pdp11

RSX-11D User's Guide

Order No. DEC-11-OXDUA-B-D

digital

RSX-11D

User's Guide

Order No. DEC-11-OXDUA-B-D

RSX-11D Version 6B (Version 6.1)

digital equipment corporation · maynard. massachusetts

First Printing, June, 1975
Revised, November, 1975

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by DIGITAL.

The postage prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DIGITAL
DEC
PDP
DECUS
UNIBUS
COMPUTER LABS
COMTEX
DDT
DECCOMM
DECsystem-10
DECtape
DIBOL
EDUSYSTEM
FLIP CHIP
FOCAL
INDAC
LAB-8
MASSBUS
OMNIBUS
OS/8
PHA
RSTS
RSX
TYPESET-8
TYPESET-10
TYPESET-11

CONTENTS

CONTENTS (Cont.)

CONTENTS (Cont.)

CONTENTS (Cont.)

CONTENTS (Cont.)

FIGURES

CONTENTS (Cont.)

TABLES

PREFACE

0.1 MANUAL OBJECTIVE AND READER ASSUMPTIONS

The objective of this manual is to provide the user with an introduction to the basic concepts of RSX-11D and to describe the use of the terminal interface with the system.

The manual is intended for use by two types of users: general and privileged. Functions available to the general user are restricted to those required for program development and checkout. Functions available to the privileged user provide the means for system control and modification.

For the general user, no other manuals in the RSX-11D documentation set are prerequisite to the use of this manual. The privileged user, however, may wish to become familiar with the RSX-11D System Manager's Guide.

0.2 STRUCTURE OF THE MANUAL

The manual is divided into three major chapter groupings. Chapters 1 through 4 introduce the user to the system. Chapters 5 and 6 detail actual system commands. Appendices A through N provide summary information.

Chapter 1 through 3 provide an introduction to RSX-11D, its components, file structure, and conventions.

Chapter 4 describes gaining access to the system, logging onto the system, and using the system.

Chapter 5 describes the system commands that are available to all users of the system.

Chapter 6 describes the system commands that can only be used by privileged users.

Chapter 7 describes the use of magnetic tape volumes within the system.

Chapter 8 describes the use and operation of the system node pool program.

Appendixes A through N contain command summaries and error messages for frequently-used RSX-11D components.

The following conventions are used to describe examples and command string formats:

1. ↵ indicates a RETURN. All command lines are normally terminated by pressing RETURN unless otherwise noted in the text.

2. ($) indicates ALTmode or ESCape.

3. In all cases, except where [UIC] or [UFD] is specified, brackets signify optional parameters.

0.3 ASSOCIATED DOCUMENTS

A synopsis of each manual in the RSX-11D documentation set is provided in the *RSX-11D Documentation Directory*, Order Number DEC-11-OXUGA-C-D. The directory also defines the readership of each manual in the set.

CHAPTER 1

INTRODUCTION

RSX-11D is a resource-sharing executive designed to run on the PDP-11/40, PDP-11/45, PDP-11/50, and PDP-11/70. It provides an ideal environment for both real-time data processing and multiuser applications. It includes the following features:

- Memory protection on a task-by-task* basis
- Multiuser approach to both system-provided tasks and user-developed tasks
- User-defined memory partitions
- Modular approach to program development
- An extensive set of monitor console routines (MCR) for task control from any terminal
- MCR routines for system control
- A task overlay system to reduce the amount of memory required for tasks
- Priority and time-slice techniques for scheduling task execution
- Capability to assign priorities (1 through 250) to tasks
- Checkpointing (roll-out/roll-in) based on task priority
- Shared disk-resident and memory-resident libraries containing often-used routines
- Batch capability to allow execution of job streams read from a peripheral device
- Macro call capability for file and I/O operations
- Device independence for I/O flexibility

*A task is the basic executable unit in RSX-11D.

- A file control system for processing RSX-11D standard (Files-11) disk and DECtape volumes, as well as ANSI standard multivolume magnetic tape
- Task approach to device handlers to allow incorporation of user-written handler tasks
- File and volume protection
- System access protection
- Capability to define task attributes during task building, rather than during assembly/compilation, for greater flexibility
- Accounting to provide statistics of task use of system resources
- Spooling of print files
- Support of Massbus devices
- Support of analog/digital devices
- Automatic logging of hardware errors
- On-line diagnostics capability
- Power recovery support
- System-performed memory parity checking
- System generation capability, on-line or off-line

1.1 RSX-11D COMPONENTS

The RSX-11D system consists of the following software components:

- An Executive to control the internal operation of the system
- A set of monitor console routines (MCR) that enable the user to communicate with the system
- An additional set of MCR commands that allow centralized control of the system
- A FORTRAN compiler and object time system (OTS) and a MACRO assembler
- A task builder to create loadable task images from relocatable object modules
- Utility programs
- File control services (FCS)
- Device handler tasks to service I/O requests
- On-line debugging aid (ODT) and trace capability to assist in program development

- Batch stream processor
- System generation package
- Task accounting package
- Hardware error logging package
- System test package

1.2 RSX-11D EXECUTIVE

The main function of the RSX-11D Executive is system control (e.g., scheduling tasks, controlling task interaction, checkpointing, placing tasks in partitioned memory, and interfacing with the user via MCR). The Executive also provides the user with a set of system directive commands which can be included in MACRO and FORTRAN programs to request Executive services.

System directive commands provide the following services for executing tasks:

- Starting, scheduling, canceling, terminating, and resuming other tasks
- Fixing a task in memory or releasing it
- Enabling a task to run or disabling it
- Enabling or disabling checkpointing for a task
- Obtaining information about the status of tasks
- Setting event flags, testing them, and performing other capabilities related to event flags
- Handling software and hardware interrupts (RSX-11D traps)
- Handling input/output

The Executive is described in detail in the RSX-11D Executive Reference Manual.

1.3 MONITOR CONSOLE ROUTINES (MCR)

The monitor console routines (MCR) provide an interface between the user and the system. MCR provides the user with complete task control features. The user issues MCR commands to request system programs (e.g., FORTRAN for task development) and to schedule and execute tasks.

From MCR, the user can obtain the following information about the system:

- The time of day and date
- A list of active tasks

- A list of partitions and common areas
- A list of the physical devices for which the system was generated
- A list of the logical units and their associated physical devices for a task

In addition, the user can log information about the system.

The MCR commands for the above functions are described in Chapter 5.

MCR also provides the user with the capability to initialize RSX-11D file-structured (Files-11) volumes, to mount and dismount Files-11 volumes, to establish user file directories (UFDs), and to establish passwords. (Files-11 and UFDs are discussed in Section 3.2). These and other MCR commands used for centralized control of the system, including the capability to enter time and date and to control spooler operation, are described in Chapter 6.

1.4 FORTRAN LANGUAGE AND OBJECT TIME SYSTEM

RSX-11D FORTRAN is an ANSI standard FORTRAN IV language accompanied by an object time system (OTS). FORTRAN programs can be compiled into object code, or they can be converted into assembly-language source statements that, in turn, can be assembled with other assembly-language modules.

The OTS is an object library of commonly used FORTRAN routines that can be called by FORTRAN programs. The user selects the OTS library that is appropriate for his applications.

1.5 MACRO ASSEMBLER

The RSX-11D MACRO assembler allows the user to take full advantage of the PDP-11 programming language, system directives, and the RSX-11D file control services (FCS). (See section 1.8.) During assembly, MACRO detects errors and produces output indicating the type of errors found, thus simplifying the process of locating program errors.

1.6 TASK BUILDER

The task builder creates loadable core images from assembled or compiled tasks. It links relocatable object files and resolves any references to global symbols, common areas, and shared libraries. The task builder also provides an overlay descriptor language to construct task overlays. The overlay descriptor language simplifies the process of dividing a task into overlaid segments and specifying loading methods.

The task builder provides switches that specify certain task attributes. Because switch information is specified during task building, such information can be changed without reassembling or recompiling the task. The switches indicate the following:

- Whether the task can be checkpointed. Normally, tasks that are not time critical when a higher priority task is requested are regarded as checkpointable
- Whether the task can be aborted by the operator or by an Executive directive
- Whether the task is multiuser
- Whether the task is privileged
- Whether the task uses the floating point processor (not available on PDP-11/40)
- Whether the task is subject to accounting
- Whether the task may be fixed in memory and other factors that control the execution of the task and define its attributes

The task builder stores the task on disk in a form that can be dynamically loaded anywhere in the system's memory; the task is not restricted to operating in a fixed memory area.

1.7 UTILITY PROGRAMS

The RSX-11D utility programs described in the following sections permit file operations such as copying, editing, transferring, verifying, compressing, dumping, modifying, and creating libraries.

1.7.1 Peripheral Interchange Program (PIP)

PIP is used to copy files from one device to another, for example, from disk to printer, to rename files, to list files, and to delete files.

1.7.2 Editor (EDI)

EDI is used to enter source programs into the system and to modify them as needed. A large set of easy-to-use commands makes EDI an effective program development tool.

1.7.3 File Transfer Program (FLX)

FLX performs the following file conversions:

- From DOS-11 to Files-11 format
- From Files-11 to DOS-11 format

- From RT-11 to Files-11 format
- From Files-11 to RT-11 format
- From DOS-11 to DOS-11 format
- From RT-11 to RT-11 format
- From Files-11 to Files-11 format

FLX also allows the user to:

1. List directories of cassettes, RT-11 volumes, or DOS-11 volumes
2. Delete files from DOS-11 and RT-11 volumes
3. Initialize cassettes, RT-11 volumes, or DOS-11 volumes

1.7.4 Verification (VFY)

VFY checks the consistency and accuracy of system files on a Files-11 device (for example, disk). It also prints the number of available blocks in a volume, locates files that could not otherwise be accessed, and lists the names of files for the volume's index file.

1.7.5 Source Language Input Program (SLP)

SLP is a batch-oriented editing program that is used to create and maintain source language files on disk.

1.7.6 File Compression (SQZ)

SQZ removes unnecessary spaces and tabs from MACRO source program files. Additionally, the SQZ program removes comment lines. The resultant file requires less storage space.

1.7.7 Dump Utility (DMP)

DMP enables the user to obtain a printout of files, in either ASCII or octal format.

1.7.8 Librarian (LBR)

LBR provides the user with the capability to create and maintain disk-resident libraries of object and source modules.

1.7.9 ZAP Utility Program (ZAP)

The ZAP utility program provides a facility for examining and modifying task image files and data files. With ZAP, permanent patches can be made to task image or data files without having to recreate the file.

1.8 FILE CONTROL SERVICES

File control services (FCS) provide the user with record-oriented and block-oriented file I/O and additional functions required for file control, such as open, close, and delete operations. At the FCS level, the user issues macro calls to specify desired file control operations.

FCS supports both sequential and direct access files. Sequential access is device independent and can be used for both record-oriented and Files-11 structured devices. Direct access is used for Files-11 structured devices only. Data records can be either fixed length or variable length.

When record I/O is used, a program can gain access to individual records in either of two ways:

1. By moving the data to a user-defined buffer (move mode),

2. By using FCS parameters to index into the desired block buffer and process the data there (locate mode).

When block I/O is used, a program can gain access to an entire file block. This file block is delivered to a user-specified buffer within the user program. This allows the user to eliminate the space and time overhead associated with record buffering. By opening the file as either a single-user or a shared file, a task can indicate whether other tasks can gain access to the file during its processing. Access can also be restricted by assigning specified volume and file access protection to a volume or file. See Section 3.2.8.

1.9 DEVICE HANDLER TASKS

In RSX-11D, support of PDP-11 devices is provided by device handler tasks which perform the necessary functions that enable physical I/O operations.

Because device handlers are tasks within RSX-11D, the user does not fill memory with handlers that are not required. Infrequently-used handlers can be loaded when needed by the user and then unloaded after use.

In addition, because handlers are tasks, the user can write his own handlers to service nonstandard devices. See the IAS/RSX-11D Guide to Writing Device Handler Task.

1.10 ON-LINE DEBUGGING TECHNIQUE

The on-line debugging technique (ODT) for RSX-11D is a system aid for debugging programs that have been assembled or compiled and task built. From the keyboard, the user interacts with ODT to achieve the following results:

- Print the contents of any location in the task for examination or alteration.
- Run the entire program or any portion of it using the breakpoint feature to halt the program at specified points
- Search the program for specific bit patterns
- Search the program for words that refer to a specific location
- Calculate offsets for relative addresses
- Fill a block of words or bytes with a designated value

A Trace capability is also provided to aid in the debugging of FORTRAN and MACRO-11 programs.

1.11 BATCH CAPABILITY

The RSX-11D batch facility provides for job execution in background mode. Each job can consist of control information and data. Control information includes such functions as requests to compile or assemble tasks and to execute them. A variety of switches gives the user desirable options.

The batch stream is read onto disk from a peripheral device, for example, a card reader. Once on the disk, the system examines the job requests and schedules them.

1.12 SYSTEM GENERATION

The RSX-11D software system distributed to the user is capable of performing all system functions once it is built onto the system disk. However, RSX-11D provides a system generation program that can be used to tailor the system to the processing requirements of individual installations.

1.13 TASK ACCOUNTING

RSX-11D accounting facilities enable the system manager to determine how executing tasks are using the system resources. Accounting statistics provide the following information for each task:

- The amount of computer memory used
- The elapsed execution time

- Central processor time used
- Peripheral device access count
- A log of the I/O
- Disk storage in use for individual disk areas.

Task accounting is described in detail in the RSX-11D System Manager's Guide.

