entry for processing. If a device's problem has not been corrected, it will be reset down and an error message will be printed:

```
   IONR L #lu E #eqt S #sub xxx
            ^     ^      ^   ^
            |     |      |   |--Device Status
            |     |      |--Subchannel
            |     |--EQT number
            |--LU number
```

```
+--------+
| LEVEL  |
|  50    |
+--------+
```

UR (Release Reserved Partition)

Releases a partition previously reserved during generation or reconfiguration.

```
+-----------------------------------------------------------------+
|                                                                 |
| UR,partition                                                    |
|                                                                 |
|-----------------------------------------------------------------|
|                                                                 |
| partition  The number of the partition to be released (1 to 64,|
|            depending upon how many memory partitions were       |
|            defined in system generation).                       |
|                                                                 |
+-----------------------------------------------------------------+
```

EXAMPLE:

```
   S=07 COMMAND ?UR,5
```

COMMENTS:

Once the command is entered, any program that fits into the partition may run in it. Note that although partitions may be released on-line, they may not be reserved on-line, since such action could prevent a currently swapped-out program from regaining use of its system-assigned partition when it is again scheduled.

```
+--------+
| LEVEL  |
|   30   |
+--------+
```

VS (Virtual Memory Size)

Displays and modifies the size for virtual memory data (located on disc) of a specified VMA program. Refer to the RTE-6/VM Programmer's Reference Manual for a description of Virtual Memory.

```
+-----------------------------------------------------------------+
| VS,program            <----  Displays the current virtual memory |
|                               size of program (pages).           |
|                                                                  |
| VS,program,lastpg     <----  Modifies the current virtual memory |
|                               size to lastpg +1 (pages).         |
|------------------------------------------------------------------|
| program   VMA program name.                                      |
|                                                                  |
| lastpg    Requested  last page  of  virtual memory.  This value |
|           must be in the range 31 to 65535 pages. Note that the |
|           value entered  will set the virtual memory size to be |
|           one  greater.  For example,  if  the  last  page  is |
|           specified  to be 53,  then  54 pages  of VMA  will be |
|           allocated. The default value of lastpg is 8191 pages. |
+-----------------------------------------------------------------+
```

EXAMPLES:

1. Display the current virtual memory size of PROGA.

```
S=07 COMMAND ?VS,PROGA
15770  22  50  30  2  8191
  ^     ^   ^   ^   ^  ^
  |     |   |   |   |  |
  |     |   |   |   |  +--Program's last page of virtual memory.
  |     |   |   |   |
  |     |   |   |   +--Program's MSEG size (pages)
  |     |   |   |
  |     |   |   +--Program's working set size (pages)
  |     |   |
  |     |   +--Minimum required partition size for program
  |     |        (pages)
  |     +--Program size (pages)
  |
  +--Address of last word plus 1 of the program.
```

2. Modify the virtual memory size of PROGA to 50,000 pages.

```
S=07 COMMAND ?VS,PROGA,49999
```

COMMENTS:

The virtual memory size information printed by the VS command is formatted as:

```
AAAAA  BB  CCCC  DDDD  EE  FFFFF
```

where:

AAAAA = The logical address of the last word plus 1 of the program.

BB = Program page requirement.

CCCC = Minimum required partition size of the program. This is equal to the program code size plus its working set size.

DDDD = The program's working set size.

EE = The program's size (the VMA firmware uses the last two pages for the MSEG).

FFFFF = The program's virtual memory size.

In order to modify the virtual memory size of a program, the following requirements must be met:

1. The program must be a virtual memory program. If the program is not a virtual memory program then the error "ILL PGM ERR" is issued.

2. The program must be dormant and not currently residing in a partition (it must not have terminated saving-resources or in a serially-reusable condition). Note that the program must pass all the checks necessary for the SZ command.

3. The new virtual memory size must be greater than or equal to 31 pages but less than or equal to 65535 pages. If the new size is not in this range, then an "INPUT ERROR" will be issued.

The size of the virtual memory programs cannot be modified by the SZ command. However, the size of the virtual memory programs can be modified by MLLDR and LOADR commands, or the system commands WS and VS. Refer to the RTE-6/VM Loader Reference Manual for details.

If the value entered for the last page of virtual memory is between 64,512 and 65,534 pages, then the new size is rounded up to 65,535 (64K) pages.

For a description of virtual memory, refer to the RTE-6/VM Programmer's Reference Manual.

```
                                                        +--------+
                                                        | LEVEL  |
                                                        |  10    |
                                                        +--------+
```

WH (Run WHZAT Program)

Schedules the WHZAT program to display the current system status.

```
+-------------------------------------------------------------------+
|                                                                   |
| WH, [lu[,option[,prog]]] <--- displays status on specified LU.   |
|    or                                                             |
| WH, [option,prog]]      <--- displays status on input terminal.  |
|                                                                   |
|-------------------------------------------------------------------|
|                                                                   |
| lu      The LU of the device on which the information is to be   |
|         displayed (default value is the LU of the terminal       |
|         from which the command is input).                         |
|                                                                   |
| option  Indicates type of status to be displayed.  Default is    |
|         to display only those programs associated with the        |
|         user's session.                                           |
|                                                                   |
|         AL  Status of all scheduled and suspended programs.       |
|                                                                   |
|         SM  Same as AL except that all state-3 programs not       |
|             having "father-son" relationship will not be          |
|             displayed.                                            |
|                                                                   |
|         PA  Status of all partitions being used.                  |
|                                                                   |
|         PL  Status of all programs (or a specific program) in     |
|       or    the system. (Lists all the ID segments.)              |
|         PR                                                        |
|                                                                   |
| prog    Program name. Only the status of the specified program   |
|         is displayed. This parameter can only be used with the    |
|         PL or PR option.                                          |
|                                                                   |
+-------------------------------------------------------------------+
```

EXAMPLES:

```
   S=07 COMMAND ?WH <---displays status of scheduled and suspended
                        programs on input terminal.

   S=07 COMMAND ?WH,PA <---displays  status, on input terminal, of
                           all partitions being used.
```

COMMENTS:

For details on the system status program (WHZAT), Refer to the RTE-6/VM Utility Programs Reference Manual.

```
+--------+
| LEVEL  |
|  30    |
+--------+
```

WS (Working Set Size)

Displays and modifies the working set size (virtual data currently in memory) of a specified virtual memory program. Refer to the RTE-6/VM Programmer's Reference Manual for a detailed explanation of working set.

```
+-----------------------------------------------------------------+
| WS,program          <--- Displays  the current working set size |
|                            of program (pages).                  |
|                                                                 |
| WS,program,wrksz    <--- Modifies the working set size to wrksz |
|                             (pages).                            |
|-----------------------------------------------------------------+
| program     Virtual memory program name.                        |
|                                                                 |
| wrksz       Requested new working set size of program  (pages). |
|             The default working set size of 31 pages.           |
+-----------------------------------------------------------------+
```

EXAMPLES:

1. Display the current working set size of PROGA.

```
S=07 COMMAND ?WS,PROGA
65211  22  50  30  2
  ^     ^   ^   ^   ^
  |     |   |   |   |
  |     |   |   |   +--Program's MSEG size (pages)
  |     |   |   |
  |     |   |   +--Program's working set size (pages)
  |     |   |
  |     |   +--Mimimum required partition size for program (pages)
  |     |
  |     +--Required program size (pages)
  |
  +--Address of the last word plus 1 of the user's program.
```

2. Modify the working set size of PROGA.

```
S=07 COMMAND ?WS,PROGA,40
```

COMMENTS:

The working set size information printed by the WS command is formatted as:

```
     AAAAA  BB  CCCC  DDDD  EE
```

where:

AAAAA = The address of the last word plus 1 of the user's program.

BB = The required program page size.

CCCC = Minimum required partition size. This is equal to the program code size plus the working set size.

DDDD = The program's working set size.

EE = The program's MSEG size.

In order to modify the working set size of a program, the following requirements must be met:

1. The program must be an existing virtual memory program.

2. The program must be dormant and not currently resident in a partition (it must not have terminated saving-resources or in a serially-reusable condition).

3. The program must not be a segment or be memory-resident.

4. The new working set size must be greater than or equal to 10 pages, but less than or equal to the maximum partition page size minus the program code size.

   If either of the first three requirements are not met, the error "NO SUCH PROG" will be issued. In addition, the sum of the program code plus the new working set size must fit in the largest partition of the correct type and be larger than 10 pages or else the "SIZE ERROR" error message will be issued.

The working set size of a program can not be modified by the SZ command. However, the working set size of a program can be modified by MLLDR or LOADR commands. Refer to the RTE-6/VM Loader Reference Manual for more details.

For a description of virtual memory and working set, refer to the RTE-6/VM Programmer's Reference Manual.

Appendix A
Error Messages Index

FMGR Error Messages

FMGR-105
D.RTR DIRECTORY TRACK BUFFER TOO SMALL
The length of the directory buffer in the routine D.BUF was defined to be less than 512 words when loaded with D.RTR. Alter the size of the directory buffer in D.BUF by altering the constant defined by D.LEN in the source, reassemble the routine D.BUF and regenerate the system with the new relocatable.

FMGR-104
REQUESTED EXTENT IS MISSING
A request was made for a file extent that was missing from the file. The file is probably corrupt. Purge the file.

FMGR-103
FILE DIRECTORY IS CORRUPT
During a directory lock done by MC, DC, IN, PK, CR, PU, the directory is scanned for internal consistency. If this occurs, copy the files to another disc, or store the ones you need.

FMGR-102
ILLEGAL D.RTR CALL SEQUENCE
A lock was not requested first or the file was not opened exclusively. Possibly an operator error, such as removing a cartridge without dismounting it first.

FMGR-101
ILLEGAL PARAMETER IN D.RTR CALL
Possibly an operator error. Recheck the previous entries for illegal or misplaced parameters. This error can also happen when a request is made to create a scratch file and that scratch file already exists and is open to another program. If D.RTR is unable to purge the existing scratch file, this error is returned. See the system manager.

FMGR-099
DIRECTORY MANAGER EXEC REQUEST WAS ABORTED
An EXEC request made by D.RTR was aborted. Make sure that all discs being accessed are up. Notify system manager.

FMGR-052
SPOOL SHUT DOWN. SPOOL FILE SETUP FAILED
Spool shut down is in progress. A write (WR) or read/write (BO) spool file cannot be set up at this time. Start up spooling using GASP's SU command and try to set up the spool file again.

FMGR-049
COPY VERIFY FAILED
The CO command detected an error during the verification of the file just copied. The 10G global contains the name of the file which did not verify. Retry the copy by specifying the name in the 10G global as the starting file in the CO command.

FMGR-048
SPOOL NOT INITIALIZED OR SMP CANNOT BE SCHEDULED
Spooling has not been initialized (run GASP to do so); or the SMP program is not found or partitions are too small to run it. The default for SMP is real-time (type 2) sized to 6 pages.

FMGR-047
NO SESSION LU AVAILABLE FOR SPOOL FILE
If the session LU to be used for the spool file is not specified during set up, SMP allocates a session LU less than 64 that is not already used in the session switch table. Use the :SL,lu,- command to release a session LU in the spare part of your session switch table.

FMGR-046
GREATER THAN 255 EXTENTS
You cannot create a file extent numbered greater than 255; make file size of the main larger. If generated during an SM command, the message is not put in the message file, but is truncated at the last valid message.

FMGR-041
NO ROOM IN SST
There are no spare entries left in the session switch table. Spare entries can be recovered by using the :SL,lu,- command, where lu is a session logical unit number that is not needed.

FMGR-040
LU NOT FOUND IN SST
You are trying to access an LU that is not in your session switch table. Use the SL command to add the LU to your SST.