1.14 HARDWARE ERROR LOGGING

An error logging facility is included in RSX-11D to enable the system manager to determine the status of the system hardware. Error logging creates a file that contains a record of hardware errors occurring on the central processing unit and peripheral devices. The system manager can request at any time, a summary of errors by device or a list of all errors. It is also possible to obtain a listing of errors that occurred during a specified time span.

1.15 ON-LINE HARDWARE DIAGNOSTICS

The hardware error diagnostic handlers are a complete set of special purpose handlers that enable the system manager to find and analyze problems on peripheral devices. These handlers run in parallel with other handlers on an active RSX-11D system.

1.16 MEMORY PARITY CHECKING

RSX-11D checks for the occurrence of parity errors in main memory and in cache on the PDP-11/70. If a parity error occurs, the system performs one of the following actions:

1. If the parity error is detected in the Executive portion of memory, system operation halts to prevent system deterioration. An analytical dump of the system can then be taken. See the description of the core dump analyzer (CDA) in the RSX-11D System Manager's Guide.

2. If the parity error is in the user's memory area, the system terminates the task, indicating that fact in the task's status information.

3. If the error occurred in the PDP-11/70 cache, the system determines whether a user-specified limit for the number of error allowed in one minute has been exceeded. If it has been exceeded, the cache group is turned off.

NOTE

The task and memory locked down as a result of a main memory parity error cannot be used until they are released by means of the MEM MCR function.

Cache errors never result in the locking of a task in memory.

1.17 SYSTEM TEST PACKAGE

The system test package is designed to systematically test the PDP-11 software while it is running RSX-11D. The hardware is also tested by a subset of the test tasks.

CHAPTER 2

FUNDAMENTAL RSX-11D CONCEPTS

RSX-11D schedules tasks to execute in defined areas of memory called partitions. The concepts of partitions, tasks, and scheduling are essential to an understanding of the capabilities and flexibility of the RSX-11D system.

2.1 PARTITIONS

Partitions contain executing tasks and tasks that are permanently resident (fixed) in memory, regardless of whether they are executing. When an RSX-11D system is generated, the system manager defines the size and number of partitions to suit the hardware configuration and processing needs. All memory not required for the RSX-11D Executive and system subroutines is available for partitions. The system manager specifies the name, size, and base address of each partition and also indicates whether it is user-controlled or system-controlled.

Unless a task is fixed in memory, it is automatically removed from the partition upon termination. A fixed task can be unfixed only when it is quiescent.

2.1.1 User-Controlled Partitions

A user-controlled partition can contain only one task at a time. Because the user has complete control over activity in such a partition, it provides an ideal environment for a real-time application. The exact partition size can be specified for tasks that are normally resident, permitting maximum use of memory.

2.1.2 System-Controlled Partitions

A system-controlled partition, one in which the RSX-11D Executive handles task execution, can contain one or more tasks at a time. The system controls the placement of tasks in memory. The only restriction is that a task cannot be loaded into a partition until there is enough contiguous memory available within the partition.

The system-controlled partitions are intended for the execution of user tasks that do not necessarily have real-time requirements, MCR functions, and other system-related tasks (for example, the I/O system). See Figure 2-1 for an example of memory divided into user-controlled and system-controlled partitions.

System controlled partitions can operate on either a priority basis or on a time-scheduled basis as defined by the system manager during system generation. The sequence of events that occur in a priority-oriented partition is described in the RSX-11D Executive Reference Manual. The algorithm for the time-scheduled partition is described in the RSX-11D System Manager's Guide.

Because of the large overhead required, tasks are not moved in memory during execution to make room for additional tasks. If sufficient memory is available within a partition, however, additional tasks can be loaded.

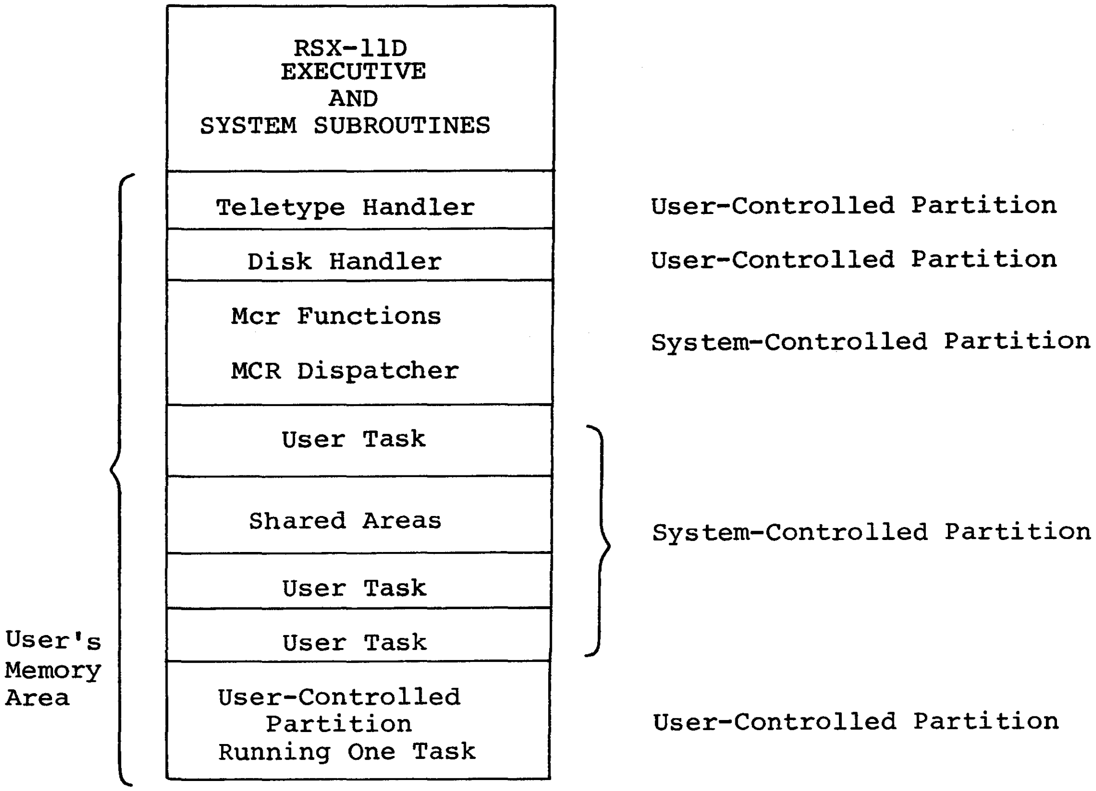

RSX-11D EXECUTIVE AND SYSTEM SUBROUTINES
Teletype Handler
Disk Handler
Mcr Functions
MCR Dispatcher
User Task
Shared Areas
User Task
User Task
User-Controlled Partition Running One Task
User-Controlled Partition
User-Controlled Partition
System-Controlled Partition
System-Controlled Partition
User-Controlled Partition
User's Memory Area

Figure 2-1
Physical Memory Divided Into Partitions

2.1.3 Memory Protection

The memory area allocated to a task is accessed via memory management hardware. A task's memory area is protected from other tasks executing in the system; and the tasks can access only those areas allocated to it. Access rights, read or write, are also controlled by the hardware.

Those portions of memory that are used by one or more tasks, such as shared libraries and common areas, also are protected. Shared libraries can be used by many tasks. For this reason, they are re-entrant (read only). Common areas can be either pure or impure (read/write); and the memory management hardware controls access. Care must be taken when a common area is shared between tasks with read/write access. In this case, the burden is on the programmer to maintain synchrony.

2.2 TASKS

A task is the basic executable unit in RSX-11D. Once a source task has been assembled or compiled, task built to make it relocatable, and installed into the system, it is referred to as a task. One or more source programs can be combined during assembly/compilation or task building to form one task.

When a task is installed in RSX-11D, the system establishes an entry in an in-memory directory called the system task directory (STD). This entry holds information concerning the task's image on disk. When a request for task execution is made, the system uses this pointer to load the task image into memory and start its execution. Tasks can also be fixed in memory, to allow faster task initiation and rescheduling by the system. Each task can execute independently or in conjunction with other tasks in its own or other partitions. Such task interaction is described in Section 2.2.3. Tasks can be requested for execution in any of the following ways:

- By a user at a terminal
- By an executing task
- By a request in a batch stream
- Through a time-dependent schedule

2.2.1 Overlays

RSX-11D supports an overlay development function that can be used to reduce the amount of memory required for a program. For example, a task can be divided into several segments: a main segment that must be resident throughout task execution and other segments that are used less frequently, which are loaded into memory only when needed.

Segments can control program operation by calling other segments. The called segments constitute the overlays. An overlay is brought into memory to replace resident code that is no longer needed. The main segment also provides a storage area for information that is to be passed from overlay to overlay.

The memory requirement for the overlay area is equivalent to the size of the largest overlay. The memory requirement for the task is the area of the main segment plus the area required for the largest overlay.

As an example of overlaying, consider the program whose task structure is depicted in Figure 2-2. The program consists of five segments: a main segment, and segments A, B, C, and D. Segments C and D can overlay each other. Within segments C and D, certain portions can also be overlaid. Figure 2-2 illustrates the space required for the program on disk and the memory area required for the program's execution.

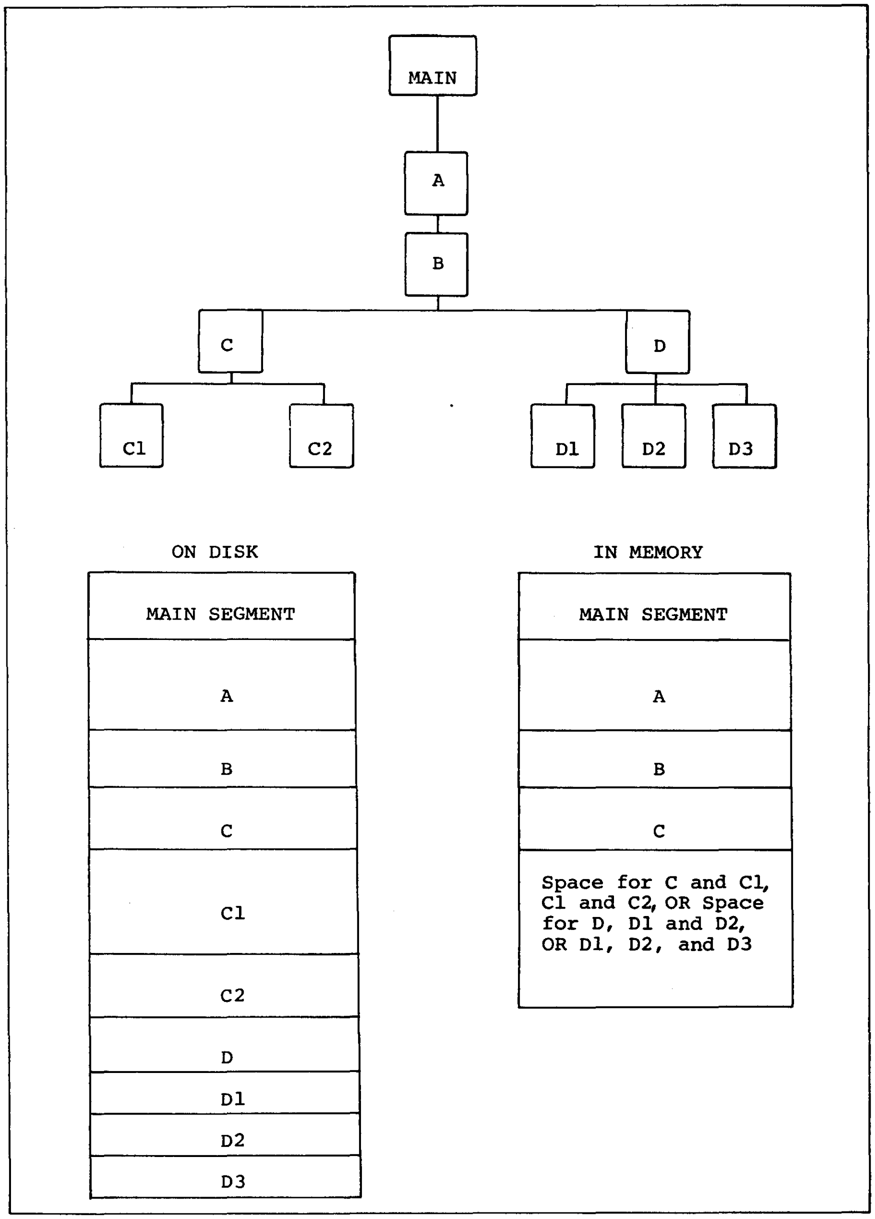

MAIN
A
B
C
D
C1
C2
D1
D2
D3
ON DISK
MAIN SEGMENT
A
B
C
C1
C2
D
D1
D2
D3
IN MEMORY
MAIN SEGMENT
A
B
C
Space for C and C1, C1 and C2, OR Space for D, D1 and D2, OR D1, D2, and D3

Figure 2-2
Overlay Structure

2.2.2 Multiuser Tasks

A multiuser task can be requested for execution by more than one user. For example, terminals 5, 7, and 8 can run the same task independently. Both system tasks and user-written tasks can be built as multiuser tasks.

In many cases, multiuser tasks are divided into pure and impure areas. The term pure area indicates that the area remains unchanged during task execution (has read-only access). The term impure area indicates that the area is modified during execution (has read/write access).