FMGR-039
SPOOL LU NOT MAPPED TO THE SPOOL DRIVER
A spool LU must point to a spool EQT. Switch all spool LUs to point to spool EQTs and try the spool file set up again.

FMGR-038
ILLEGAL SCRATCH FILE NUMBER
The range for scratch file numbers is 0 through 99. Try again with a number in the correct range.

FMGR-037
ATTEMPT TO PURGE AN ACTIVE TYPE 6 FILE
Type 6 files that have been RP'd cannot be purged. Off the program if no one is running it, and try again.

FMGR-036
LOCK ERROR ON DEVICE
A call to OPENF attempted unsuccessfully to lock a device. Perhaps the device is already locked or there are no resource numbers available.

FMGR-035
ALREADY 63 DISCS MOUNTED TO SYSTEM
There are already 63 discs mounted. A disc will have to be dismounted before a new one may be mounted.

FMGR-034
DISC ALREADY MOUNTED
A disc with this ID is already mounted in the cartridge directory. Either dismount the duplicate disc or mount a different one.

FMGR-033
NOT ENOUGH ROOM ON CARTRIDGE
The cartridge does not have enough space on it. Try using another cartridge, decreasing the file size, or packing the disc.

FMGR-032
CARTRIDGE NOT FOUND
The specified cartridge cannot be found in the cartridge list.

FMGR-030
VALUE TOO LARGE FOR PARAMETER

1. The value supplied in the parameter is beyond the defined range.
2. If a parameter supplied in a calling sequence receives information returned from the routine, the parameter must be large enough. If the parameter supplied is a single word but the value to be returned is a double word, this error results.

FMGR-026
QUEUE FULL OR MAX PENDING SPOOLS EXCEEDED
The spool queue is full or the maximum number of spools pending has been exceeded. Re-run the job when the space becomes available.

FMGR-025
NO SPLCON ROOM
The SPLCON control-record area is full. This error may occur when the spool system is competing with programs using their own spooling file and running outside of batch. Re-run the job when SPLCON control-record entry space is available.

FMGR-024
NO MORE BATCH SWITCHES
The LU switch table is full; the size of the switch table specified at system generation is inadequate. Notify the system manager.

FMGR-023
NO AVAILABLE SPOOL FILES
All spool files are currently being used. Re-run the job after a spool file becomes available.

FMGR-022
NO AVAILABLE SPOOL LU'S
All spool logical units are currently being used. Re-run the job after a spool LU becomes available.

FMGR-021
ILLEGAL DESTINATION LU
The specified LU was not allocated by GASP. Try again using an LU allocated by GASP.

FMGR-020
ILLEGAL ACCESS LU

1. The logical unit number specified in the LU or cs command was not a positive logical unit number. Re-enter the corrected command.
2. There is an LU entry in the cartridge list that does not point to a disc device. After the disc was mounted, the LU command switched the device. Switch the LU back to its disc definition. If desired, dismount the disc. The LU can then be switched to a non-disc device.

FMGR-019
ILLEGAL ACCESS ON A SYSTEM DISC
The system manager is the only user with the capability to write on a system disc.

FMGR-018
ILLEGAL LU
The LU is either not assigned to the system, or is not defined in your session switch table (SST).

FMGR-017
ILLEGAL READ/WRITE ON TYPE 0 FILE

1. Some type 0 files do not support all read, write, or position requests. For example, a spool file may be open for read-only access; attempting to write to the file will fail. Check the file parameters or the namr.
2. There is not enough space on the cartridge to write to a spool file.

FMGR-016
ILLEGAL TYPE 0 OR SIZE=0
One of the following occurred: the wrong file type was specified; a type 0 file received a purge or create request; the size specified was zero; or the new type specified in the RN command is illegal. Check the size and type parameters.

FMGR-015
ILLEGAL NAME
The file name does not conform to syntax rules; a common cause is the presence of an illegal character.

FMGR-014
DIRECTORY FULL
There is no more room in the file directory. Purge unused files and pack the disc, or else try another cartridge.

FMGR-013
DISC LOCKED
A program has locked the disc. Initialize the cartridge if it was not initialized, or wait until it becomes unlocked.

FMGR-012
EOF OR SOF ERROR
A read, write, or position attempt was made to access a file beyond its boundaries. Check the record position parameters.

FMGR-011
DCB NOT OPEN
An unopened DCB was accessed before a create or open call, or the DCB was already closed when another close was attempted.

FMGR-010
NOT ENOUGH PARAMETERS
One or more of the required parameters were omitted. Enter all necessary parameters.

FMGR-009
ATTEMPT TO USE APOSN OR FORCE TO 1 A TYPE 0 FILE
A type 0 file cannot be positioned with APOSN or be forced to a type 1 file. Check the file type.

FMGR-008
FILE OPEN OR LOCK REJECTED
A file may not be opened if it is opened exclusively to another program, opened non-exclusively to 7 programs, or if its cartridge is locked. The CL or DL commands tell which program has the disc locked. If the cartridge is being packed, check to see if spooling is shut down.

FMGR-007
ILLEGAL SECURITY CODE OR ILLEGAL WRITE ON LU 2 OR 3
1. You specified a security different from the actual one, or you specified none and one is required.
2. You tried to write on LU 2 or 3, but only the system manager can write to files on these LUs.

FMGR-006
FILE NOT FOUND
The file cannot be found. Perhaps the file name or cartridge reference you typed was different from the actual one.

FMGR-005
RECORD LENGTH ILLEGAL
The record length specified is not valid. For a read or position attempt, the record may not have been written; on a write attempt, the record length for the data is illegal. The file may be corrupt.

FMGR-004
RECORD SIZE OF TYPE 2 FILE IS 0 OR UNDEFINED
A type 2 file create attempt was made without a record size or with a size of 0.

FMGR-003
BACKSPACE ILLEGAL
The device or type 0 file cannot be backspaced. Check its device type.

FMGR-002
DUPLICATE FILE NAME
A file already exists with the name specified. Repeat the command with a new name or purge the existing file.

FMGR-001
DISC ERROR
The disc is down. Try again and then report the problem to the system manager.

FMGR 000
BREAK
This is to acknowledge the BR command. No error has occurred.

FMGR 001
DISC ERROR - LU REPORTED
The disc associated with the LU reported is down. Report the problem to the system manager. Example:

```
   FMGR 001
   FMGR 034
```

indicates that a disc error has been found on disc LU 34.

FMGR 002
INITIALIZE LU 2!
This error indicates a request for the user to initialize the system disc (LU 2) by assigning specific system tracks to FMGR. Before it is initialized, FMGR obtains all the available tracks on the system and auxiliary discs and assigns them to itself. After it is initialized, FMGR owns only those tracks specifically assigned to it. Thereafter, each time the system is loaded from disc (booted up), it recovers these tracks automatically and no further initialization is required. To initialize the system disc, use the FMGR initialize (IN) command.

For example, :IN,SC,-2,2,SYS,100 initializes LU 2. It sets the master security code to SC, LU 2's CRN to 2, its label to SYS and its starting track for FMP files to 100.

The starting FMP track must be at least 8 tracks greater than the last track used by the system. (System size is reported at the end of system generation.)

FMGR 003
INITIALIZE LU 3!
This error indicates a request for the user to initialize the auxiliary disc (LU 3) by assigning specific system tracks to FMGR. Before it is initialized, FMGR obtains all the available tracks on the system and auxiliary discs and assigns them to itself. After it is initialized, FMGR owns only those tracks specifically assigned to it. Thereafter, each time the system is loaded from disc (booted up), FMGR recovers these tracks automatically and no further initialization is required. To initialize the auxiliary disc, use the FMGR initialize (IN) command.

For example, :IN,SC,-3,3,AUX,70 initializes LU 3, setting its CRN to 3, its label to AUX and its starting track for FMP files to 70.

If auxiliary disc tracks are not to be assigned to FMGR, the IN command should still be given in response to FMGR 003, but the cartridge reference number should be specified as 0.

FMGR 004
ILLEGAL RESPONSE TO FMGR 002 OR FMGR 003
A command other than an initialize (IN) command was entered in response to either a FMGR 002 or FMGR 003 error. Enter the appropriate initialize command.

FMGR 005
REQUIRED TRACK NOT AVAILABLE - RELATIVE TAT POSITION REPORTED
The first track specified in the initialize command is not available. The starting track must be at least 8 tracks greater than the last track used by the system. Re-enter the initialize command with the first available track reported in this message.

FMGR 006
FMGR SUSPENDED
FMGR has suspended itself. Ready the down device and enter GO,FMGR.

FMGR 007
CHECKSUM ERROR
A checksum error occurred on a binary file or the file being read is not binary (type 5 or 7). Check the file type.

FMGR 008
D.RTR NOT LOADED
The program D.RTR was not found in the system. Load D.RTR as a permanent program.

FMGR 009
ID SEGMENT NOT FOUND
An RP command was used to deallocate or reassign the ID segment to the program being restored. The system looks for a blank ID segment.

FMGR 010
INPUT ERROR
A syntax error occurred. Look for a missing or extra colon, a misspelled command, an error in the subparameters, a command that is too long, etc. Re-enter the command. If received after entering an abort command, there were no active jobs.

FMGR 011
DO 'OF,XXXXX,8' ON NAMED PROGRAMS
These programs are still RP'd and their type 6 files are on the disc you tried to pack. Enter either RP,,program or OF,program,8 to remove the named programs.

FMGR 012
DUPLICATE DISC LABEL OR LU
The cartridge you tried to mount has the same label or LU number as an already-mounted cartridge (the duplicate label is displayed with the error). Re-enter the command using another label or LU, or dismount the duplicate cartridge. This error may also occur if you dismounted a private cartridge from your session and attempted to re-mount it as a group cartridge, or if you dismounted a group cartridge from your session and attempted to re-mount it as a private cartridge (which is the default).

FMGR 013
TR STACK OVERFLOW
More than 10 TR commands have been encountered without transferring back. Enter :TR,-11 to clear the stack.

FMGR 014
REQUIRED ID SEGMENT NOT FOUND
An ID segment cannot be found for the specified program; check the program name or load the program. A blank ID segment cannot be found for a program being restored; enter an OF command to release an ID segment.

FMGR 015
LS TRACK REPORT
This message reports the logical unit number and track of the current LS area; it is not an error.

FMGR 016
INSUFFICIENT SYSTEM TRACKS FOR RP
There is insufficient space in the system track pool to copy the program to RP. Either wait until more track pool space becomes available, or move the file to LU 2 or LU 3 and re-enter the command.

FMGR 017
ID SEGMENT NOT SET UP BY RP
In order for an ID segment to be released by an RP command, it must have been set up by an RP command. Try using OF,program to release it.

FMGR 018
PROGRAM NOT DORMANT
An RP,namr,program was attempted when the program is active. Enter OF,program and then repeat the RP command.

FMGR 019
FILE NOT SET UP BY SP ON CURRENT SYSTEM
The program file being restored had a parity error, was not set up correctly, or was not set up by a SP command in the current system. Reload the program and try again.

FMGR 020
ILLEGAL TYPE 0 FILE
A type 0 file to be created must point to an LU already assigned in the system. Re-enter the command using another logical unit.

FMGR 021
ILLEGAL DISC SPECIFIED
A copy to the same file or to/from an unmounted cartridge was attempted. Mount another cartridge or use one that is already mounted.

FMGR 022
COPY TERMINATED
The copy process has been terminated due to a copy error. Check the parameters and the specified discs.

FMGR 023
DUPLICATE PROGRAM NAME
The program being restored is already defined. Change the name of the program, enter OF,program, or release the ID segment.

FMGR 038
ATTEMPT TO PURGE ACTIVE FILE
This type 6 file is RP'd. Off the RP'd program and try again.