When a task consisting of a pure and an impure area is run by more than one user, only one copy of the pure area resides in memory. One copy of the impure area is required for each user. Figure 2-3 depicts a partition containing multiuser tasks. Therefore, dividing a multiuser task into pure and impure areas results in conservation of memory for the executing tasks.

2.2.3 Task Interaction

Because many tasks can execute at the same time, four techniques for task interaction are provided: global event flags, a common area, shared access to data files, and SEND/RECEIVE Executive directives.

RSX-11D provides a set of 32 global event flags that can be set and tested by all tasks in the system. Using the global flags, a task can wait for another task to perform a designated function before resuming execution. The waiting task tests a specified event flag; when the executing task has performed the designated function, it sets the flag, thus allowing the waiting task to resume. The use of global event flags should be coordinated by the system manager. Figure 2-4 depicts the use of event flags.

Common areas are read-write shareable global areas (see Section 2.7) and are created and installed in the same way as tasks. Figure 2-5 shows the use of the common area for intertask communication. Many tasks can read from or write into common areas. Thus, a task can read information from an I/O device, store it in a common area, and set a global event flag to notify another task that data is available for it in that common area.

Shared data files provide the same function as common areas except that they are maintained on a Files-11 volume rather than in main memory.

SEND/RECEIVE Executive directives allow tasks to transmit data using the system node pool as the data storage medium.

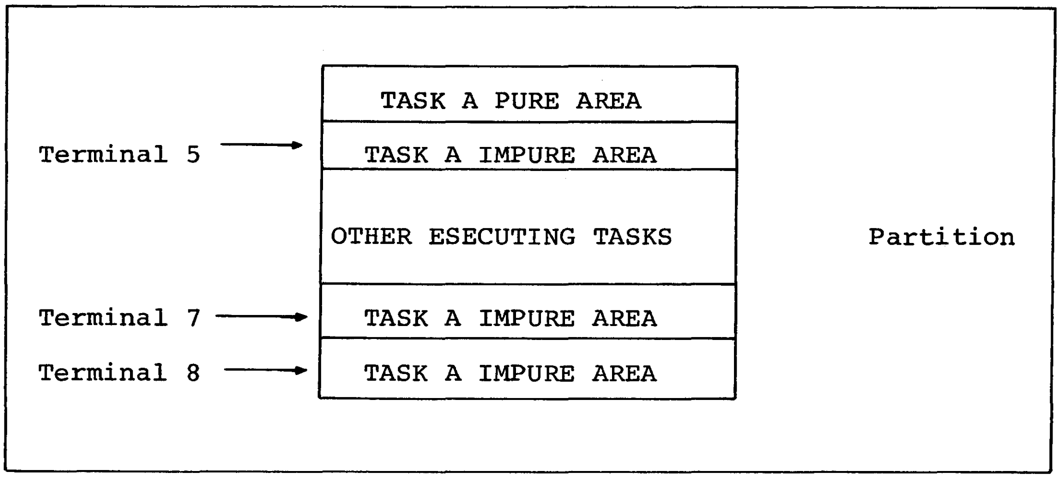


Figure 2-3
Multiuser Tasks

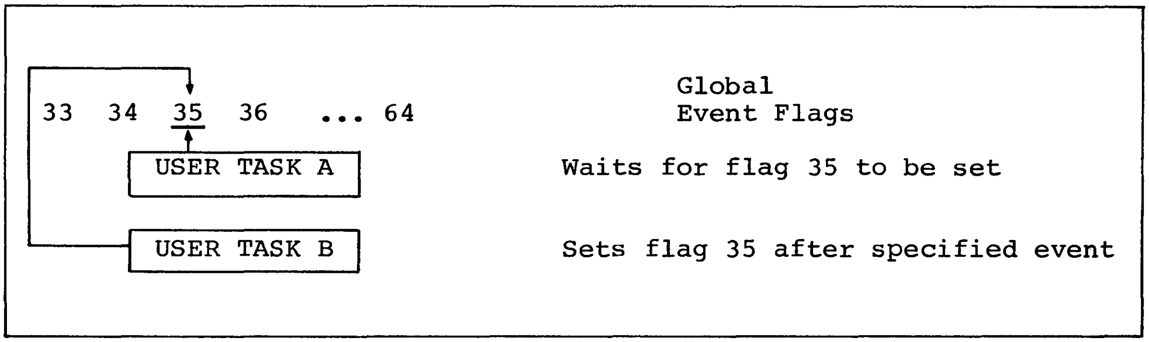

33 34 35 36 ... 64
Global
Event Flags
USER TASK A
Waits for flag 35 to be set
USER TASK B
Sets flag 35 after specified event

Figure 2-4
Use of Event Flags for Intertask Communication

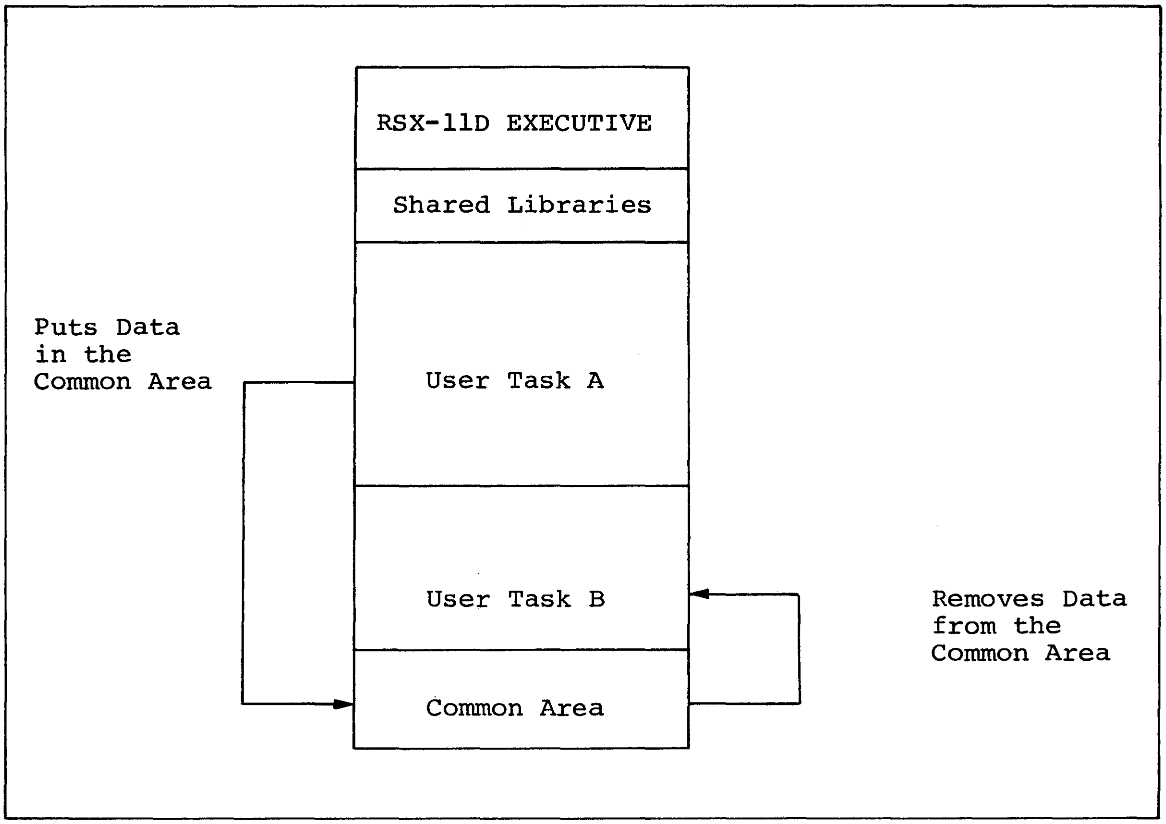


Figure 2-5
Use of the Common Area for Intertask Communication

2.3 SCHEDULING TASK EXECUTION

Under RSX-11D, a task can be requested for immediate execution or it can be scheduled for execution at a later time. The scheduling request can originate from the user or from another executing task. In a priority-oriented partition, scheduling always depends on task priority (1 through 250; where 250 is the highest or most urgent priority) and memory availability within the desired partition. In a time-scheduled partition, scheduling always depends on the time scheduler, which give each task in turn an opportunity to execute.

In addition to the scheduling performed automatically in a partition by the Executive, the user can schedule tasks based on any of the following factors:

- A request to execute the task in a number of minutes, seconds, or clock ticks from the issuance of the request, for example, in 30 minutes, and optionally, to reschedule the task at specified intervals.

- A request to execute the task at a certain time of day and at specified intervals thereafter; for example, at 10:00 p.m. and every ten minutes thereafter.

- Synchronization with the next clock tick, second, minute, or hour, and optionally, to reschedule the task at specified intervals.

When a task of a higher priority than the currently running task is scheduled for execution in a priority-oriented partition, the task currently in memory is checkpointed and replaced by the higher priority task if necessary. See Section 2.4.

Both the terminal user and tasks executing within the system can affect the order of task execution. If necessary, the user can cancel the scheduled request and then reschedule tasks at specified intervals.

2.4 CHECKPOINTING

Checkpointing is the option of suspending the operation of one task to allow the running of another task. The suspended task is placed on a user-designated checkpoint disk device in a form that allows the system to reload and restart the task when memory is again available. The task is restarted from the interruption point, not from the beginning.

In a system-controlled, priority-oriented partition, checkpointing occurs to allow the running of a task with a higher priority. In a system-controlled, time-scheduled partition, checkpointing can occur when the currently executing task exceeds its time limit.

At task build time the user must designate those tasks that cannot be checkpointed. For example, checkpointing frequently is not appropriate for real-time applications. Checkpointing, a roll-out/roll-in operation, can be used for both system-controlled and user-controlled partitions.

NOTE

A task that can be fixed in memory cannot be checkpointed.

2.5 DEVICE INDEPENDENCE

One of the major factors in determining whether a program can run is resource availability, that is, whether the necessary peripheral devices are free for use. To allow greater flexibility in the area of resource availability, RSX-11D provides device independence. This feature permits a programmer to code a program with input and output assigned to logical units rather than to specific physical devices. Before scheduling the task for execution, the user can assign physical devices to the logical units specified in the task's I/O. If the task uses the default device assignments, that is, devices that are readily available for use, the user need not assign devices.

2.6 PRINT FILE SPOOLING, DESPOOLING, AND INPUT SPOOLING

RSX-11D provides three types of file spooling:

1. Automatic output spooling to disk,

2. Automatic despooling of disk files queued for printing,

3. Automatic input spooling from a card reader to disk.

Output spooling eliminates program contention for line printers, teleprinters, and other serial output devices. Because programs do not have to wait to use a serial device, higher system throughput is achieved.

The need to wait for a particular device is removed by temporarily redirecting the output intended for the serial device to a disk file. This temporary redirection is defined as automatic output spooling. The files redirected to the disk are later produced on the original device. This process is defined as despooling.

RSX-11D provides two levels of automatic output spooling/despooling. On the simplest level, the automatic output spooler can queue files for the single device despooler which prints them one at a time on a first-in/first-out basis. Only one output device is used at a time.

On a more sophisticated level, the automatic output spooler and the terminal user both can queue files for printing. In this case, the multiple device despooler is used. It sends queued files for printing on a priority basis. Up to 28 devices can be used for despooled output simultaneously.

The input spooler eliminates program contention for the card reader and serves as a batch job input device. Card files are read onto a disk and placed under a specified UFD (see Section 3.2.5).

NOTE

Spooling can be enabled and disabled only by a privileged user.

2.7 SHAREABLE GLOBAL AREAS

Shareable global areas (SGAs) consist of shareable task image libraries and common areas. SGAs are created by the task builder, and are stored on disk.

Shareable task image libraries provide a storage area for subroutines that perform often-needed functions, for example, file control services (FCS) routines. By installing such subroutines in the system, the user avoids their duplication into each task that requires them. The one copy stored in the library can be used by all tasks without requiring additional physical space in the task's area of memory, although the task's address space is used.

Shareable libraries are dynamically loaded into memory when a task that uses them becomes active and are automatically released from memory when they are no longer required by any task running in the system.

Common areas are normally used to pass shared data from one executing task to another (see Section 2.2.3) (for example, commonly-used tables of reference data). Like shared libraries, they are dynamically loaded when required, but are written back to disk when released on the assumption that the data has been modified.

Like tasks, SGAs are files on disk. Therefore, all the file protection conventions described in Section 3.2 apply. In addition, there is a set of access rights assignable to the code and data of SGAs. When a task binds to an SGA, the requested access is specified. When the SGA is installed, the access permitted to binding tasks is specified. Finally, on installation of the task, the access requested by the task is validated against that permitted by the SGA as installed.

NOTE

> SGAs can be installed only by a privileged user, usually, the system manager.

Five SEND/RECEIVE Executive directives provide for the transfer of data blocks of up to 255 words from one task to another. The recipient task need not be in memory at the time of the transfer. The data blocks will be queued for the attention of the task when it returns to memory. SEND/RECEIVE Executive directives are described in the RSX-11D Executive Reference Manual. Their use is illustrated in Figure 2-6.

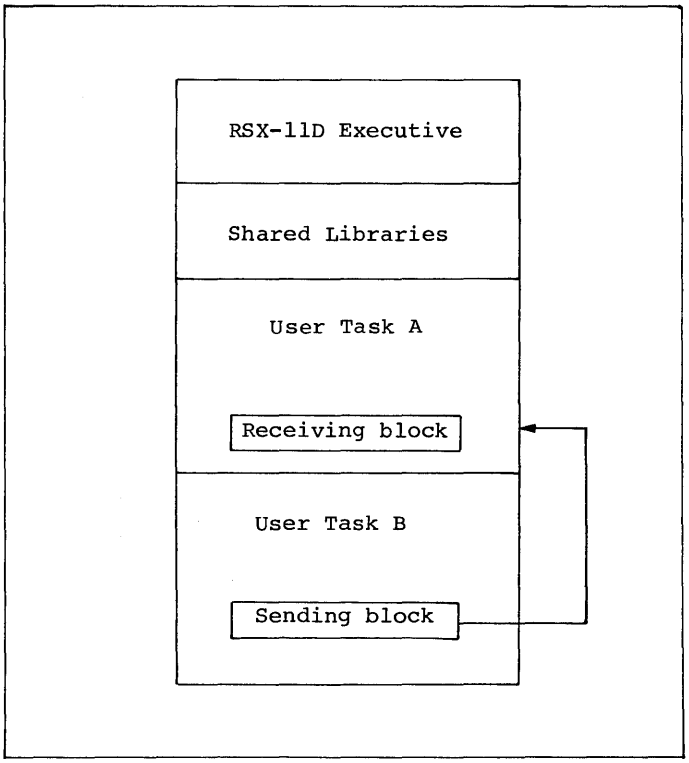


Figure 2-6
Use of SEND/RECEIVE Directive

CHAPTER 3

RSX-11D CONVENTIONS

To simplify operations, RSX-11D observes certain conventions with respect to the following items:

- Peripheral devices
- Files
- Filename strings
- Indirect files
- Terminal control

To ease the transition from other DEC systems to RSX-11D, many of the conventions used by other DEC systems are followed by RSX-11D.

3.1 DEVICE-RELATED CONVENTIONS

All peripheral devices are referred to using a 2-letter name and a 1- or 2-digit unit number followed by a colon (for example, DK0: and TT12:). If the unit number is omitted, the system uses unit 0 by default: for example LP: indicates line printer 0. Peripheral devices can be referred to by mnemonics or by pseudo-device names. Pseudo-devices are linked by the system manager to regular device mnemonics. Logical unit numbers of each task are in turn associated with peripheral devices to provide another level of device independence. Table 3-1 lists the mnemonics of devices supported by RSX-11D.

3.1.1 Pseudo-Devices

In addition to supporting actual physical devices, RSX-11D provides seven pseudo-devices: system disk (SY), terminal interface (TI), console input (CI), console output (CO), console log (CL), message output (MO), batch (BP), and spooling output (SP). The linking of pseudo-devices to physical devices permits a system manager to dynamically determine the physical devices that will send or receive system information.

Table 3-1
RSX-11D Device Mnemonics

NAME	DEVICE
TTn: (0 - 63)	User Terminal
DKn:	RK05 Disk
DFn:	RF11 Disk
DPn:	RP03 Disk
DSn:	RS03 or RS04 Disk
DBn:	RP04 Disk
MTn:	TU10 Magnetic Tape
MMn:	TU16 Magnetic Tape
DTn:	DECtape
PPn:	Paper Tape Punch
PRn:	Paper Tape Reader
LP:	Line Printer
CR:	Card Reader
CT:	Cassette Tape
AFn:	AFC11 Analog Input
LSn:	LPS A/D Converter
ADn:	AD01 A/D Converter
UDn:	UDC11 Universal Digital Controller
SY:	System Disk
TI:	Individual Terminal Interface
CI:	Console Input
CO:	Console Output
CL:	Console Log
MO:	Message Output
SP:	Spooling Output
BP:	Batch pseudo-device

3.1.2 System Device (SY)

The system device is the disk (DK, DP, or DB) that contains the operating system currently loaded into memory. The term SY indicates the device on which the system disk is mounted.

3.1.3 Terminal Interface (TI)

Terminal interface (TI) is the logical device specification for a terminal. Each terminal has a unique TI. The system uses TI to determine which terminal requested execution of a specific task. For identification purposes each task is assigned the TI of the requesting terminal. When more than one user has activated the same task, the task name and TI assignment provide the system with a means of determining which task is associated with a particular terminal.

3.1.4 Console Input (CI), Console Output (CO), and Console Log (CL)

Pseudo-device CI is used by the system manager to communicate with the system.

Pseudo-device CO is used by the system to communicate with the system manager.

Pseudo-device CL is used for the listing of files; normally output is directed to the printer.

3.1.5 Message Output (MO)

The pseudo-device MO is used to provide detailed error messages. When an error occurs, the system prints a message; however, if MO is loaded, more explicit information is also presented. The advantage of treating message output as a pseudo-device is that the MO device handler needs to be resident in memory only when additional messages would be helpful, for example, during program development. MO should be loaded when Utility programs are in use, because they depend critically on it for error message generation.

3.1.6 Spooling Output (SP)

Pseudo-device SP is used for all spooled output files. Use of SP enables the spooled output to be temporarily written to any Files-11 formatted device.

3.1.7 Batch Pseudo Device (BP)

Pseudo-device BP is used in place of TI to execute batch jobs. Use of BP permits the queue manager to identify those output files generated by tasks initiated by batch. These files are concatenated by the queue manager and queued for output once the batch job completes. Batch automatically redirects the BP device to the originating TI terminal. It temporarily sets both the originating TI terminal and the BP pseudo-device to spooling status. The TI terminal remains spooled for the duration of the execution of the batch job. The terminal reverts to its spooled or non-spooled state at conclusion of the batch job.

3.1.8 Logical Unit Numbers (LUNs)

Logical unit numbers (LUNs) provide the mechanism for programs to maintain device independence. The logical unit numbers used in a program can be assigned by means of device mnemonics to any available peripheral device that performs the desired function. LUNs can be assigned by the programmer at assembly time or at task build time, or by the task at run time. Because the system provides default LUN assignments, it is not always necessary to assign a LUN to a task. Default LUN assignments are listed in Table 3-2.

Table 3-2
Default LUN Assignments

LUN	ASSIGNMENT
1	SY:
2	SY:
3	SY:
4	SY:
5	TI:
6	CL:

3.2 SUPPORTED FILE STRUCTURES

RSX-11D provides support for two file structures: Files-11 for disk and DECtape volumes, and ANSI Standard Level 3 for single-volume or multivolume magnetic tape. Because Files-11 is an integral part of RSX-11 and its operation, it is discussed in detail in this section. For an in-depth description, see the IAS/RSX-11 I/O Operations Reference Manual.

ANSI Standard Level 3 conforms to the proposed revision of the following American National Standard document, dated June 19, 1974:

Magnetic Tape Labels

And File Structure

For Information Interchange

Order No. ANSI X3.27-1969

ANSI Standard Level 3 is also discussed in detail in the IAS/RSX-11 I/O Operations Reference Manual.

3.2.1 Files-11

Files-11 is a general purpose file control system that provides a facility for the dynamic creation, extension, and deletion of files on disk or DECtape.

Designed into Files-11 is a scheme for volume and file protection which allows the owner of a volume or file to deny access to other users in the system. This scheme for volume and file protection provides the key to system protection, in that only users with access privileges to the system device are allowed access to the system. Methods for gaining system access, and volume and file protection are explained in later sections.

3.2.2 Files-11 Volumes

A Files-11 volume is a collection of files which reside on a single disk or DECtape. The system can directly address each file on the volume by means of file pointers which reside in the volume's directory files.

Each Files-11 volume has two kinds of directory files that are used for file management: the Master File Directory (MFD) file, and the User File Directory (UFD) files.

The Master File Directory (MFD) file is automatically generated by the file system when a volume is initialized as a Files-11 volume, and is used to store pointers to all of the User File Directory (UFD) files on the volume. MFDs are described in greater detail in Section 3.2.4.

User File Directory (UFD) files are created as needed. They are used to store pointers to all of the files belonging to, or associated with, the user whose account number (UIC) corresponds with the UFD file name. UIC is described in Section 3.2.7. UFDs are described in greater detail in Section 3.2.5.

3.2.3 File Format

All Files-11 files, whether MFD, UFD, or user file, have the same basic format (see Figure 3-1 for illustration of a sample Files-11 file structure). All have a file header area, and one or more data area(s).

The file header area contains all the pertinent information required by the file system to process the file. For the purposes of this introduction, the user need be familiar with only the following fields:

- File Owner Field

  The file owner field contains the account number (UIC) of the user who created the file. UICs are described in Section 3.2.7.

- File Name Field

  The file name field contains the name assigned to the file when it was created. File names can be a maximum of nine alphanumeric characters long, and are discussed in Section 3.3.

- File Type Field

  The file type field contains the mnemonic that identifies the file by its functionality; for example, FTN defines a FORTRAN source file. File types are discussed in Section 3.3.

- Version Number Field

  The version number field identifies the particular version or generation of the file. Version numbers are discussed in Section 3.3.

- File Protection Field

  The file protection field contains a 16-bit code that describes who is allowed access to the file. It also describes the type of access allowed (i.e., Read, Write, Extend, or Delete). Volume and file protection are discussed in Section 3.2.8.

- Data Pointer Field

  The data pointer field describes the physical allocation of the file on the volume. Each data area pointer describes a physically contiguous portion of the file. See Figure 3-2.

By establishing pointers to blocked data in the files header area, as opposed to storing the data immediately following the files header, the system accomplishes two things: (1) All files on the volume have the same structural format, regardless of functionality; (2) All fragmented or noncontiguous areas of the volume can be put to optimum use; that is, a file can be expanded by merely attaching another pointer to a blocked data area to its file header.

Users should always address data in a file-relative manner. The translation of file-relative addresses into physical addresses is performed by the file system and is completely transparent to the user.

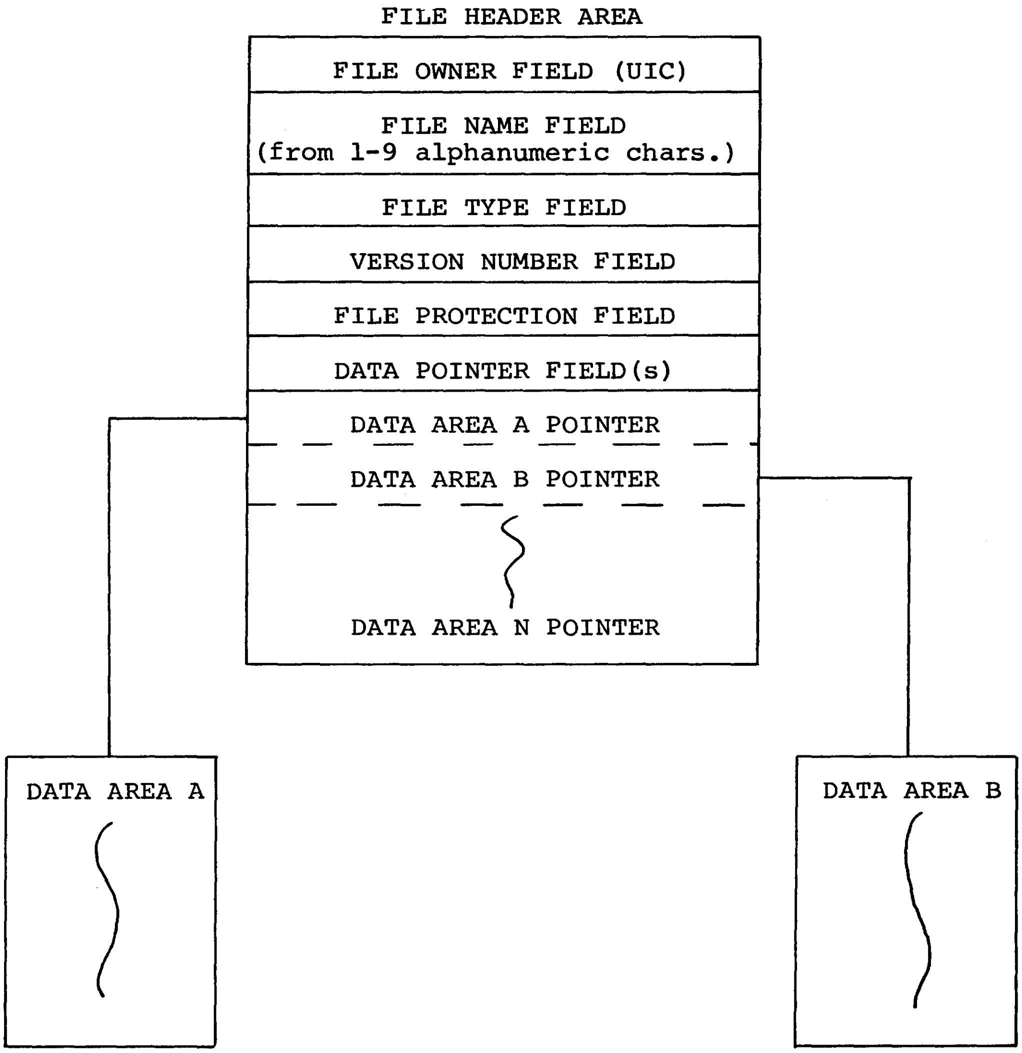


Figure 3-1
Sample Files-11 File Structure

3.2.4 Master File Directory

The Master File Directory (MFD) is used by the file system to locate requested UFDs. The MFD of the system device serves a dual purpose: it aids the file system to locate UFDs on that volume, and it is used by the operating system to identify users logging onto the system.

Under RSX-11D, only users with UICs (account numbers) that correspond to established UFD file names listed in the MFD on the system volume are allowed access to the system.