FMGR 041
PROGRAM CANNOT BE A SEGMENT
The program specified is a program segment (type 5). LS tracks cannot be assigned to a program segment. Ordinarily, the LS tracks are assigned to the program EDITR when MS is executed. If the LS tracks are to be assigned to a different program, specify this program's name.

FMGR 042
LU CANNOT BE SWITCHED
This logical unit cannot be switched. If it is a disc LU, the session LU must be the same as the system LU. Session LU 1 cannot be switched.

FMGR 043
LU NOT FOUND IN SST
This logical unit is not defined in your session switch table. Use the SL command to add the LU to your SST.

FMGR 044
NO MESSAGES WAITING
There are no messages waiting for your ME command.

FMGR 045
SESSION COMMAND ONLY
This command operates only in the session environment.

FMGR 046
INSUFFICIENT CAPABILITY
Your account's capability level is too low to execute the command (the :DP,9P command displays your level). The system manager can increase your capability level.

FMGR 047
SPOOL SET UP FAILED
There are no available spool files or LUs, or the logical unit table is full. Try running the job again. If the error is due to lack of spool LUs, or the logical unit table is full, reconfigure the system.

FMGR 048
GLOBAL SET OUT OF RANGE
The specified global is out of range. Check the parameters and re-enter the command correctly.

FMGR 049
CAN'T RUN RP'ED PROGRAM
The program restored from the file does not execute. Usually this is caused by attempting to run a segment of the specified program. Check the program.

FMGR 050
NOT ENOUGH PARAMETERS
Less than the required number of parameters were specified. Re-enter the command with all necessary parameters.

FMGR 051
ILLEGAL MASTER SECURITY CODE
The master security code you entered didn't match the stored one.

FMGR 052
ILLEGAL LU

1. Attempt to switch a session LU to a system LU which is a disc, but the session LU number does not equal the system LU number. (The system LU and session LU for a disc must be the same.)
2. An illegal LU was specified in the SL command. The LU must be positive and less than the largest LU defined in the system, and all session LUs must be less than 64.
3. Response to FMGR 002 must be to initialize LU 2, and response to FMGR 003 must be to initialize LU 3 (or 0 if no LU 3). A response other than 2 or 3 was entered.

FMGR 053
ILLEGAL LABEL OR ILABEL
The specified cartridge reference number or cartridge ID is illegal. The cartridge reference number must be a positive non-zero integer and the cartridge ID must be a legal file name.

FMGR 054
DISC NOT MOUNTED
An attempt was made to dismount or reference a disc cartridge that isn't mounted to you. To reference it, mount the disc cartridge using the MC command. If under session control, the AC command could be used instead to allocate disc space with the specified CRN. This error also occurs if an attempt is made by a session user (not the system manager) to dismount a system disc. A session user is allowed access to a system disc even though he has no control over the mounting or the dismounting of it.

FMGR 055
MISSING PARAMETER
A required parameter has been omitted. Check the command syntax and re-enter all necessary information.

FMGR 056
BAD PARAMETER
A parameter was specified incorrectly or a track parameter specifies a track that is outside the range of the FMGR tracks.

FMGR 057
BAD TRACK NOT IN FILE AREA
The specified track is in the system area or is a directory track. Correct the command and re-enter it.

FMGR 058
LG AREA EMPTY
You attempted to save the contents of an empty LG area. Use the MR command to move a file to the LG area.

FMGR 059
REPORTED TRACK UNAVAILABLE
A re-initialization attempt will lower the first track into the system area. The last track is reported. Re-enter the command with the first track equal to the last track + a minimum of 8.

FMGR 060
DO YOU REALLY WANT TO PURGE THIS DISC?
A re-initialization attempt will raise the first track or lower the directory tracks into the file area and destroy a file. Enter ?? or NO to stop the reinitialization. Enter YES to continue.

FMGR 061
DO A "DC" AND A "MC" ON THIS CR
You are attempting to replace a mounted cartridge with a cartridge that has not been previously initialized without entering the DC and MC commands. Enter the DC and MC command for this cartridge. (On the DC, be sure to use the RR option to release resources.)

FMGR 062
MORE THAN 63 DISCS
Session is limited to 63 cartridges; you tried to mount the 64th. Dismount a cartridge to make room, if possible.

FMGR 063
EXCEEDING SESSION DISC LIMIT
You attempted to mount more discs to a session than are allowed in your account. Dismount an unused disc and re-enter the command. To increase your account's disc limit, consult the system manager.

FMGR 064
NO DISC AVAILABLE FROM DISC POOL
All discs in disc pool are allocated or there are no available discs that are big enough. This error can also occur if the number of directory tracks specified is too large. A reasonable number of tracks must be specified in relation to the total number of tracks on the disc. If disc space is being allocated from the disc pool and size was not specified (i.e., first free disc is allocated), the mount routine will continue to search the disc pool until a disc is found that will pass the reasonable test. In this case, it is possible that even though there are free discs in the pool, none will be allocated because the number of directory tracks was so large.

FMGR 065
CONFLICT IN SST DEFINITION
The specified LU number is already defined as a session LU in your session switch table (SST). This will occur if you specify a disc LU number in the mount command, but this number is already defined in the SST. If it is necessary to mount this disc LU, change the conflicting entry in the SST. This can be done by using the SL command to remove the SST entry with the conflicting session LU and, if desired, re-entering it in the switch table with a different session LU number.

FMGR 066
NO ROOM IN SST
There are no spare entries left in the session switch table. Spare entries can be recovered by using the :SL,LU,- command, where LU is a session logical unit number that is not needed.

FMGR 067
PROGRAM NOT FOUND
The program to be executed was not found among the system ID segments, nor was it found as a type 6 file. Check the program name you specified or reload the program. On a HE (help) command, this indicates the program HELP could not be found. On a WH (WHZAT) command, the error indicates that WHZAT could not be found.

FMGR 068
LU NOT IN VARIABLE PART OF SST
Only LUs in the variable part of the session switch table (SST) may be deleted.

FMGR 069

JOB LOGON FAILED

The job account could not be logged on. The reason for the failure is printed on the system console.

FMGR 070

SECTORS/TRACK VALUE TOO LARGE

The sectors per track value specified in the initialize command is larger than the actual sectors per track value for the disc. Let the parameter default to the actual sectors per track value for the disc, or specify a smaller value.

FMGR 071

DO "EX,SP" TO SAVE OR "EX,RP" TO RELEASE PRIVATE CARTRIDGES

A private disc is mounted to your session; more than EX is needed to log off. Use EX,SP to save (or EX,RP to release) your private cartridges. If the disc was allocated from the disc pool, it is returned to the pool for possible re-allocation to another user. If EX,SP is specified, your private disc(s) will remain mounted to you; on the next log-on, the disc(s) will be mounted to the new session. Note that group discs are, by default, left mounted at log-off. To release group discs at log-off, use EX,,RG.

FMGR 072

LU NOT INTERACTIVE

The logical unit specified in a CT command must refer to an interactive device.

FMGR 073

ACCOUNT NOT FOUND

The account you specified does not exist. Check the User.Group name or the order of the parameters in the SM command.

FMGR 074

JO COMMAND EXPECTED

The first command in a job must be (and was not) a JO command.

FMGR 075

CANNOT RESTORE TYPE 6 PROGRAM FILE (USER PROTECTED)

The specified program is saved as a type 6 file with user protection (SP,PROG,PR). It can only be run or RP'ed from the type 6 file by the user who issued the SP command, or by users who are linked to the same account.

FMGR 076
CANNOT RESTORE TYPE 6 PROGRAM FILE (GROUP PROTECTED)
The specified program is saved as a type 6 file with group protection (SP,PROG,GR). It can only be run or RP'ed from the type 6 file by users with the same group as the user who issued the SP command.

FMGR 077
CANNOT RESTORE TYPE 6 PROGRAM FILE (INSUFFICIENT CAPABILITY)
The specified program is saved as a type 6 file with capability level protection (SP,PROG,,<cap>, where <cap> is the minimum capability level required to run or RP the program). The program can only be run or RP'ed from the type 6 file by users possessing a capability level greater than or equal to the level specified when the program was SP'ed. For example, the command SP,PROG,,50 will save program PROG and only users with a capability level of 50 or greater will be allowed to run or RP the program from the type 6 file. Note that command capability checking is still in effect. (The user still must have sufficient capability to invoke the RU or RP command, regardless of the capability level specified in the SP command.)

FMGR 078
CANNOT RESTORE TYPE 6 PROGRAM FILE (INTERNAL ERROR)
Internal consistency checks have failed while attempting to restore a program file.

FMGR 079
WARNING - RECORDS TRUNCATED TO 128 WORDS
In a type 2 file, records longer than 128 words have been truncated to 128 words.

FMGR 080
CANNOT FIND NAMED EMA
Cannot RP or run this program because the sharable EMA partition name does not exist in the system. You may either re-load this program with an existing shared EMA label or reboot and reconfigure memory to add the shareable EMA label your program references.

System and Break-Mode Command Error Messages

OP CODE ERROR

ILLEGAL OPERATOR REQUEST CODE. ENTER CORRECT OPCODE.

NO SUCH PROGRAM

THE NAME ENTERED IS NOT A MAIN PROGRAM IN THE SYSTEM. ENTER CORRECT PROGRAM NAME OR LOAD PROGRAM.

INPUT ERROR

A PARAMETER IS ILLEGAL. ENTER COMMAND WITH CORRECT PARAMETER.

ILLEGAL STATUS

PROGRAM IS ALREADY SCHEDULED. CHECK STATUS WITH "ST" COMMAND. EITHER WAIT UNTIL PROGRAM TERMINATES ITSELF OR OFF IT WITH "OF" COMMAND AND RE-ENTER "RU" COMMAND.

CMD IGNORED-NO MEM

NOT ENOUGH SYSTEM AVAILABLE MEMORY EXISTS FOR STORING THE PROGRAM'S COMMAND STRING. RE-ENTER THE COMMAND (RU,ON,GO) OR ENTER THE INHIBIT FORM (IH) OF THE COMMAND.

ILLEGAL PART'N

PARTITION DOES NOT MATCH COMMAND REQUEST. RE-ENTER COMMAND WITH CORRECT PARAMETER NUMBER.

SIZE ERROR

ILLEGAL PROGRAM SIZE SPECIFIED OR SIZE OF PROGRAM SPECIFIED LARGER THAN ITS ASSIGNED PARTITION OR ANY PARTITION. RE-ENTER COMMAND WITH CORRECT SIZE OR ADJUST PROGRAM SIZE WITH "SZ" COMMAND.

XXXXX NO SWAP TRACKS

NOT ENOUGH SWAP TRACKS AVAILABLE TO SWAP OUT A PROGRAM ON BEHALF OF XXXXX. RUN LGTAT TO DETERMINE WHO OR WHAT IS USING THE SYSTEM SCRATCH TRACK AREA.

I/O Error Messages

IO00
Illegal class number specified; outside the class table, not allocated, or bad security code.

IO01
Not enough parameters were specified.

IO02
Illegal logical unit number specified.

IO03
Illegal EQT referenced by LU in I/O call (its select code is 0).

IO04
Illegal user buffer specified: extends beyond RT/BG area or not enough system available memory to buffer the request.

IO05
Illegal disc track or sector specified.

IO06
A reference to a protected track or to unassigned LG tracks was made.

IO07
The driver has rejected the call.

IO09
LG tracks overflowed.

IO10
A class get call was issued while one call was already outstanding.

IO11
A type 4 program made an unbuffered I/O request to a driver that did not do its own mapping.

IO12
LU not defined for use by this session. The SL command reports all logical units available to your session.

IO13
LU either locked to another program, or I/O request pointed to an EQT locked to another program.

IO14
An I/O request was issued with the no-suspend option.

IO15
Buffer size of a type 6 program is greater than what will fit in the user map.

IO16
CPU backplane failure or I/O extender timing failure.

IO20
You may not read a write-only spool file. Revise the call to SPOPN or check your SL command parameters.

IO21
Cannot read End-of-file. Revise your program and re-run it.

IO22
Second attempt to read JCL card from batch input file by other than FMGR. Revise program and re-run.

IO23
You may not write to a read-only spool file. Revise your call to SPOPN or check your SL command parameters.

IO24
Write attempted beyond end-of-file; usually, spool file overflow. Obtain more spool room on the disc (using the PK command).

IO25
You tried to access a spool LU that is not currently set up. The spool may have been killed (with the GASP KS command).

IO26
I/O request made to a spool that has been terminated by the GASP KS command. Reset the session logical unit with the :SL,lu,- or :SL,lu,,,lu command.

IOET
End-of-tape condition on the specified LU. Mount a new tape and UP its EQT.

IONR
The specified LU is not ready. Make the device ready and UP its EQT.

IOTO
The specified LU timed out. Examine the device, correct the problem, and UP its EQT.

IOPE
Parity error in the data transmission from the specified LU. Examine the device, correct the problem, and set the EQT up.

ILL INT
Illegal interrupt on the specified channel.

I/O Error Messages

""

IO00

ILLEGAL CLASS NUMBER SPECIFIED. OUTSIDE TABLE, NOT ALLOCATED, OR BAD SECURITY CODE.

""

IO01

NOT ENOUGH PARAMETERS WERE SPECIFIED.

""

IO02

ILLEGAL LOGICAL UNIT NUMBER SPECIFIED.

""

IO03

ILLEGAL EQT REFERENCED BY LU IN I/O CALL (SELECT CODE=0).

""

IO04

ILLEGAL USER BUFFER SPECIFIED. EXTENDS BEYOND RT\BG AREA OR NOT ENOUGH SYSTEM AVAILABLE MEMORY TO BUFFER THE REQUEST.

""

IO05

ILLEGAL DISC TRACK OR SECTOR SPECIFIED.

""

IO06

A REFERENCE TO A PROTECTED TRACK OR TO UNASSIGNED LG TRACKS.

""

IO07

THE DRIVER HAS REJECTED THE CALL.

" "

I009

LG TRACKS OVERFLOWED.

" "

I010

CLASS GET CALL ISSUED WHILE ONE CALL ALREADY OUTSTANDING.

" "

I011

A TYPE 4 PROGRAM MADE AN UNBUFFERED I/O REQUEST TO A DRIVER THAT DID NOT DO ITS OWN MAPPING.

" "

I012

LOGICAL UNIT NOT DEFINED FOR USE BY
THIS SESSION. THE "SL" COMMAND WILL REPORT ALL LOGICAL UNITS
AVAILABLE TO YOUR SESSION.

" "

I013

LU EITHER LOCKED TO ANOTHER PROGRAM, OR I/O REQUEST
POINTED TO AN EQT LOCKED TO ANOTHER PROGRAM.

" "

I014

AN I/O REQUEST WAS ISSUED WITH THE NO-SUSPEND OPTION.

" "

I014

BUFFER SIZE OF A TYPE 6 PROGRAM IS GREATER THAN WHAT WILL FIT
IN THE USER MAP.

" "

I016

CPU BACKPLANE FAILURE OR I/O EXTENDER TIMING FAILURE.

" "

I020

CANNOT READ WRITE-ONLY SPOOL FILE.
REVISE PROGRAM CALL TO SPOPN OR CHECK "SL" COMMAND PARAMETERS.

""
IO21

CANNOT READ END-OF-FILE.
REVISE PROGRAM AND RE-RUN.

""
IO22

SECOND ATTEMPT TO READ JCL CARD FROM BATCH INPUT FILE BY OTHER THAN FMGR. REVISE PROGRAM AND RE-RUN.

""
IO23

CANNOT WRITE TO READ-ONLY SPOOL FILE.
REVISE PROGRAM CALL TO SPOPN OR CHECK "SL" COMMAND PARAMETERS.

""
IO24

WRITE ATTEMPTED BEYOND END-OF-FILE; USUALLY, SPOOL FILE OVERFLOW. OBTAIN MORE SPOOL ROOM ON DISC (SEE PK COMMAND IN BATCH AND SPOOLING MANUAL) OR DO NOT USE SPOOLING AT THIS TIME.

""
IO25

ATTEMPT TO ACCESS SPOOL LU THAT IS NOT CURRENTLY SET UP. MAY BE CAUSED BY GASP KS COMMAND - IF OTHER REASON, CORRECT OFFENDING PROGRAMS.

""
IO26

I/O REQUEST MADE TO A SPOOL THAT HAS BEEN TERMINATED BY THE GASP KS COMMAND. RESET THE SESSION LOGICAL UNIT WITH THE CS OR SL COMMAND.

""
IOET

END-OF-TAPE CONDITION ON THE SPECIFIED LU. MOUNT A NEW TAPE AND SET THE EQT UP.

""
IONR

SPECIFIED LU NOT READY. MAKE THE DEVICE READY AND SET THE EQT UP.

""
IOTO

SPECIFIED LU TIMED OUT. EXAMINE THE DEVICE, CORRECT THE PROBLEM, AND SET THE EQT UP.

""
IOPE

PARITY ERROR IN THE DATA TRANSMISSION FROM THE SPECIFIED LU. EXAMINE THE DEVICE, CORRECT THE PROBLEM, AND SET THE EQT UP.

""
ILL INT

ILLEGAL INTERRUPT ON THE SPECIFIED CHANNEL.

Appendix B
Logical Source and Load-And-Go Areas

The Logical Source (LS) and Load-and-Go (LG) areas are no longer required by HP supported subsystems. Earlier RTE Compiler and Loader versions used these areas for temporary program storage. Source programs could be stored on the LS tracks where they could be edited if necessary then compiled or assembled. The resulting binary relocatable code could then be moved to the LG area before invoking the loader.

The commands described in this Appendix are not recognized by the Session Monitor. They can only be input from a non-session environment.

The Logical Source (LS) areas can be allocated on the system (LU 2) or auxiliary (LU 3) disc. Each LS disc area is allocated in units of whole tracks. The system maintains a pointer to the last source program moved to an LS area. The user may reset or clear this pointer with the File Manager or System LS command. A program can be moved to an LS area with the File Manager MS command, and a program can be saved from the LS area as a type 4 file (source program) using the File Manager SA command.

The Load-and-Go (LG) area was used by earlier RTE compiler and assembler versions for storing the binary relocatable code of programs and merging files. The number of tracks in this area must have been specifically declared with the LG command before running an assembler or compiler. The relocatable binary program resulting from assembly or compilation may be placed in the LG area. The earlier Loader versions expected a binary relocatable program to be in the LG tracks by an LG command. Any program segments must have also been loaded into the LG area with the main program. If a main program or segment used subroutines not in a resident library, these also must have been loaded into the LG area.

LS And LG Areas

The LG area must have had enough tracks to hold the main program plus any segments or subroutines. The LG area was cleared automatically when the Loader program terminated normally or at the end of a job by the FMGR EOJ command. The LG area could also be cleared by the LG command. A file in the LG area could be saved with the File Manager SA command as a type 5 file. A type 5 file could be moved to the LG area with the File Manager MR command.

Following are the LS and LG area related commands. By inspecting the prompt character, the user can determine whether the command is a File Manager, System, or Editor command.

MS (Move Source File)

The MS command moves a program on a FMGR file to an LS (logical source) area where it can be edited, compiled or assembled.

```
+-------------------------------------------------------------------+
|                                                                   |
| :MS,namr[,program[,IH]]                                           |
|                                                                   |
| Parameters                                                        |
|                                                                   |
| namr      File name  or LU of program to be transferred; (refer  |
|           to the Namr Parameter, Chapter 3).                      |
|                                                                   |
| program   Name of program  to which LS tracks  are assigned; if  |
|           omitted, tracks are assigned to EDIT.                   |
|                                                                   |
| IH        Inhibit  the pointer  to the LS track from being set,  |
|           which requires use of the LS command  to set pointer,  |
|           if omitted, pointer is set to the LS track containing  |
|           namr.                                                   |
+-------------------------------------------------------------------+
```

NOTE

A program file to be moved with MS must not have records longer than 128 words.

When MS is entered, the LU and first track of the LS area are reported on the log device as:

```
FMGR 015
LS LU n TRACK nnn
```

If you use the LS command to set or reset the pointer to this LS track, you will need to know these numbers.

Ordinarily, the LS tracks are assigned to the program EDIT when MS is executed. If, however, you have a reason to assign these tracks to another program, it may be specified as program. Care should be taken that the program specified does not release the LS tracks. FMGR, for instance, should never be specified as program since all tracks assigned to FMGR are released when the MS command executes and also when FMGR is terminated.

LS And LG Areas

If FMGR was scheduled from a program using the RTE Program Schedule EXEC call with wait, the scheduling program can retrieve the LU and track number in the second parameter returned by a call to RMPAR:

```
    DIMENSION PRAM (5)
    CALL RMPAR (PRAM)


     15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0
    +-----------------------------------------------+
    |  |     TRACK NUMBER        |     SECTOR = 0    | <--PRAM(2)
    +-----------------------------------------------+
      ^
     BIT 15:  = 0 if LU2
              = 1 if LU3
```

Logical source tracks are RTE system tracks so the LU can be only 2 or 3. RMPAR returns the absolute track number (from system track 0, not FMP track 0). The maximum source track number is 255 for LUs 2 and 3. FMGR 005 is issued if there are no free tracks between 0 and 255.

The track number and LU of the LS area are important in case the LS pointer is moved and no longer points to the particular area containing your program. This may occur if another user resets the pointer. In this case, use the LS command (SET LOGICAL SOURCE POINTER) to reset the pointer to the area you want.

LS tracks assigned to the EDIT program are not released automatically; they may be released with the RT command (refer to Chapter 3 for a description of the RT command).

EXAMPLES:

1. Create a source file from the terminal and then move it to a logical source area for compilation.

```
:ST,1,SPFILE
enter program statements
CNTL/D <-------------------------------- end file with CNTL/D.

:MS,SPFILE
FMGR 015
 LS LU 2 TRACK 010 <-------- SPFILE is now on system track 10.
```

2. If IH is specified, the editor, compilers or assembler will not be able to access the particular logical source until :LS is used to set the pointer:

```
:MS,SOURC2,,IH
FMGR 015
 LS LU 2 TRACK 012<-------use this LU and track number in LS
                          command
```

LS (Set Logical Source Pointer)

A pointer to a particular logical source track can be set with the LS command; this command will also clear the current pointer. It is identical to the RTE *LS command.

```
+------------------------------------------------------------------+
|                                                                  |
| :LS[,lu,track]                                                   |
|                                                                  |
| Parameters                                                       |
|                                                                  |
| lu        LU of LS area;  must be 2 or 3  to set pointer,  0 to  |
|           clear a current pointer.                               |
|                                                                  |
| track     Track number to which pointer is set.                  |
|                                                                  |
|           If both track and lu are omitted, the current pointer  |
|           is cleared.                                            |
+------------------------------------------------------------------+
```

Unless the pointer is set to a particular LS area, the source program in that area cannot be referenced. Since the pointer is set whenever MS is executed, LS need be used only if the pointer has been reset to a different LS area.