The Master File Directory (MFD) is automatically generated when the volume is initialized as a Files-11 volume. Files-11 volumes can be initialized only by the system manager or another privileged user. See Section 6.10.

Once the MFD file is established, the system manager or privileged user can begin establishing UFD files. For each UFD file created, its file name and location are recorded in the MFD file.

3.2.5 User File Directory

The User File Directory (UFD) files are used by the file system to locate user files that are associated with or owned by the UFD's owner.

UFD files are generated, as needed, by the system manager or another privileged user. See Section 6.24. When a user file is created, a pointer to the file's header area is established in the owner's UFD. The protection attributes assigned to the UFD establish the access rights to the files it points to. That is, a user whose access rights are not consistent with the protection attributes assigned to the UFD is not allowed access to any of the files pointed to. See Section 3.2.8.

3.2.6 File Ownership

File ownership is determined by the UIC stored in the file owner field of the file header. The user whose account number (UIC) exactly matches the UIC stored in the file owner field is considered, by the system, to be the file's owner.

3.2.7 User Identification Code

User Identification Codes (UICs) are unique account numbers assigned to each user by the system manager. For a UIC to be valid, the system manager or privileged user must have first created a corresponding UFD on the system device.

UICs perform the following functions:

1. Identify the user as one with authorized access to the system. See Section 5.8.
2. Identify the user as a privileged or nonprivileged user. A privileged user is assigned a UIC with a group code of 10 octal or less.
3. Identify the user's access privileges. See Section 3.2.8.
4. Establish the user's default UFD. See section 3.3.1.

NOTE

> Under RSX-11D, tasks are normally run using the UIC under which the user logged onto the system. See the discussion of the RUN command in Chapter 6 for exceptions to this rule.

Whenever a UIC specification is required, it is always enclosed in brackets and specified in the following format:

```
[ggg,mmm]
```

where:

ggg is an octal number ranging from 1 to 377 representing the user's group or department number.

mmm is an octal number ranging from 1 to 377 representing the user's unique identification within the group or department.

Example

```
[200,200]
```

3.2.8 Volume And File Protection

Files-11 provides the user with a multilevel scheme for protecting volumes and files against unauthorized access. This is accomplished by allowing the user to specify protection attributes for the entire volume, as well as for each file within the volume regardless of the file's hierarchy or functionality.

Four types of actions can be permitted on a file; read, write, extend, and delete. Tasks that are allowed to perform any combination of these actions are also divided into four categories:

- System - This category comprises all tasks that run under a system account number. System account numbers (UICs) are those having a group number of 10 octal or less; i.e., [2,200].
- Owner - This category comprises all tasks that run under a UIC that matches the UIC in the file's file owner field.
- Group - This category comprises all tasks that run under a UIC whose group number matches the group number of the UIC in the file owner field of the file being accessed.
- World - Any task not included in the three categories described above.

Every file has a file protection field associated with its file header. This field contains a 16-bit code which describes the file's protection attributes consistent with the categories of user tasks described above. Figure 3-3 shows the format of the file protection word. This 16-bit code is divided into four categories: System, Owner, Group, and World. Each category is subdivided into action fields. These action fields describe the type of action (read, write, extend, and delete) that each category of task is allowed to perform on the file. Before file access is allowed, this file protection code is interrogated by the file system to determine if the requesting task is allowed to perform a desired action.

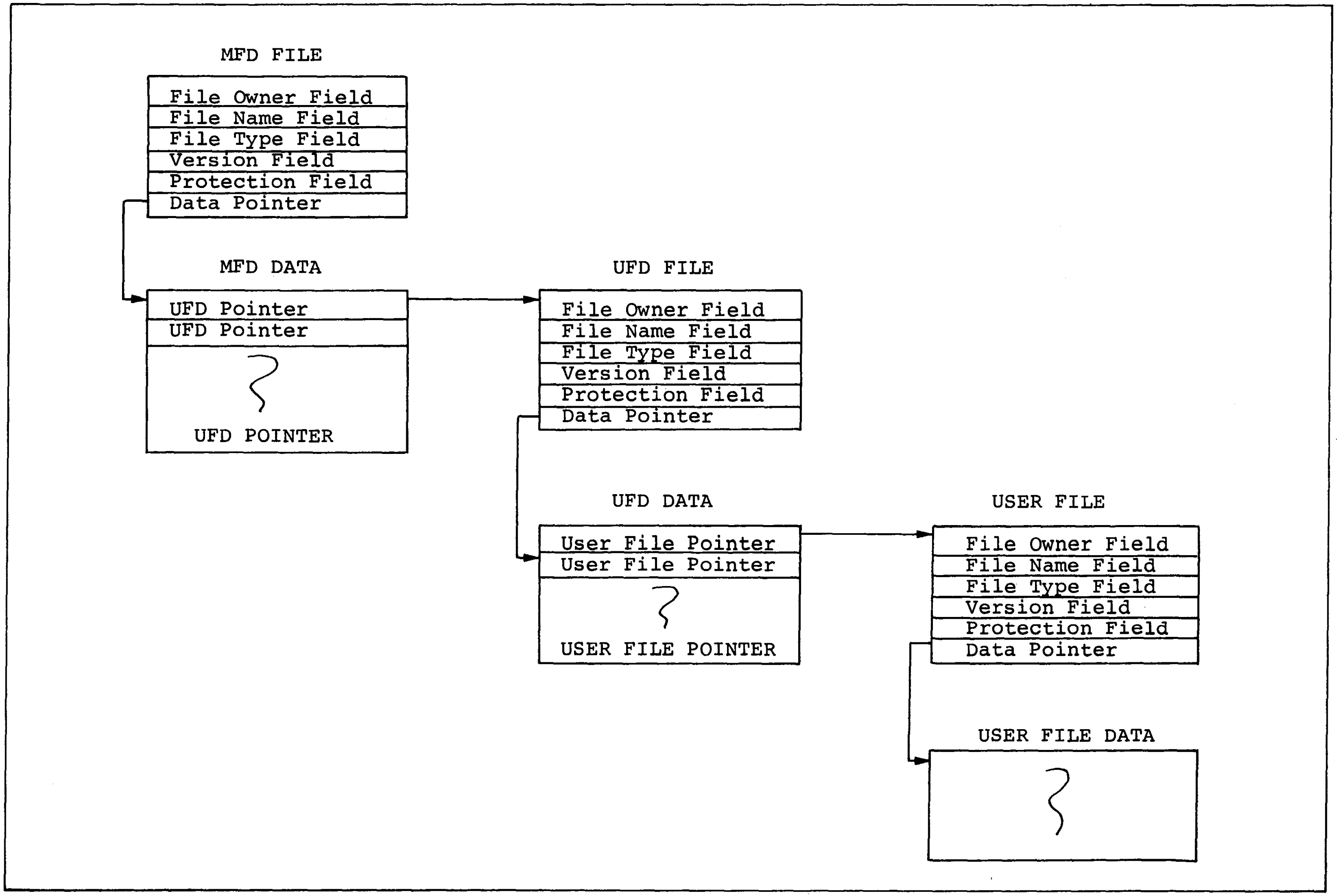
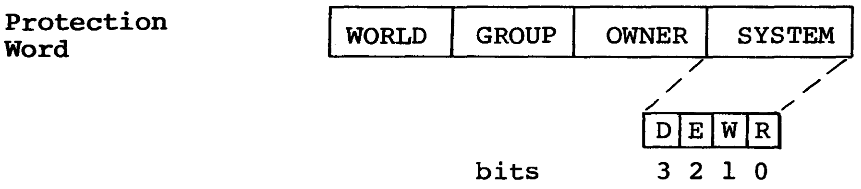


Figure 3-2
Files-11 File Access Scheme



NOTE

Bit set means NO access permitted

Example

	WORLD	GROUP	OWNER	SYSTEM
Protection Word	1 1 1 1	1 0 0 0	0 0 0 0	0 0 0 0

In this example, all system and owner accounts are allowed read, write, extend, and delete access privileges; group accounts are allowed read, write, and extend only; and world accounts are denied all access privileges.

Figure 3-3
Format of File Protection Word

When a task attempts to access a volume or file, the file system performs the following checks to ensure that the user task is allowed access:

1. The task UIC is compared to the file owner UIC to determine the task's category (system, owner, group, world); see Figure 3-4.

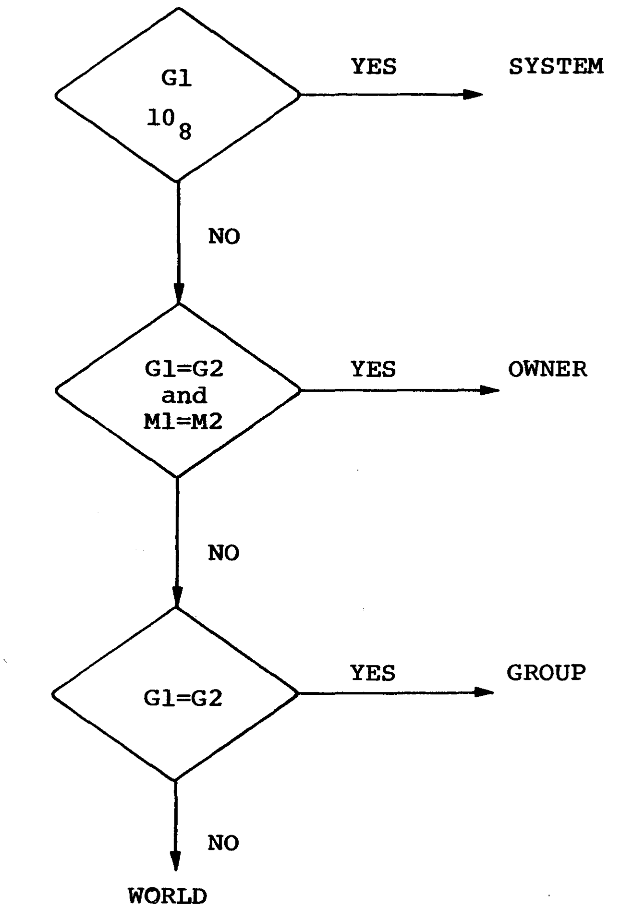
2. When the task category is determined, the file system interrogates the file protection word to determine if the task's category is allowed access to the file.

3. The file system then further analyzes the file protection word to determine if the function to be performed (read, write, extend, or delete) is an allowable function for this category of task.

If all of the above checks yield positive results, that is, the task is authorized file access, then access is granted.

NOTE

> This check is performed three times; once for the volume, once for the UFD, and once for the particular file. The task must pass all three checks.



Legend

G1 = Group number of task UIC
MI = Member number of task UIC
G2 = Group number of file owner UIC
M2 = Member number of file owner UIC

Figure 3-4
Formula For Determining Task File Access Category

3.3 COMMAND SYNTAX AND FILE SPECIFIERS

The majority of commands issued to RSX-11D have the following format:

```
outputfile/switch=inputfile1/switch1,inputfile2/switch2,...,
inputfilen/switchn
```

Both input and output files are designated using file specifiers. Input and output file specifications are separated by an equal sign (=). Optional switches are used to indicate desired actions. File specifiers have the following format:

```
dev:[ufd]filename.type;version/switch
```

dev: = the physical device on which the volume containing the desired file is mounted (for example, DK0: or DT1:). The colon is required as part of the device specifier.

[ufd] = the user file directory containing the desired file.

filename = the name of the file. In RSX-11D, filenames can be up to nine alphanumeric characters in length. Filename and type are always separated by a period.

.type = a means of distinguishing among various forms of the same file. For example, a source FORTRAN file might be named COMP.FTN, while the object file associated with that program might be called COMP.OBJ. File types are discussed in Section 3.3.2. The file type and version are always separated by a semicolon.

;version = an octal number used to differentiate among versions of a file. For example, when a file is first entered using the Editor, it is assigned a version number of 1. If the file is subsequently opened for editing, the Editor keeps the original file for backup and creates a new file with the same filename and type, but with a version number of 2. Version numbers can range from 1 to 77777 octal.

/switch = is usually an optional qualifier. Switches are normally used either to direct the execution of a task, or to qualify an input parameter.

3.3.1 Defaults in File Specifiers

If any of the elements of the filename string, except the filename, is omitted, the system uses a default value. A task can establish defaults for a file. When a task does not specify defaults, the defaults listed in Table 3-3 apply.

Table 3-3
Filename Specifier Defaults

STRING ELEMENT	SYSTEM DEFAULT
dev:	If omitted in the first or only file specifier, SY is assumed. If omitted in the second through n file specifiers, the device, specified or default, for the previous file specifier is used.
[ufd]	If omitted in the first or only file specifier, the UFD which corresponds to the UIC under which the task is running is used. If omitted in the second through n file specifiers, the UFD, specified or default, for the previous file specifier is used.
filename	No default -- must be specified either explicitly or implicitly using an asterisk as described in Section 3.3.3, except in certain PIP and FLX strings.
.type	A meaningful file type (e.g., if a list file is created, the system assigns it a file type of .LST). See Section 3.3.2 for a list of default file types.
;version	For input files, the most recent version number. For output files, the next higher version number. An exception is the PIP file delete function that requires an explicit version number; this feature prevents the user from inadvertently deleting the latest version of a file.

3.3.2 RSX-11D File Types

RSX-11D has a defined set of file types that the system uses when the type is omitted from a filename string. While the user can assign his own file types, the file types presented in Table 3-4 are easy to use. File types can be defaulted in most commands. For example, if the FORTRAN compiler is given the source file XYZ to compile, it automatically searches for filename and type XYZ.FTN.