EXAMPLE:

```
    :MS,AFILE,,IH <------------------------------- no pointer set.
    FMGR 015
     LS LU 2 TRACK 013
    :MS,BFILE
    FMGR 015
     LS LU 2 TRACK 014 <----- pointer set to track 14, file BFILE.
      .
      .
      .
    :LS,2,13 <-------------------- pointer set to track 13, AFILE.
```

LG (Assign LG Tracks)

A set of tracks is allocated to the LG area with the LG command; this area must contain binary relocatable programs that are to be loaded with the LOADR.

```
 :LG[,#tracks]

 Parameters

 #tracks     Positive integer indicating the number of tracks to
             reserve on the system or auxiliary disc  for the LG
             area;  if omitted  or  zero  the current tracks are
             returned to the system.
```

Whenever LG is specified, any binary relocatable program in the current LG area is cleared. The LG area is also cleared upon successful completion of LOADR and at the beginning or end of a batch job by the JO and EO commands.

If #tracks is not specified, the LG tracks are returned to the system and no tracks are available for a compiled or assembled program. If #tracks is specified, then the current LG area is cleared and a new area is allocated for the LG tracks.

The FMGR LG command is the same as the RTE LG command and the same error messages may be generated.

EXAMPLE:

```
    :LG,3 <------------------------- allocate 3 tracks to LG area.
    :RU,ASMB,2,99 <----------- assemble program source in LS area;
                               relocatable binary to LG.
    :SA,LG,BFILE <------- save binary relocatable program as file.
    :LG <--------------------------- return LG tracks to system.
```

MR (Move Relocatable Program)

A program saved in relocatable binary form (type 5 file) can be moved to the LG area with the MR command.

```
+-----------------------------------------------------------------+
|                                                                 |
| :MR,namr                                                        |
|                                                                 |
| Parameters                                                      |
|                                                                 |
| namr     File name or LU of file to be transferred; may include |
|          security  and Cartridge Reference Number (CRN)  (refer |
|          to the Namr Parameter section, Chapter 3).             |
|                                                                 |
+-----------------------------------------------------------------+
```

If a file namr is not terminated by an END record, FMGR prints the message FMGR 006 when the end of file is reached. FMGR suspends and you should then load the next relocatable module and enter *GO,FMGR. This situation may occur when more than one relocatable binary module is entered through the paper tape reader. If it occurs within a batch job, FMGR is terminated and you must re-schedule it with the FMGR RU command.

If no LG tracks are allocated when this command is given, the file size is checked and an LG allocation large enough to hold the file is made automatically.

Unlike the MS command, repeated MR commands add each specified file to the same LG area until it is cleared, whereas MS creates a new LS area for each specified file.

EXAMPLES:

1. A relocatable binary program requiring 2 tracks has been punched on paper tape; to move it to the LG area:

```
:LG,2 <------------------------ clear and assign two LG tracks.
:MR,5 <------------- read program from paper tape to LG tracks.
```

2. Move type 5 file XX to LG area with 1 track:

```
:LG,1
:MR,XX
```

3. Merge a program into the LG area:

```
:LG,4                       a program is divided into 3 parts, each
:MR,PROGA                   stored as type 5 file,  are moved to an
:MR,SEGA1                   LG area one following the other.
:MR,SEGA2
:SA,LG,%MAIN                save the merged program into %MAIN.
```

SA (Save Program As File)

To save the contents of a logical source (LS) area or the load-and-go (LG) area as a file, use the SA command. A file is created by SA.

```
+-----------------------------------------------------------------+
|                                                                 |
|      ,LS                                                        |
| :SAVE    ,namr                                                  |
|    --,LG                                                        |
|                                                                 |
| Parameters                                                      |
|                                                                 |
| LS          Source program in LS area is saved.                 |
|                                                                 |
| LG          Relocatable program in LG area is saved.            |
|                                                                 |
| namr        File name  or LU of file in which LS or LG is saved; |
|             file is created and full namr can be used; (refer to |
|             the Namr Parameter section, Chapter 3).             |
|                                                                 |
+-----------------------------------------------------------------+
```

NOTE

The SA command truncates records longer than 128 words; no HP compiler or assembler generates records longer than 128 words.

If there is not enough cartridge space available when namr is a disc file, error message FMGR -033 is issued. Any portion of the file that was saved is purged.

Save LS

For SA, LS namr the file type when omitted defaults to type 4.

The file size defaults to one half the number of blocks on a system disc track. After the final end-of-file is written on the new file, its position is checked. If there are no extents, any remaining unused disc space is returned to the system. If extents were created, the file size is not shortened.

Save LG

For SA,LG,namr, the file type of namr when omitted defaults to type 5. A checksum is provided.

Following each relocatable module (main program, program segment or library subroutine) including the last, a zero-length record is written if namr is a disc file or an EOF if namr is a non-disc device.

The file size, if omitted, is computed as the maximum size possible from the amount of LG area used. Extents are not created and file will be only as long as needed.

EXAMPLES:

```
1.  :LS,2,36 <----------- set LS pointer to system disc, track 36.
    :SA,LS,SFILE::13 <----------- save LS area as type 4 file, on
                                  cartridge 13; default file size.

2.  :SA,LG,4 <-punch relocatable program in LG area to paper tape.

3.  :SA,LG,LFILE:JM:-14 <-------- save relocatable program as type
                                  5 file on LU 14 protected by
                                  security code JM.
```

LG (LG Tracks)

Allocates or releases a group of disc tracks for the LG area. LG tracks may be used as temporary storage for relocatable code in FMGR operations.

```
+----------------------------------------------------------------+
|                                                                |
| *LG,numb                                                       |
|                                                                |
| where:                                                         |
|                                                                |
| numb=0       (zero) releases the allocated LG area.            |
|                                                                |
| numb>0       release currently allocated LG tracks and then    |
|              allocate numb contiguous tracks for an LG area.   |
|                                                                |
+----------------------------------------------------------------+
```

Enough LG tracks for storing relocatable code must be allocated before storing into this area. Insufficient tracks cause the program to abort and one of the following diagnostics to be displayed on the system console:

```
 IO06 -     LG area not defined.
 IO09 -     LG area overflow.
```

An LG request should not be used while anyone is using the LG tracks. Doing so may result in the message:

```
            LGO IN USE
```

being displayed on the system console, and no change in the current number of LG tracks. In most cases, however, the attempt to do so results in an IO06 error being issued.

LS (Source File)

Designates the disc LU and starting track number of source code stored in the track pool.

```
+------------------------------------------------------------------+
|                                                                  |
| *LS,disc lu,trk numb                                             |
|                                                                  |
| where:                                                           |
|                                                                  |
| disc lu   is the LU of the disc containing the source file.     |
|                                                                  |
|           2 or 3 = system or auxiliary disc units.               |
|                                                                  |
|                0 = eliminate  the current source file designa-   |
|                    tion.                                         |
|                                                                  |
| trk numb  is the starting track number (decimal) of the source  |
|           code.  Defaults to 0 if not specified.                 |
|                                                                  |
+------------------------------------------------------------------+
```

LS replaces any previous declarations with the current source code area. Only one area may be declared at a time.

Appendix C
Tables and Directories

Program ID Segment

Each user program has a 36-word ID segment located in memory that contains static and dynamic information defining the properties of the program. The static information is set during generation time or when the program is loaded on-line. The dynamic information is maintained by the operating system Executive.

The number of ID segments contained in a system is established during system generation, and is directly related to the number of programs that can be in main memory at any given time. If all the ID segments are in use, no more programs can be added on-line unless some other existing program is first "offed" (removed from the system) to recover an ID segment.

The format of the ID segment is illustrated in Figure C-1. Each ID segment's address is located in the Keyword Table (see location 01657).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Word	
List Linkage																0	←--XEQT
TEMP 1																1	
TEMP 2																2	
TEMP 3																3	
TEMP 4																4	
TEMP 5																5	
Priority																6	
Primary Entry Point																7*	
Point of Suspension																8	
A-Register																9	
B-Register																10	
EO-Registers																11	
Name 1								Name 2								12*	
Name 3								Name 4								13*	
Name 5								TM	ML	DS	SS	Type				14*	Memory-Resident Programs
NA	NS	NP	W	A	FS	O	LP	R	D			Status				15	
Time List Linkage																16	
RES			T	Multiple												17	
Low Order 16 Bits of Time																18	
High Order Bits of Time																19	
BA	FW	M	AT	RM	RE	PW	RN	Father ID Segment No.								20	
RP	# pgs. (no BP)					MPFI			DE	Partition No. −1						21	
Low Main Address																22*	
High Main Address + 1																23*	
Low Base Pg Addr (Non-MLS Prog) or # Sectors on LU 2 or 3 for MLS Prog																24*	
High Base Page Address + 1																25*	
	Program: Track #															26*	
LU					Swap: Track #											27	
ID Extension No.						EMA Size										28	
High Address + 1 of Largest Segment or Node																29	
Timeslice word																30	
SEQCNT				SH	DC	CP	DS	Session ID								31	Memory-Residents
SCB Pointer																32	
MS	# pages disc resident					# pages memory-resident										33	
MP	# pgs dynamic buffer area					E	DB	# of swap tracks								34	
Start sector address of program								LU # of prog								35	

8100-2A

Figure C-1. ID Segment Format

where:

* = words used in short ID segments for program segments.

TM = temporary load (copy of ID segment is not on the disc).

ML = memory lock (program may not be swapped).

TS = the program is transportable.

SS = short segment (indicates a nine-word segment).

Type = specified program type (1-5).

NA = no abort (instead, pass abort errors to program).

NS = the no-suspend bit was set on the I/O request.

NP = no parameters allowed on reschedule.

W = wait bit (waiting for program whose ID segment address is in word 2).

A = abort on next list entry for this program.

FS = file system bit (this program has files open).

O = operator suspend on next schedule attempt.

LP = load in progress; program is being dispatched from disc.

R = resource save (save resources when setting dormant).

D = dormant bit (set dormant on next schedule attempt).

Status = current program status

T = time list entry bit (program is in the time list).

BA = batch (program is running under batch).

FW = father is waiting (father scheduled with wait).

M = Multi-Terminal Monitor bit.

AT = attention bit (operator has requested attention).

RM = reentrant memory must be moved before dispatching program.

RE = reentrant routine now has control.

PW = program wait (some other program wants to schedule this one).

RN = Resource Number either owned or locked by this program.

RP = reserved partition (only for programs that request it).

MPFI = memory protect fence index.

DE = defer EXEC 6 (terminate program) request.

LU = 0 if LU 2, 1 if LU 3

TIMESLICE WORD (30):

The timeslice word defines the timeslicing status of a program. This word is defined as follows:

0 = This program has just been rescheduled or is not timesliced.

>0 = This program has used a full timeslice or program is not scheduled.

<0 = This program was running (under timeslice control) and was "bumped" from execution by a higher priority program. This word represents the remaining timeslice for this program.

OPEN FLAG WORD(31):

SEQCNT = sequence counter. Each time a program is aborted or terminates (unless saving resources) the counter is incremented. The counter value is used to build FMP open Flags.

SH = the program uses shareable EMA.

DC = do not copy flag. Set by the generator (if 128 is added to program type) or the loader (using Don't Copy op-code).

DS = used by the distributed systems software.

CP = copy flag. Indicates that the program is a copy.

Session ID = System LU of terminal that program was loaded from.

SESSION WORD(32):

The session word identifies the owner of a program.

A negative value represents the logical unit number of the terminal from which the program was invoked (not under session).

A positive value represents the address of the SST length word or the session control block for the session currently using this program (under session).

Programs scheduled by interrupt will have a zero in this word.

MS	= the program is a multilevel segmentation program.
MP	= the program is using modified maps for I/O.
E	= the program needs an EXEC 4 allocation request.
DB	= debug bit.

ID Segment Extensions

Each EMA program requires a 3-word ID segment extension in addition to its 36-word ID segment. The number of ID extensions contained in the system is also set at generation time, and if all are in use, no more EMA programs can be added on-line. The format of the ID segment is illustrated in Figure C-2.

<table>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th></th></tr>
<tr><td colspan="16">Not Used</td><td>Word 0</td></tr>
<tr><td colspan="5">MSEG Start Page (logic.)</td><td>DE</td><td colspan="10">(Physical) EMA Start Page</td><td>Word 1</td></tr>
<tr><td>COM</td><td></td><td>SW</td><td></td><td></td><td></td><td colspan="10">Index # into $EMTB</td><td>Word 2</td></tr>
<tr><td colspan="16">Maximum Page Number Allowed in VMA.</td><td>Word 3</td></tr>
<tr><td colspan="16">Reserved</td><td>Word 4</td></tr>
</table>

8100-14

where:

SW = 0 PTE table does not contain valid data.

= 1 PTE table is still intact (working set has not been moved).

DE = 0 if the EMA size was specified by the user.

= 1 if the EMA size is allowed to default to the maximum size available to the system.

COM = 1 the program is using shareable EMA.

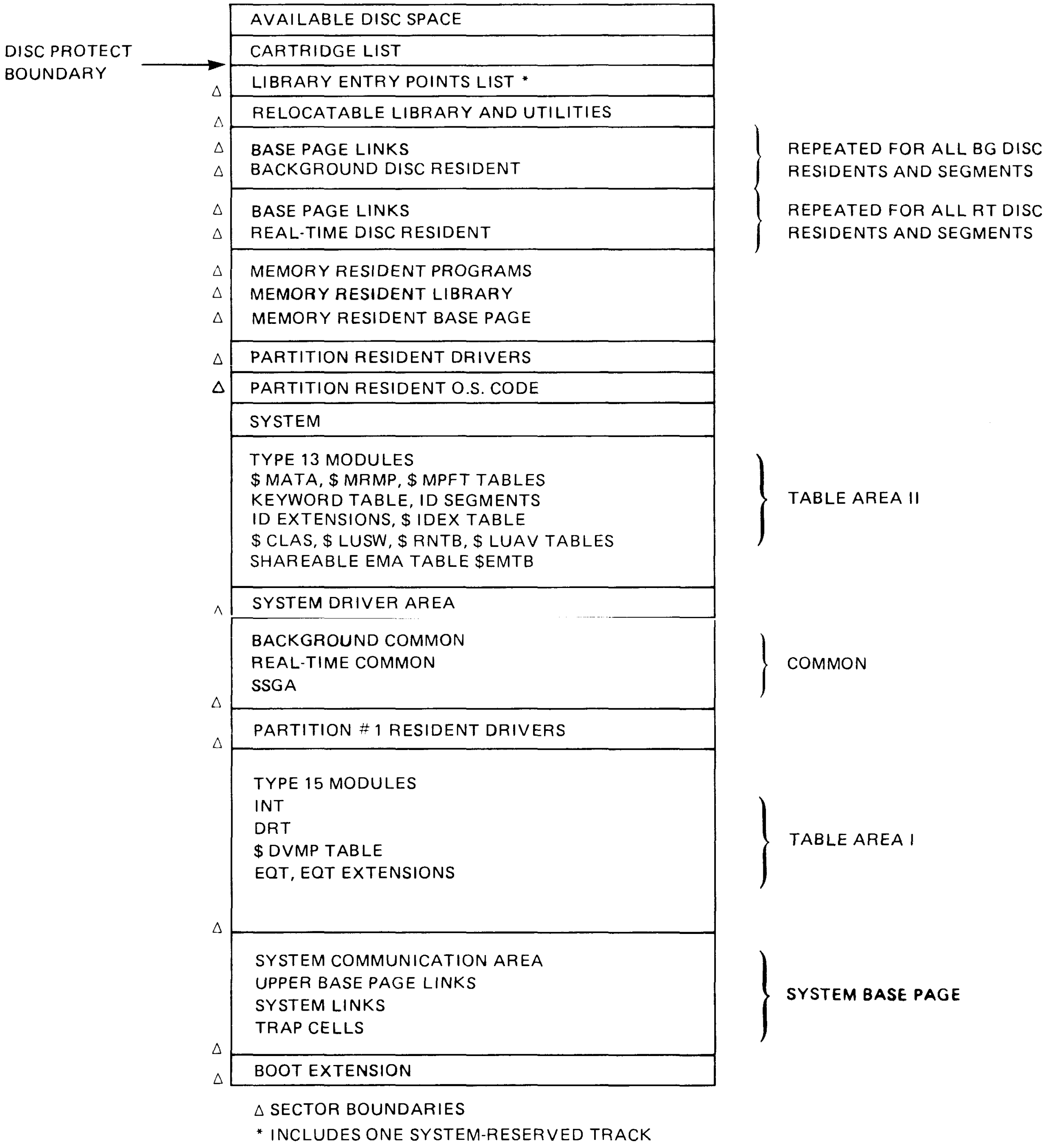
Figure C-2. ID Segment Extension

Short ID Segments

Short ID segments requiring nine words are used only for program segments. A short ID segment is required for each segment of a segmented program. If no empty short ID segments are available during an on-line load, a standard 33-word ID segment will be used. The information contained in a short ID segment is illustrated in Figure C-1.

RTE-6/VM System Disc Layout

Figure C-3 illustrates how disc space is allocated when a RTE-IVB system is generated.



DISC PROTECT BOUNDARY

AVAILABLE DISC SPACE

CARTRIDGE LIST

Δ LIBRARY ENTRY POINTS LIST *

Δ RELOCATABLE LIBRARY AND UTILITIES

Δ BASE PAGE LINKS
Δ BACKGROUND DISC RESIDENT

REPEATED FOR ALL BG DISC RESIDENTS AND SEGMENTS

Δ BASE PAGE LINKS
Δ REAL-TIME DISC RESIDENT

REPEATED FOR ALL RT DISC RESIDENTS AND SEGMENTS

Δ MEMORY RESIDENT PROGRAMS
Δ MEMORY RESIDENT LIBRARY
Δ MEMORY RESIDENT BASE PAGE

Δ PARTITION RESIDENT DRIVERS

Δ PARTITION RESIDENT O.S. CODE

SYSTEM

TYPE 13 MODULES
$ MATA, $ MRMP, $ MPFT TABLES
KEYWORD TABLE, ID SEGMENTS
ID EXTENSIONS, $ IDEX TABLE
$ CLAS, $ LUSW, $ RNTB, $ LUAV TABLES
SHAREABLE EMA TABLE $EMTB

TABLE AREA II

Λ SYSTEM DRIVER AREA

BACKGROUND COMMON
REAL-TIME COMMON
SSGA

COMMON

Δ PARTITION #1 RESIDENT DRIVERS

Δ TYPE 15 MODULES
INT
DRT
$ DVMP TABLE
EQT, EQT EXTENSIONS

TABLE AREA I

Δ SYSTEM COMMUNICATION AREA
UPPER BASE PAGE LINKS
SYSTEM LINKS
TRAP CELLS

SYSTEM BASE PAGE

Δ BOOT EXTENSION Δ

Δ SECTOR BOUNDARIES

* INCLUDES ONE SYSTEM-RESERVED TRACK

8100-8

Figure C-3. RTE-6/VM System Disc Layout

Data Control Block Format

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WORD 0*	SECTOR OFFSET			SECTOR # OF FILE DIRECTORY							LU # OF FILE DIRECTORY OR OF FILE IF ON DISC					
1*	TRACK# OF FILE DIRECTORY															
2	FILE TYPE (MAY BE OVERRIDDEN AT OPEN, UNLESS TYPE 0)															
3	TRACK ADDRESS OF FILE (TYPE ≥ 1) OR LU# OF FILE (TYPE = 0)															
4	SECTOR ADDRESS OF FILE (TYPE >= 1) OR EXEC FUNCTION CODE (TYPE = 0)															
5	FILE SIZE IN -CHUNKS +SECTORS (TYPE ≥ 1) OR SPACING CODE (TYPE = 0)															
6	RECORD LENGTH (TYPE =2) OR END-OF-FILE CODE (TYPE =0)															
7	WA	PF	NUMBER OF BLOCKS IN DCB BUFFER							FM	EX	RA	OM	IB	EF	WR
8	NUMBER OF SECTORS PER TRACK (TYPE ≥ 1)															
9	OPEN/CLOSE INDICATOR															
10	TRACK # OF CURRENT FILE POSITION (TYPE > = 1) OR THE VALUE OF THE A-REGISTER AFTER THE LAST EXEC CALL (TYPE = 0)															
11	TRACK # OF CURRENT FILE POSITION (TYPE > = 1) OR THE VALUE OF THE B-REGISTER AFTER THE LAST EXEC CALL (TYPE = 0)															
12	LOCATION OF NEXT WORD IN FILE (TYPE ≥ 1)															
13	RECORD # OF CURRENT FILE															
14	POSITION (DOUBLE WORD INTEGER)															
15	EXTENT NUMBER (TYPE ≥ 3)															
16	DCB BUFFER AREA (128 - N)															

FILE DIRECTORY ADDRESS (words 0–1)

CURRENT POSITION IN FILE (words 10–12)

16-WORD CARTRIDGE (words 0–15)

*FILE DIRECTORY ADDRESS

8100-4A

Legend for Data Control Block

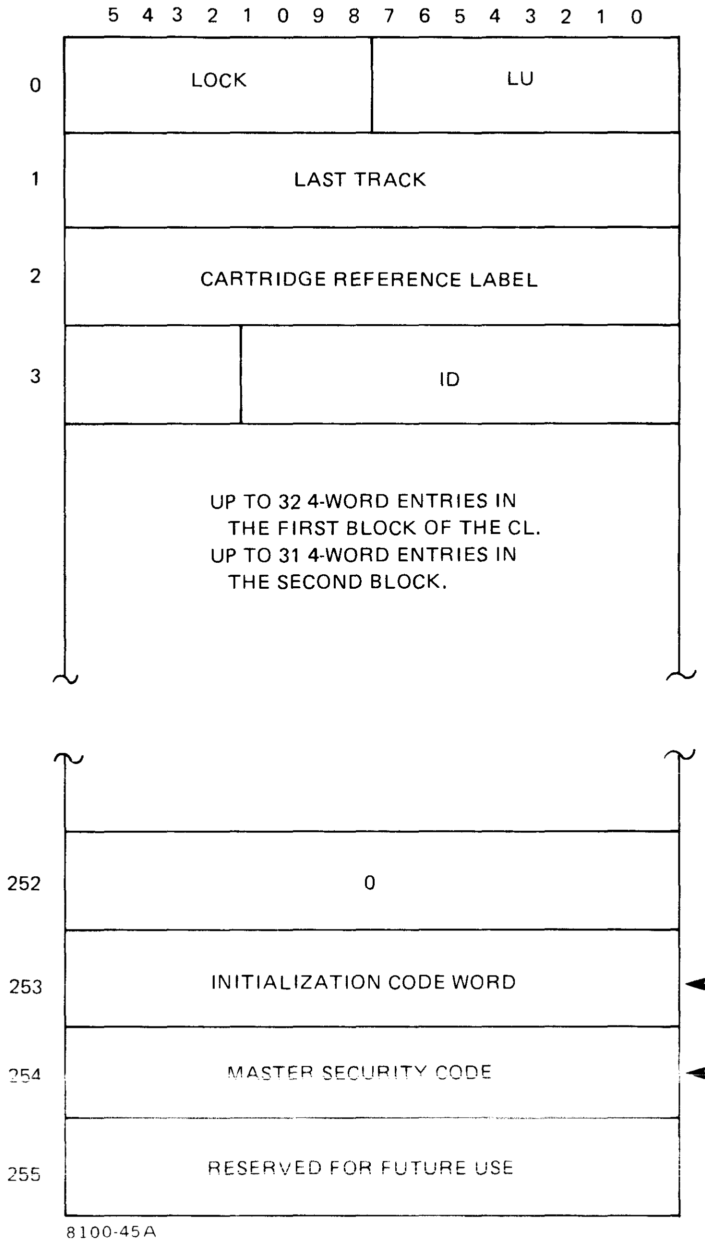
Word		Contents
0	File Directory Address:	bits 6-12 = Physical sector number of the file directory. bits 13-15 = Entry offset from the beginning of the block (origin 0).
5	Spacing Code: (type 0 file)	bit 15 = 1 - backspace legal. bit 0 = 1 - forward space legal.
6	End-of-File Code: (type 0 file)	01 lu = EOF on Magnetic Tape. 10 lu = EOF on Paper Tape. 11 lu = EOF on Line Printer.
7	Status Information	
	(WA) Write Allowed:	bit 15 = 1 - write to file allowed. = 0 - write to file not allowed.
	(PF) Partially Full:	bit 14 = 1 - DCB is only partially full. = 0 - DCB is full.
	DCB Buffer:	bits 13-7 = Number of blocks in the DCB buffer.
	(FM) File Modify:	bit 6 = 1 - File has been modified. = 0 - File has not been modifed.
	(EX) Extendable:	bit 5 = 1 - File is not extendable. = 0 - File is extendable.
	(RA) Read Allowed:	bit 4 = 1 - Read from file allowed. = 0 - Read from file not allowed.
	(OM) Open Mode:	bit 3 = 1 - update open 0 - standard open
	(IB) In Buffer Flag:	bit 2 = 1 - data in DCB buffer = 0 - data not in DCB buffer
	(EF) EOF Read Flag:	bit 0 = 1 - EOF has been read = 0 - EOF has not been read