Table 3-4
RSX-11D File Specifier Defaults

FILE TYPE	MEANING
BIS	A Batch input stream
CMD	A file containing a list of commands to a system program (i.e., an indirect file)
DAT	A data file (as opposed to a program file)
DIR	A directory file (e.g., a UFD directory)
EXE	A Batch execute file
FOR	A Batch FORTRAN file
FTN	A FORTRAN language source program
LIB	A Batch Library file
LIS	A Batch listing stream
LST	A file in print-image format
MAC	A MACRO assembly-language source program
MAP	A file associated with Task Builder output
MLB	A macro library file
OBJ	An object program (output from MACRO or FORTRAN)
ODL	An overlay file
OLB	An object library file
SAV	An RSX-11D system image file, or a file generated by the EDI utility program
SML	A System Macro Library file
SOB	A Batch concatenated object file
SPR	A spooled output file
SRT	A Sort work file
STB	A symbol table file
TMP	A temporary file
TSK	A file that has been task built

3.3.3 Asterisk Conventions (wildcards)

In addition to relying on defaults in the filename string, the user also can place asterisks (*) in the string to indicate a particular element of the string that is to be ignored (wildcard). By ignoring the content of elements within a string, the user can affect more than one file at a time. For example, to delete three files named PROG.MAC;1, PROG.OGJ;1, and PROG.TSK;1, the following wildcard specification can be used rather than three explicit delete requests.

```
PIP>PROG.*;1/DE
```

The four characters, PIP>, printed by the system as a prompting sequence, indicate that the PIP utility is running. PIP is used to delete the files. The rest of the command is typed by the user.

If the file PROG exists with other version numbers (e.g., PROG.MAC;2), they are not affected by the delete request.

Not all system programs accept wildcards; their primary use is in conjunction with commands to PIP.

An asterisk can be placed in any portion of the filename string except the device indicator, which must always be specified or defaulted to SY.

3.3.4 Examples of Command Strings

In the following examples, the first three letters and the right angle bracket (>) are printed by the system. The remaining characters are typed by the user.

```
MAC>DK0:[200,200]CRGPT=DK0:[200,200]CRGPT
```

A request to assemble CRGPT.MAC and call the object program CRGPT.OBJ. Both files are on DK0 under [200,200].

```
PIP>ABCD.MAC=ABBD.MAC/RE
```

A request to rename (/RE) file ABBD.MAC to ABCD.MAC. Both files are on device SY under the default UFD.

```
PIP>SBG.OBJ;5/DE
```

A request to delete file SBG.OBJ;5. The output filename string is omitted because it is not applicable.

```
TKB>CRGPT=CRGPT
```

A request to task build the object file CRGPT.OBJ. The output is named CRGPT.TSK because TSK is the default file type for the Task Builder.

```
PIP>CRGPT.*;*/DE
```

A request to delete all files with the name CRGPT, regardless of file type or version.

3.4 INDIRECT FILES

An indirect file is a sequential file containing a list of commands. The commands contained in the file are task specific, that is, they can be interpreted only by a specific task such as MACRO-11 or the Task Builder.

Rather than repeatedly typing commonly used sequences of commands, the user can type the sequence once and store it on a file. To execute the sequence, the user types an at sign (@) and then the filename. The affected task (for example, MACRO-11 or Task Builder) locates the indirect file and executes the commands contained therein. For example,

```
MCR>MAC @CMDFIL.CMD
```

In response to the command, MACRO-11 accesses the file CMDFIL.CMD for all of its commands.

NOTE

1. An indirect file may contain any command that is interpretable by the task to which it is directed, but no others.
2. Indirect files can be created using the Editor utility program, as described in the RSX-11D Utility Programs Procedures Manual.
3. Not all system programs allow the use of indirect command file specifiers. The user should carefully read the appropriate system program documentation before attempting to use the programs.

3.5 TERMINAL CONVENTIONS

Several special keyboard characters that cause specific functions to be performed are recognized by the system. These functions are described in Table 3-5. Most of them require the holding down of the CTRL (control) key while a second character is pressed; e.g., CTRL C.

Table 3-5
Terminal Control Conventions

KEYS	FUNCTION
CTRL C	Causes MCR to be activated. The system prints the prompt characters, MCR>. NOTE Typing CTRL C does not affect the execution of any tasks currently running.
CTRL Z	Logical end-of-file; when typed in response to a prompt from most system programs, causes that program to exit.
RETURN	Terminates the current line and causes the system to print the prompt for the next command. All lines are terminated using RETURN unless otherwise noted in the documentation.
RUBOUT	Causes the most recently typed character to be deleted from the command string. RUBOUT echoes as a backslash (\). Successive typing of RUBOUT causes the most recently typed characters to be deleted. One character is deleted for each successive pressing of RUBOUT.
ALT	Terminates MCR. Normally used when requesting a program (user or system) that is to interact with the operator after the command is executed (e.g., RUN ...MAC <ALT>). NOTE On some keyboards, the ALT key is labeled "ESC".
CTRL I	Causes a horizontal tab. Tab stops are set, by the software, at every eighth character position (e.g., 9, 17, 25, 33).
CTRL K	Causes a vertical tab of one line.
CTRL L	Causes a form feed to the top of the next page.

Table 3-5 (Cont.)
Terminal Control Conventions

KEYS	FUNCTION
CTRL O	Interrupts system output to the terminal. Successive pressings of CTRL O cause output to start and stop. For example, if a directory listing on the terminal is requested and the first few lines present the desired information, CTRL O can suppress the printing of the rest of the directory. (Large TTY Handler only)*
CTRL R	Causes the system to print the current terminal line. When errors make the line difficult to read, pressing CTRL R produces a clear copy. Input can continue on the newly-printed line (large TTY handler only).
CTRL U	Cancels the current input line. The prompt (e.g., MCR> or PIP>) is not printed on the next line, but the system is ready for the typing of a new command.
CTRL X	Causes a user-written program named TTYNnn to execute; nn is the two digit unit number of the terminal with which the program is to interact during execution.
CTRL S and CTRL Q	These two keys correspond to XON and XOFF respectively. (Pressing CTRL Q (XOFF) stops output to the terminal until CTRL S (XON) is pressed. Unlike CTRL O, the XON/XOFF function stops and starts output without any loss of characters. The silo of the VT05 terminal generates XON/XOFF automatically. (Large TTY Handler only)*

*The system manager is responsible for installing the large TTY handler. Installation usually is performed during system generation.

3.6 CARD READER CONTROL CONVENTIONS

The card reader handler recognizes three control characters. The control character code identifies the alphanumeric format for either 029 card codes or 026 card codes, or it indicates end-of-file. Table 3-6 defines the control characters.

Control character codes appear in column 1 of a multipunch control card; however, they are not transferred to the user's buffer nor included in the word count.

Table 3-6
Card Reader Control Conventions

CARD COLUMN 1 MULTIPUNCH	MEANING
12-11-0-1-6-7-8-9	End-of-file ASCII mode*
12-0-2-4-6-8	029 mode
12-2-4-8	026 mode

NOTE

Control Cards must enclose input data streams.

*End-of-file binary mode requires the same multipunches in card columns 1 to 8.

CHAPTER 4

USER/SYSTEM INTERFACE

4.1 MONITOR CONSOLE ROUTINE

The Monitor Console Routine (MCR) is the terminal interface between the user terminal and the RSX-11D operating system. This interface is accomplished by means of simple MCR System Commands.

MCR's system command repertoire enables the user to perform the following functions.

- Gain access to the system
- Initiate and terminate execution of user/system programs
- Adjust, modify, and control the system environment

4.1.1 MCR Organization

MCR services are organized internally as a re-entrant dispatcher task and a set of independent system command tasks. The individual command tasks are called by the dispatcher task as needed and are released automatically after execution.

The dispatcher portion of MCR must be active in order to receive requests (system commands) for MCR function tasks from the user terminal.

4.1.2 Command And Slave Terminals

User terminals consist of two distinct classes: command terminals and slave terminals.

Command terminals are used to activate MCR and thus gain access to the system. Once system access is established, the terminal is used to initiate system commands, or respond to system prompts.

Slave terminals, on the other hand, are passive. They can only respond to or interact with user tasks. For example, a slave terminal might be used in a process control environment to display or print out data relevant to an ongoing operation; or in a data processing environment to communicate data to an order-entry system.

NOTE

> Any attempt to use a designated slave terminal as a command terminal is ignored by the system.

The classification (command or slave) of terminal use is the responsibility of the system manager, or another privileged user. The system manager can establish a terminal as being system or slave by issuing the SET system command. The SET command is described in Chapter 6.

4.1.3 Activating MCR

The dispatcher portion of MCR must be active in order to receive requests (system commands) for MCR function tasks from the user terminal. Activating MCR consists of typing CTRL C on the terminal keyboard. When MCR becomes active, the following prompt is issued:

```
MCR>
```

At this point the link between the user and MCR is established.

If the user has not gained access to the system (i.e., logged-on), the only MCR system command available for use is the HELLO command (see Section 5.8). The HELLO command identifies the user to the system, establishes the user's privileges (see Section 4.1.4), and grants the user access to the system. When access has been granted, the user can execute MCR system commands and access files consistent with his access privileges.

NOTE

> The typing of CTRL C does not affect the execution of any tasks currently running.

4.1.4 User Privileges

The system determines user privileges by the UIC under which the user logs-on. Privileged users are assigned UICs whose group code is 10 octal or less. Nonprivileged users are assigned UICs with group codes greater than 10 octal. UICs are assigned to the user by the system manager.

Two categories of users are granted access to the operating system: privileged and nonprivileged.

A nonprivileged user uses the system on a day-to-day basis, but is not responsible for its maintenance and control. A subset of MCR system commands is provided for this user. These commands are described in Chapter 5.

A privileged user, like a nonprivileged user, uses the system on a day-to-day basis, but also has responsibility for the maintenance and control of the operating system. This user has access to the general

MCR system commands described in Chapter 5 as well as to the privileged commands described in Chapter 6.

4.1.5 Entering System Commands

System commands are entered in response to the MCR prompt (MCR>). Once the prompt is issued, MCR waits for a system command to be entered immediately following the right angle bracket character (>). If the system command is not entered within a preset period of time, MCR times out. (See Section 4.1.6). To reactivate MCR, the user need only retype CTRL C.

System commands (command lines) are entered in the following format.

```
          MCR>Command [command string]
```

where:

Command — is the system command function to be performed.

command string — if required, are command modifiers or qualifiers. Command string formats vary from command to command, and for this reason, are individually described in their respective command descriptions. See User Command Chapters 5 and 6.

The system command line, typed in by the terminal user, is stored by the MCR dispatcher. The dispatcher then initiates the corresponding MCR system command task and exits. The MCR system command task then interprets the command line that the dispatcher stored and performs the desired function. The command task executes under the UIC under which the user logged in.

When the system command task exits, the MCR dispatcher is or is not reactivated depending on how the command line was terminated. If the command line was terminated by a RETURN, the MCR dispatcher is reactivated for that terminal. If the command line was terminated by an ALTMODE, the MCR dispatcher does not reactivate. In this case, the user must enter CTRL C to reactivate to MCR.

4.1.6 MCR Timeout

MCR waits for a specific length of time (5 minutes set at system generation, unless changed by the system manager) for a command to be entered. See the SET command in Chapter 6. If this wait time is exceeded, MCR times out. At timeout, all information, since the previous line terminator is lost.

When MCR times out, a carriage return occurs, and the terminal does not respond to commands typed in by the user. To re-establish communications with MCR, the user must type CTRL C.

4.2 TASK RUN-TIME UICs AS FUNCTION OF USER TYPE

Table 4-1 illustrates what the UIC of an executing task will be as a function of how the task is initiated by a privileged or nonprivileged user and the state of the task prior to initiation. Also refer to Section 5. for non-privileged versions of run command and Section 6. for privileged version.

A user can initiate a task from a terminal in any one of four ways:

```
1.  MCR>xyz               e.g., MCR>PIP

2.  MCR>RUN ...xyz        e.g., MCR>RUN ...PIP

3.  MCR>RUN $file-specification    e.g., MCR>RUN $EDI

4.  MCR>RUN file-specification     e.g., MCR>RUN [200,200]MYTASK
```

4.2.1 Error Reporting

Errors encountered by system command tasks are reported in the form of error messages listed on the user terminal. Error messages consist of a 3-letter command task name and two dashes (--) followed by the error message text.

Example

```
SYS -- COMMAND SYNTAX ERROR
```

The RSX-11D system error messages are listed and described alphabetically in Appendix A.2. The only messages that do not have the 3-letter prefix are those issued by the task MCRERR. Section A.2.1 describes these messages.

Table 4-1
UIC of an Executing Task

	TASK ALREADY INSTALLED		TASK ALREADY INSTALLED, BUT TO BE REMOVED WHEN NO ACTIVE VERSIONS		TASK NOT INSTALLED	
Method of Initiation	Privileged User	Nonprivileged User	Privileged User	Nonprivileged User	Privileged User	Nonprivileged User
1	User's UIC	User's UIC	Not applicable	Not applicable	Not applicable	Not applicable
2	Task's UIC	Illegal	Not applicable	Not applicable	Not applicable	Not applicable
3	User's UIC	User's UIC	User's UIC	User's UIC	User's UIC	User's UIC
4	Task's UIC	Illegal	Task's UIC	User's UIC	Task's UIC	User's UIC

NOTES

1. The user's UIC is the UIC specified in the HELLO command when the user logged onto the system.

2. The task's UIC is the UIC specified during the building of the task, or when the task is installed, or when an explicit UIC is given in a RUN command. If no UIC is specified, the user's UIC becomes the task's UIC when the task is actually activated.