(WR) To Be Written: bit 0 = 1 - data in DCB buffer to be written
= 0 - data in DCB buffer not to be written

9 Open/Close Indicator: if open, contains ID segment location of program performing open. If closed, set to zero.

Cartridge Directory Format

The cartridge directory is located in the system area on LU 2. Its length is two blocks.



Word	5 4 3 2 1 0 9 8	7 6 5 4 3 2 1 0
0	LOCK	LU
1	LAST TRACK	
2	CARTRIDGE REFERENCE LABEL	
3		ID
	UP TO 32 4-WORD ENTRIES IN THE FIRST BLOCK OF THE CL. UP TO 31 4-WORD ENTRIES IN THE SECOND BLOCK.	
252	0	
253	INITIALIZATION CODE WORD	
254	MASTER SECURITY CODE	
255	RESERVED FOR FUTURE USE	

8100-45A

LOCKER DISCS ARE AVAILABLE ONLY TO THE LOCKER.

LOCK = 0 IF NOT LOCKED; ELSE IS KEYWORD TABLE OFFSET OF ID SEGMENT ADDRESS OF LOCKING PROGRAM

IF THE CARTRIDGE REFERENCE LABEL IS EQUAL TO ZERO, THIS IS A HIERARCHICAL FILE VOLUME.

ID IDENTIFIES TO WHOM THE CARTRIDGE IS MOUNTED.

ID = 0000 → NON-SESSION
ID = 7777 → SYSTEM CARTRIDGE
0<ID<7777→ SESSION MONITOR GROUP OR PRIVATE CARTRIDGE

NOTE: WORDS 124, 125, 126, AND 127 ARE UNIQUE ONLY IN THE SECOND BLOCK OF THE CL. THE FIRST BLOCK WILL HOLD 32 ENTRIES IN WORDS 0 THROUGH 127.

SUM OF CONTENTS OF BASE PAGE WORDS 1650 THRU 1657 AND 1742 THRU 1747 AND 1755 THRU 1764.

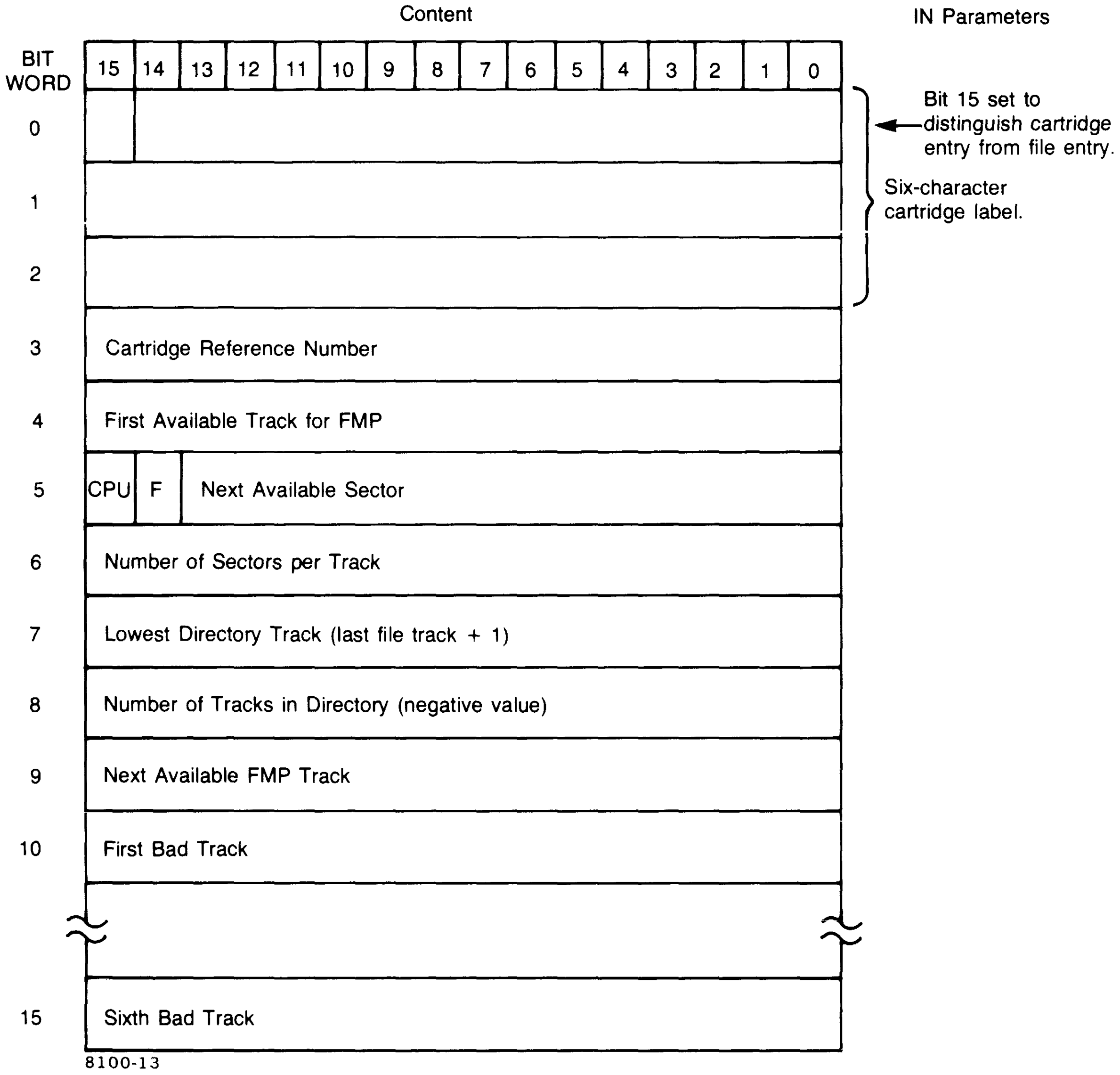
SET WHEN SYSTEM CARTRIDGE IS INITIALIZED.

File Directory

The first entry in each File Directory is the specification entry for the cartridge itself. The directory starts on the last FMP track of each cartridge in sector zero on all discs. The directory blocks are written using sector skipping. The directory sector address can be obtained from the block address by the following formula:

```
sector address = (block *14) modulo S/T
```

where S/T is the number of sectors per track. Directory blocks are 128 words long. Each Directory entry is 16 words long.



BIT / WORD	Content (bits 15–0)	IN Parameters
0	Bit 15 | Bits 14–0	Bit 15 set to distinguish cartridge entry from file entry. Six-character cartridge label.
1		
2		
3	Cartridge Reference Number	
4	First Available Track for FMP	
5	CPU | F | Next Available Sector	
6	Number of Sectors per Track	
7	Lowest Directory Track (last file track + 1)	
8	Number of Tracks in Directory (negative value)	
9	Next Available FMP Track	
10	First Bad Track	
15	Sixth Bad Track	

8100-13

Disc File Directory

```
   bit   15          8 7          0        CR parameters:
           |         | | |        |
       +--------------------------+
word  0|                          |\
       |   6-character file name  | \
      1|                          |  <--- file name
       |                          | /
      2|                          |/
       |--------------------------|
      3| file type (1 thru 32767) |
       |--------------------------|
      4| starting track           |
       |--------------------------|
      5| extent #     |  starting |
       |                  sector  |
       |--------------------------|
      6| file size in + sector or |
       |              - chunks    |
       |--------------------------|
      7|record length(type 2 only)|
       |--------------------------|
      8| security code            |
       |--------------------------|
      9|                          |\
       |                          | \
     10|                          |  \
       |                          |   \
     11|                          |    \
       |                          |     \
     12|                          |      Open Flags
       |                          |      15 = 1 for exclusive open
     13|                          |     /14-12 = reserved
       |                          |    /  11-8 = sequence counter
     14|                          |   /    7-0 = keyword offset of
       |                          |  /           opening program's
     15|                          |/             ID segment
       +--------------------------+

word 0 = 0 if the last entry in directory; = -1 if file is purged
```

Type 0 File Directory Entry

The entries for non-disc (type 0) files differ from those for disc files in words 3 through 7:

```
    bit   15|                     | 0       CR parameters:
          +-----------------------------+
  word 3  |   0 (file type default)     |
          |-----------------------------|
       4  |     logical unit number     |
          |-----------------------------|
       5  |  end of file subfunction    |<---EO,LE,PA or control
          |-----------------------------|
       6  |        spacing code         |<----------BS,FS, or BO
          |-----------------------------|
       7  |     input-output code       |<----------RE,WR, or BO
          +-----------------------------+
```

Words 5-7 are octal codes:

```
  end-of-file subfunction = 01LU for MT(EO)
                            10LU for paper tape (LE)
                            11LU for line printer (PA)
                            or subfunction code

  spacing code  =  bit 15 = 1 backspace legal (BS)
                   bit  0 = 1 forward space legal (FS)

  input/output code  = bit 15 = 1 input legal (RE)
                       bit  0 = 1 output legal (WR)
```

Session Control Block (SCB)

A Session Control Block (SCB) is established for each user who has successfully "logged-on" to the system. The SCB contains the information necessary to identify the user to the system and describe his capabilities in terms of command processing and I/O addressing space.

The format of the SCB is shown in Figure C-4.