CHAPTER 5

GENERAL USER COMMANDS

The general user commands listed below are available to all users. Commands are presented in alphabetic order in this chapter.

- ABORT
- ACTIVE TASK LIST
- BYE
- CONTINUE
- DISMOUNT
- HELLO
- LOG
- LUNS
- MOUNT
- OPERATE
- PASSWORD
- QUEUE
- RESUME
- RUN
- SYS
- TIME
- WHO

NOTE

Please reread the Preface for explanation of characters (, /, [], etc.) that indicate command format conventions.

5.1 ABORT COMMAND (ABO)

FUNCTION:

The ABORT (ABO) command allows the terminal user to terminate the execution of tasks which have been initiated from that terminal.

FORMAT:

```
ABO[RT]Δtaskname[,taskname,...]
```

where:

taskname is the name of the task being aborted.

EXAMPLE:

```
MCR>ABO ...MAC,RICK
```

The above command causes the execution of the tasks named MAC and RICK to be terminated.

5.2 ACTIVE TASK LIST COMMAND (ACT)

FUNCTION:

The Active Task List (ACT) command enables the terminal user to obtain a list of the tasks active within the system, along with status information about the tasks.

FORMAT:

```
ACTΔ[taskname][/switch(s)]
```

where:

taskname — is the name of the task whose status information is to be listed. If taskname is not specified, all of the tasks having the same TI as the terminal from which the command is issued are listed.

/switch(s) — is one or more of the following switch options:

NOTE

If no switch is specified, the following information is listed for all active tasks having the same TI as the terminal from which the command was issued:

Task Name

Task Status (see Table 5-1)

TI Device

/FU — specifies that a full listing of the named task's status information is to be generated. Use of this switch has the following restrictions:

1. It can be specified only when taskname is specified.

2. It cannot be specified in conjunction with the /ALL switch.

   Specifying /FU causes the following information to be listed:

   Task Name

   Task Status (see Table 5-1)

   TI Device

   Run Priority

   Partition

   Real Memory Starting Address

   Number of Active Versions

   Requesting Task's Name

   Registers

   RS, PC, R0, R1, R2, R3, R4, R5, SP

   Event Flags 1-16

   Event Flags 17-32

   ATL Flags Word

   STD Flags Word

   Event Flag Masks

   Mark Time Count

   ATL Node Address (Kernel Virtual)

   STD Node Address (Kernel Virtual)

   Maximum Task Size

   I/O Pending Count

   I/O In Progress Count

   Pool Limit Count

   Pool Usage Count

   Global Libraries and Commons (if any)

   PARS-- 0-7

   PDRS-- 0-7

Table 5-1
ACT Task Status Characters

CHARACTERS	DESCRIPTION
LQ	Load request queued
LS	Load request unqueued
LF	Load request failed
RN	Task running
AQ	Task AST queued
SU	Task suspended
W0	Task waiting for event flag 1-16
W1	Task waiting for event flag 17-32
W2	Task waiting for event flag 33-48
W3	Task waiting for event flag 49-64
W4	Task waiting for event flag 1-64
EX	Task exited
IR	I/O rundown required
IP	I/O rundown in progress
ID	I/O rundown finished
IF	I/O rundown failure
TF	Task termination fault
TR	Task termination notice requested
TS	Task suspended for termination notice
SC	Task suspended for checkpoint
RQ	Record request queued
RS	Record request succeeded
RF	Record request failed
PE	Parity error

/SH

Specifies that a shorter version of the /FU listing is required. The following information is listed:

Task Name

Task Status (see Table 5-1)

TI Device

Run Priority

Partition

Real Memory Address

/ALL

Specifies that every active task in the system is to be listed. Use of this switch has the following restrictions:

1. It cannot be specified in conjunction with the /FU switch.
2. It cannot be specified in conjunction with the /TI=dev switch.

The following information is listed about each task:

Task Name

Task Status (see Table 5-1)

TI Device

/TI=dev

Specifies that all of the tasks of a specific TI are to be listed. dev is the terminal identifier for the specified TI. The following information is listed about each task:

Task Name

Task Status (see Table 5-1)

TI Device

EXAMPLES:

1. MCR>ACT

```
TT....  WO TT07
...ACT RN TT07
 ..PIP WO TT07
```

2. MCR>ACT ...PIP

```
...PIP WO TT07
```

3. MCR>ACT ...PIP/SH

```
...PIP W0 TT07 055 GEN    233300
```

4. MCR>ACT ...PIP/AL/SH

```
...PIP W1 TT01 055 GEN    213300
...PIP W1 TT02 055 GEN    244400
```

5. MCR>ACT ...FLX/FU

```
...FLX W0 TT00 050 GEN    371000 ACT VERS 002 REQ TASK ...MCR
REGS 174000 002510 021322  000010  027522  000000  000101  000101
000342
EV 1-16 000000 EV 17-32 140000 ATL FLGS 000200 STD FLGS 060100
EV MASKS 000002 000003 040647 030040 MKTM CNT 000
ATL ADDR 107000 STD ADDR 103700 TSK SIZE 063700
I/O PEND 003 I/O PROG 001 POOL LIM 040 POOL USE 014
GBLS SYSRES
PARS 003720 004120 004320 004520 000000 000000 000000 001747
PDRS 077606 077606 077606 013206 000000 000000 000000 071502
```

6. MCR>ACT /ALL

```
DP.... W0 TT00
TT.... W0 TT10
DK.... W0 TT00
LP.... W0 TT00
DF.... W0 TT00
DT.... W0 TT00
MT.... W0 TT00
...MCR W0 TT10
...ACT W0 TT07
MO.... W0 TT05
...TKB RN TT00
```

7. MCR>ACT /TI=TT10

```
TT.... W0 TT10
...MCR W0 TT10
```

8. MCR>ACT ...PIP/TI=TT2/FU

```
...PIP W1 TT02 055 GEN    213300 ACT VERS 001 REQ TASK ...MCR
REGS 174004 167762 004644  013522  000000  006003  000040  013710
000236
EV 1-16 000000 EV 17-32 040000 ATL FLGS 000200 STD FLGS 020100
EV MASKS 100000 000256 000401 000002 MKTM CNT 000
ATL ADDR 106300 STD ADDR 100660 TSK SIZE RO 007300 RW 021600
I/O PEND 002 I/O PROG 000 POOL LIM 040 POOL USE 003
PARS 002143 002351 002306 000000 000000 000000 000000 001747
PDRS 061206 035102 021206 000000 000000 000000 000000 071502
```

5.3 BYE COMMAND (BYE)

FUNCTION:

The BYE command provides the user with a facility for logging-off a terminal. The only command MCR recognizes after this command is executed is HELLO. The BYE command has no effect on active tasks previously initiated from this terminal.

FORMAT:

```
BYE
```

EXAMPLE:

```
MCR>BYE
```

5.4 LIST COMMON BLOCKS COMMAND (COM)

Refer to the SYS /COM command in Section 5.18.

5.5 CONTINUE COMMAND (CON)

FUNCTION:

The CONTINUE Command (CON) is used to continue the execution of a task that as selected the SUSPEND action parameter option when using the message output handler task.

FORMAT:

```
CON[TINUE]Δtaskname[,taskname,...]
```

EXAMPLES:

```
MCR>CON XKE

MCR>CONTINUE XKE,NK111
```

In the examples above the respective requests are to:

1. Continue task XKE.
2. Continue task XKE and task NK111.

5.6 DEVICES COMMAND (DEV)

Refer to the SYS /DEV command in Section 5.18.

5.7 DISMOUNT VOLUME COMMAND (DMO)

FUNCTION:

The Dismount Volume (DMO) command allows the user to logically dismount (make invisible to the system) a previously mounted volume. DMO deletes the Volume Control Block (VCB) and its extensions, thus declaring the volume to be logically off-line, and rendering it inaccessible.

Before shutting down an RSX-11D system, all volumes should be logically dismounted to ensure that no files are currently being accessed. Any files currently being accessed when the system is shut down may contain inconsistent data.

If the volume contains files currently being accessed, the following message is issued, and the dismount operation is aborted:

```
DMO -- VOLUME BUSY.  TRY AGAIN LATER
```

FORMAT:

```
DMOΔdev:[volumelabel][/UIC=[uic]][/LOCK]
```

where:

dev: is the device specifier for the device on which the volume is mounted.

volumelabel
is label of the volume being dismounted. If specified, the value entered is compared with the corresponding value in the VCB. If unequal, the volume is not dismounted.

/UIC=[uic]
is the volume UIC. Like the volume label, it is compared to its corresponding entry in the VCB. If unequal, the volume is not dismounted.

/LOCK If this option is specified, the volume is marked for dismount regardless of whether files are currently accessed on it. Further accesses are then disallowed and the volume is actually dismounted only when all current file accesses are terminated.

If the DMO command is issued for any volume of a multivolume magnetic tape file, all of the mounted volumes are automatically dismounted.

The following message is issued by the file processor when the logical dismount is completed and it is safe to physically remove the volume from the system:

```
     F11ACP - - DK0:  ** DISMOUNT COMPLETE **
```

The first six characters of the message (F11ACP) are the task name of the file processor, and may vary with the device and the manner in which it was mounted.

EXAMPLES:

1. `MCR>DMO DK1:`

   In this example, the VCB for the volume mounted on DK1: is deleted, and the volume becomes inaccessible.

2. `MCR>DMO DK1:RICKSVOL/UIC=[300,300]`

   In this example, the VCB for the volume mounted on DK1: is interrogated. If it contains a volume label of "RICKSVOL" and a UIC of "[300,300]", the VCB is deleted, and the volume becomes inaccessible; otherwise, the volume remains mounted.

3. `MCR>DMO MT1:VOLUM1`

   In this example, the VCB for the volume mounted on MT1: is interrogated. If it contains a volume label of VOLUM1 the VCB is deleted; otherwise the volume remains mounted.

If the specified volume is one of a multivolume magnetic tape file, all of the mounted volumes of the file are automatically dismounted.

5.8 HELLO COMMAND (HEL)

FUNCTION:

The HELLO command (HEL) allows the user to log onto a terminal and be identified as a valid user by the system. It also identifies the user as a general or privileged user. (See Section 4.1.4.)

FORMAT:

```
HEL[LO]Δ[uic]
```

NOTES

1. If the user-entered [UIC] (brackets required) does not have a corresponding UFD on the system device (SY:), system access is denied and the following message is issued:

   ```
   HEL -- CANNOT FIND DIRECTORY FILE
   ```

2. If the UFD is password-protected, the following prompt is issued:

   ```
   PASSWORD>
   ```

   The user must enter the correct password immediately following the prompt. (The user-entered password does not print on the terminal.)

3. If the user-entered password is in error, system access is denied and the following message is issued:

   ```
   HEL -- ILLEGAL PASSWORD
   ```

EXAMPLES:

1. The log-on procedure for a nonpassword-protected UIC is:

   ```
   MCR>HELLO [200,200]
   MCR>
   ```

2. The log-on procedure for a password-protected UIC is:

```
MCR>HELLO [200,200]
PASSWORD>
MCR>
```

NOTE

The password entered by the user is not printed on the terminal.

5.9 LOG COMMAND

FUNCTION:

The LOG command allows the user to type a comment on the terminal. This feature can be used to note information about the current system. Data typed following this command has no effect on the system.

FORMAT:

```
LOGΔ     comment-line
             or
;Δ       comment-line
             or
!Δ       comment-line
```

EXAMPLE:

```
MCR>LOG STANDARDS CHANGED BY 20 MG.  PER CUP.
                        or
MCR>;  STANDARDS CHANGED BY 20MG.  PER CUP.
                        or
MCR>!  STANDARDS CHANGED BY 20MG.  PER CUP.
```

NOTE

A log message can fill an entire line.

5.10 LOGICAL UNIT NUMBERS COMMAND (LUNS)

FUNCTION:

The Logical Unit Numbers (LUNS) command lists, on the user's terminal, the physical device units and corresponding logical unit numbers for an indicated task or tasks. This command is used when the operator wants to determine which physical devices a task requires.

FORMAT:

```
LUN[S]Δtaskname[,taskname,...]
```

EXAMPLE:

```
MCR>LUN ...SYS

**** ...SYS
TI  2
TT0 6
CL0 5
LP0 3
SY0 1,4
```

TECHNICAL NOTES:

The LUNS command lists:

1. I/O assignments taken from the task's disk image

2. Only the devices to which assignments have been made

Any run-time changes in LUN assignments made by an active task are not reflected in the listing.

5.11 MOUNT COMMAND (MOU)

FUNCTION:

The MOUNT command (MOU) allows the user to make a selected volume visible to the system. MOU creates the Volume Control Block (VCB) and declares that the volume is logically on-line for access by the File Control Primitives.

FORMAT:

The format listed below is the general format for the MOUNT command. MOUNT for multivolume magnetic tape is somewhat different in format, and is discussed in section 5.11.1.

```
MOU[NT]Δdev:volumelabel[keyword(s)]
```

where:

dev: is the device specifier for the device that contains the volume being mounted.

volumelabel
is the volume label of the volume being mounted. The label specified is compared to the label field of the volume's home block, to ensure that the physically mounted volume is the one to be logically mounted (made visible) by the system.

NOTE

Disk and DECtape volumes are allowed to contain null (all zeros) volume labels. In this case, the volume label need not be specified. The system assumes a null label.