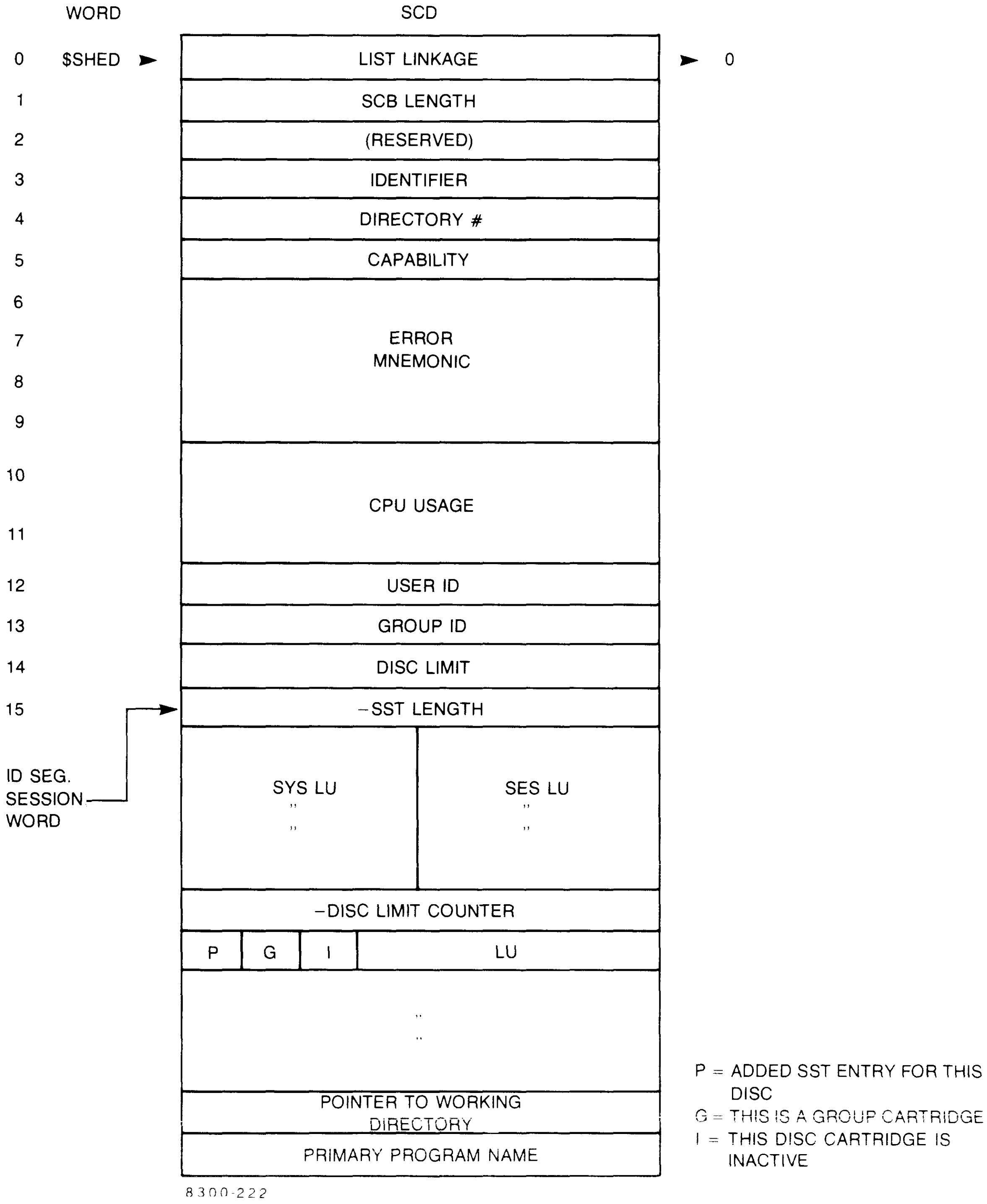


Figure C-4. Session Control Block (SCB)

Session Switch Table (SST) and Configuration Table

When operating in the session environment every I/O request is routed to the appropriate I/O device via the Session Switch Table (SST). Each SST entry describes a session LU, which the user addresses, and associated system LU where the I/O request will actually be directed. The SST describes the session user's I/O addressing capabilities be defining the system LUs the user has access to and the associated session LUs by which the user accesses them.

When the user makes an I/O request the SST is searched for the specified session LU. If the requested LU is found, it is switched to the associated system LU as specified in the SST entry and the I/O request is processed. If the requested LU is not found, an error is returned (IO12-LU not defined for this session).

The Session Switch Table is maintained in memory as part of the Session Control Block (SCB). The format of the SST is shown in Figure C-5.

System LUs can be integer numbers between 1 and 255. Session LUs can be integer numbers between 1 and 63. Session LUs are assigned:

* at log-on, via user and group account file entries, or

* on-line using SL command (see Chapter 3), or

* at log-on, via Configuration Table entries.

The Configuration Table describes the default logical units to be used for specific device logical units. Each station (terminal) logical unit defined in the Configuration Table has associated with it a set of device logical units which are assigned default logical units to be used when a user logs on at this station (terminal). The default logical unit associated with the station itself is always 1.

At log-on, these default values are written from the Configuration Table in the account file into the user's Session Control Block (SCB), unless overridden by entries in this particular user's SST. The format of the Configuration Table is shown in Figure C-6.

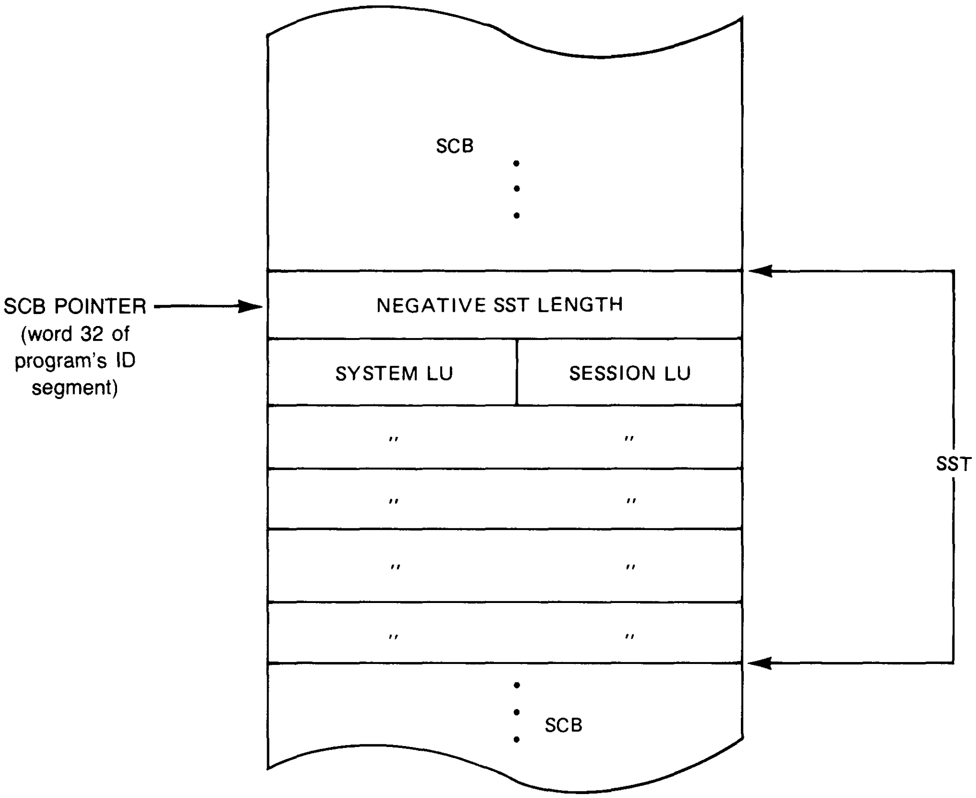


Figure C-5. Session Switch Table (SST) Format

LENGTH	
STATION LU	1
SYSTEM LU	DEFAULT LU
SYSTEM LU	DEFAULT LU
LENGTH	
STATION LU	1
SYSTEM LU	DEFAULT LU
SYSTEM LU	DEFAULT LU
SYSTEM LU	DEFAULT LU
⋮	
0	

8100-5

Figure C-6. Configuration Table

Appendix D
Scheduling FMGR Interactively

To request FMGR from the system console or an auxiliary terminal, the system RU command can be used (refer to Chapter 4 for a description of the RU command). FMGR can also be scheduled programmatically through an EXEC call (refer to the RTE-6/VM Programmer's Reference Manual).

```
+---------------------------------------------------------------+
|                                                               |
|*RU,FMGR [,lu [,list [,severity [,log]]]] <---commands entered |
|                                                  from device  |
|           file                                                |
|*RU,FMGR,descriptor [,list [,severity [,log ]]]  <---commands  |
|                                                     entered   |
|                                                     from file |
|---------------------------------------------------------------|
|                                                               |
| lu         LU of input device  for FMGR commands.  If omitted,|
|            LU 1 is assumed.   In a multi-terminal environment,|
|            the default input  is the LU  of device  where FMGR|
|            scheduled.                                         |
|                                                               |
| log        LU of the log device used to log and to correct any|
|            diagnosed errors; must  be  an  interactive device.|
|            If omitted,  the input device is assumed  unless it|
|            is non-interactive, in which case, LU 1 is assumed.|
|                                                               |
| file       File name of file containing command input to FMGR.|
| descriptor Refer  to  the  file  name   discussion  below  for|
|            restrictions.                                      |
|                                                               |
| list       LU of device used to list results of FMGR commands;|
|            if omitted, LU 1 is assumed.                       |
|                                                               |
+---------------------------------------------------------------+
```

RUNNING PROGRAM FMGR . . . cont'd

severity Severity code defines action in case of error messages;

0 Display error codes and echo commands on log device (default).

1 Display error codes on log device, inhibit command echo.

2 Error code displayed only if error requires transfer of control to log device for correction, in this case, active job terminates; no command echo.

3 Same as 2 except active job not terminated when an error causes transfer to log device. You can transfer back to the job. No command echo.

4 If a FMGR command error is encountered, job continues automatically; no command echo, no transfer to log device, and no job abort occurs.

*** NOTE ***

Once FMGR is running, "log" can be changed with the File Manager LO command, "list" with the LL command, and "severity" with the SV command (refer to Chapter 3 for a description of these commands).

EXAMPLES:

```
1. *RU,FMGR <--------FMGR is scheduled with default values for all
                     parameters: input from LU 1, errors logged on
                     LU 1, list output to LU 1.

2. *RU,FMGR,5,,,1 <---Input from right cartridge tape unit (LU 5),
                      log errors on LU 1, list to LU 1, severity
                      code inhibits command echo on LU 1.

3. *RU,FMGR,FILEA1,,1 <----Input from FILEA1, severity inhibits
                           command echo on console, default list
                           is LU 1.
```

COMMENTS:

The device through which commands are sent to the FMGR program can be any device to which a LU has been assigned and that can be used for input. Normally, when input is from disc, a file name is used to specify the input device.

If a file name is specified and begins with NO, these characters are assumed by RTE to be the scheduling parameter NOW. Specifying a second parameter as any two ASCII characters, directs FMGR to derive the file name from the first parameter. The second parameter is ignored by FMGR in this case. For example:

```
*RU,FMGR,NOBLE,XX [,list [,severity [,log ]]]
```

Note that XX can be any two ASCII characters. If the file name starts with NO, then XX must be specified.

NOTE: For information on the method of scheduling FMGR programmatically via an EXEC 23 or EXEC 24, refer to the RTE-6/VM Programmer's Reference Manual.

INDEX

READER COMMENT SHEET

HP 1000 Computers

RTE-6/VM Terminal User's Manual

92084-90004 January, 1986

We welcome your evaluation of this manual. Your comments and suggestions help us to improve our publications. Please explain your answers under Comments, below, and use additional pages if necessary.

Is this manual technically accurate? □ Yes □ No

Are the concepts and wording easy to understand? □ Yes □ No

Is the format of this manual convenient in size, arrangement, and readability? □ Yes □ No

Comments:

This form requires no postage stamp if mailed in the U.S. For locations outside the U.S., your local HP representative will ensure that your comments are forwarded.

FROM: **Date** ____________

Name ____

Company ____

Address ____

____

____

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 0141 CUPERTINO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Publications Manager
Hewlett-Packard Company
Data Systems Division
11000 Wolfe Road
Cupertino, CA 95014-9974

FOLD FOLD

Part No. 92084-90004
Printed in U.S.A. October 1984
E1084

HEWLETT
PACKARD