/keyword(s)
are optional modifiers that allow the user to override the volume characteristics that were assigned to the volume when it was initialized. The following keyword options are provided.

NOTE

Notations listed to the left of keyword options equate to the following:

* Applies to disk and DECtape only

= Overrides the corresponding option specified at INITVOL

+ Applies to magnetic tape only

= /CHA=[characteristic word]

This option changes the device characteristic word. Possible parameters for this option are:

*+	FOR	-not an RSX-11D structured volume (FOREIGN). When this parameter is specified DCF is automatically assumed.
	ATCH	-device may be attached for exclusive use by one task.
	DCF	-device control functions permitted; removes restrictions.

NOTE

ATCH and DCF are legal for magnetic tape only when it is mounted as FOREIGN.

Example

/CHA=[FOR,ATCH]

* /UNL

This option specifies that the volume's index file is to be left unlocked, thus giving tasks write access to the index file.

+ /DENS=magnetic tape density

Possible parameters are:

800 for 800 BPI

1600 for 1600 BPI

Example

/DENS=800

/ACP=task name

This option allows the user to designate the task that is to be the file processor for the volume.

Example

/ACP=DSKFI

*= /EXT=default file extend increment in blocks

Example

/EXT=5

*= /FPRO=[default file protection]

This option allows the user to change the default file protection assigned to new files created on the volume. Each entry consists of from one to four letters which have the following meanings.

R - for READ access

W - for WRITE access

E - for EXTEND access

D - for DELETE access

The absence of one of these letters in an entry signifies the access right is denied to the user.

Protection code subparameters (system,owner,group,world) are positional; therefore, the location of entry in the parameter string defines the user to whom the codes apply.

Example

/FPRO=[RWED,RWED,RW,RW]

In this example, group and world are denied Extend and Delete access.

*= /LRU=number of directories to keep pre-accessed

Example

/LRU=3

+ /OVRFSID

This option is used to override the set identifier check. This option is required when processing magnetic tape with inconsistent file set identifiers.

+ /OVREXP

This option is used to override the expiration date check. Allows the user to overwrite unexpired magnetic tape files.

5.11.1 Format For Mounting Multivolume Magnetic Tape

The following format should be used to mount multivolume magnetic tape.

```
MOU[NT]ΔMT(n1[,n2,...,nn]):(label1[,label2,...,labeln]
          [/keyword(s)]
```

where:

n — is the logical tape drive number(s) (in the order of selection) representing the drive(s) to be dedicated to the processing of the multivolume set. If only one drive is being dedicated, the parentheses may be omitted.

label — is the volume label(s) (in order) that constitute the volume-set. Only the volume label for the first volume in the set need be specified. If further labels are specified, they will be used by the file processor to validate further volumes as they are requested. If no further labels are specified, it is up to the user to ensure that the correct volume is placed on the appropriate drive when requested.

NOTE

A separate unit number need not be specified for each volume in the set. The file processor processes volumes sequentially down the list of specified units, until the last unit is reached. If more volumes are to be processed, a mount request, for the next sequential volume to be mounted is issued for one of the units listed.

/keyword(s)
See keywords provided for the general class of Mount command.

TECHNICAL NOTES:

Because of the large number of options available with the MOU command, it may not be possible to enter the entire command string on a single line. For this reason, MOU has been modified to accept multiline commands. To accomplish this, the user must follow the procedure outlined below.

1. Initiate MOU as follows:

```
MCR>MOU
```

When MOU is ready to accept command lines, issue the following prompt:

```
MOU>
```

2. At this point, multiple lines can be typed in to MOU. Each line except the last must contain a hyphen (-) as the last character entered. When MOU recognizes the hyphen, it assumes that the command line is incomplete, and reissues the MOU> prompt for more parameters. When the last line is entered (no hyphen), MOU begins processing the command. See example number 3.

EXAMPLES:

1. MCR>MOUNT DK1:SYS004

In this example, a request is made to mount disk volume SYS004 on DK1:. None of the volume's attributes are to be overridden.

2. MCR>MOU MT(1,2):(RICK,SHEILA,LAURA,JEN)

In this example, a request is made to mount multivolume magnetic tape. Logical tape units 1 and 2 are reserved for processing. At the time the mount request is being processed, tape volume RICK must be physically mounted on logical tape unit 1; otherwise, the label check will fail. No label check is made for tape unit 2 until the file processor is ready to process the next volume (SHEILA).

3. MCR>MOU

```
MOU>DK1:ACCTSYS/WIN=10.-

MOU>/LRU=4/FPRO=[R,RWED,R,]
```

5.12 OPERATE COMMAND (OPR)

FUNCTION:

The OPERATE (OPR) command enables the terminal user to direct the output processing of the Multi-Output Despooler. See Section 2.6.

FORMAT:

```
OPR[ATE]Δdev:/switch[,dev:/switch,...]
```

where:

dev: is the device specifier for the Multi-Output Despooler device being controlled.

/switch is the command modifier that directs OPR operation. OPR switches are described in Table 5-2.

Table 5-2
OPR Switches

SWITCH	DESCRIPTION
/SP	Start printing, on the specified device, all files whose forms-type match the forms-type currently specified for this device.
/AB	Abort printing of the current file on the specified device. Select the next eligible file for processing. NOTE Although the aborted file is removed from the queue, it is not deleted. Therefore it is still available and can be requeued for printing at some later time.
/ST	Stop printing of the current file on the specified device. Printing of the file is stopped in place. Printing can be resumed by specifying one of the /RS switches; or it can be aborted by specifying the /AB switch.
/RS	Resume printing of the current file on the specified device.
/RS:T	Resume printing of the current file from the last encountered form feed. If no previous form feed was encountered, printing begins at the top of the file.
/RS:TOF	Restart printing of the current file from the beginning. NOTE All of the /RS options can be used without a previous stop option having been issued.

EXAMPLES:

1.
   ```
   MCR>OPR TT1:/ST
   ```

   In this example, the user has directed that the printing of files on TT1: be suspended.

2.
   ```
   MCR>OPR TT1:/RS:TOF
   ```

   In this example, the user has directed that the printing on TT1: be resumed, and that the file be reprinted from the beginning.

5.13 PARTITIONS COMMAND (PAR)

Refer to the SYS /PAR command in Section 5.18.

5.14 PASSWORD COMMAND (PWD)

FUNCTION:

The PASSWORD Command (PWD) allows the terminal user to change or create a password for his UFD on the system disk.

FORMAT:

```
PWDΔ[ufd]
```

where:

[ufd]   is the UFD receiving the new password.

When the above command is received by PWD, the following prompting message is issued.

```
PASSWORD>
```

The user now enters the desired password immediately following the right angle bracket (>). The newly entered password is not typed on the terminal.

NOTE

Only the UFD which corresponds to the UIC under which the user logged on can be changed.

EXAMPLE:

```
MCR>PWD [200,200]

PASSWORD>

MCR>
```

5.15 QUEUE COMMAND (QUE)

FUNCTION:

The QUEUE command (QUE) is the terminal user's interface to the RSX-11D queue manager task. The Queue Manager maintains a queue of elements. Each element defines a Files-11 file specification for a file that is to be printed on a line printer or a terminal.

The QUE command provides the user with the ability to perform the following functions:

1. Print queue files on a line printer or terminal.
2. List user-related elements in the queue manager task's queue.
3. List all elements in the queue manager task's queue.
4. Kill (delete) user-related element(s) in the queue manager task's queue.
5. Modify user-related element(s) in the queue manager task's queue.

NOTE

QUE works only with the Multi-Unit Spooler.

FORMAT:

The format of the QUE command varies depending on the function being performed. The format for each function is described in its respective section.

5.15.1 Queue A File For Printing

This function of the QUE command allows the user to queue a file or files for printing on a line printer or terminal.

FORMAT:

```
     QUEΔ[dev:=]infile-1[,infile-2,...,infile-n][/switch(es)]

                                   or

     QUE @indirect
```

where:

dev:= is the optional listing device specifier (default is LP:).

infile is a file specifier for the file(s) being queued for printing. (See Table 5-3 for a list of defaults in QUE file specifiers.)

NOTE

QUE allows wildcard specifiers.

/switch (es) is one or more of the QUE option switches. (See Table 5-4). QUE option switches are global; that is, they may be specified once for an entire list of input file specifiers. They may be appended to any of the file specifiers in a list. However, if a switch is specified repeatedly, the last value specified is used.

@indirect is an indirect command file specifier.

NOTE

The QUE function supports three levels of indirection.

Table 5-3
Defaults in QUE File Specifiers

Specifier	Default
dev:	SY:
[ufd]	For the first or only file specifier, the UFD which corresponds to the UIC under which the user logs on. For the second through n file specifiers, the UFD specified or defaulted for the previous file specifier.
filename	Must be specified.
.type	.LST
;ver	The latest version for the file.

Table 5-4
QUE Command Optional Switches

SWITCH	DESCRIPTION	DEFAULT
/PR:nnn	Set the priority of output selection to nnn (nnn is a decimal number ranging from 1 to 250).	/PR:50
/FO:n	Set the output forms type to n (n is a decimal number ranging from 0 to 6). Specific form types are defined by the system manager; for example, form type 0 should correspond to standard line printer forms (single ply, 11" x 15"); form type 2 could correspond to payroll check forms; form type 3 could correspond to standard billing forms.	/FO:0
/CO:nn	Set the number of copies to be printed to nn (nn is a decimal number ranging from 1 to 31).	/CO:1
/DE	Delete the input file after it has been spooled.	/-DE
/TE	Test for forms alignment. This switch is used to ensure that the desired forms are properly aligned in the receiving device. The priority entry is automatically set to 250, and the forms type entry is set to 7. This option, although documented in this chapter, is normally used by the system manager; it is also normally preceded by the following privileged MCR system command: MCR>OPR LP:/CH	None

EXAMPLES:

1. MCR>QUE TEST

   In this example, the latest version of file TEST.LST, residing in the user's directory file on SY:, is queued for printing. The following options are assigned to the queued entry.

- Printing device	LP:	(default)
- Priority	50	(default)
- Forms type	0	(default)
- Number of copies	1	(default)
- Delete	No	(default)

2. MCR>QUE TT1:=TEST.MAC/PR:250/CO:4/DE

   In this example, the latest version of file TEST.MAC, residing in the user's directory file on SY:, is queued for printing. The following options are assigned to the queued entry.

- Printing device	TT1:
- Priority	250
- Forms type	0 (default)
- Number of copies	4
- Delete	Yes

3. MCR>QUE [20,20]TEST,DK1:TEST.MAC,TEST.CMD/PR:250

   In this example, the latest version of the file TEST.LST, residing in the directory file [20,20] on SY:, and the latest version of files TEST.MAC and TEST.CMD, residing in directory file [20,20] on DK1:, are queued for printing. The following options are assigned to each file entry.

- Printing device	LP: (default)
- Priority	250
- Forms type	0 (default)
- Number of copies	1 (default)
- Delete	No (default)

4. `MCR>QUE [202,12]*.MAC`

   In this example, the latest version of all files of the type MAC residing in directory file [202,12] on SY:, are queued for printing. The following options are assigned to the queued entries.

- Printing device	LP: (default)
- Priority	50 (default)
- Forms type	0 (default)
- number of copies	1 (default)
- Delete	No (default)

5.15.2 List Elements In Queue (/LI)

The list function (/LI) of the QUE command allows the user to obtain a listing of all entries in the queue manager task's queue. Output from this function provides the user with pertinent information concerning the status of each entry in the queue, as well as the options specified for each entry. A sample queue listing, along with a description of the information it provides, is contained in Figure 5-1.

```
 1    2        3               4                5    6    7    8    9

DEV  ACT  ACCOUNT    FILE SPECIFICATION      SEQ  PRI   FO   CO    P

LP0   *   [200,200]  DP0:TEST.LST;6               100    0    4    *
TT5       [200,200]  DK1:RSXMAC.SML;17         4  240    0    1    *
TT1       [200,200]  DK0:TEST.MAC;40           2   20    0    1
TT5       [200,200]  DP0:QUE.CMD;3             3   50    0    1
```

1. DEV/A= Destination device.

2. ACT = Asterisk denotes an input file which is currently being output by the Despooler.

3. ACCOUNT = UIC of user who queued the request.

4. FILE SPECIFICATION = Input file specification

5. SEQ = Sequence number that uniquely identifies an inactive queued entry.

6. PRI = Priority of output selection.

7. FO = The forms types.

8. CO = Number of printed copies.

9. P = Delete/Preserve indicator. An asterisk indicates the file is to be preserved after output is complete.

Figure 5-1
Sample Output from QUE /LI Function

FORMAT:

```
     QUEΔ/LI
```

EXAMPLE:

```
     MCR>QUE /LI

DEV ACT  ACCOUNT    FILE SPECIFICATION     SEQ  PRI FO  CO P

LP0  *   [200,200]  DP0:TEST.LST;6               100  0   4      *
TT5      [200,200]  DK1:RSXMAC.SML;17        4   240  0   1      *
TT1      [200,200]  DK0:TEST.MAC;40          2    20  0   1
TT5      [200,200]  DP0:QUE.CMD;3            3    50  0   1
```

5.15.3 List All Elements in Queue (/AL)

The list all (/AL) performs the same function as list (LI) except that all entries in the queue are printed regardless of the user's UIC.

5.15.4 Kill (Delete) Elements in Queue (/KI)

The kill function (/KI) allows the user to delete elements from the queue manager task's queue. The user accomplishes this deletion by specifying the entry's queue sequence number as a parameter to the /KI function. Queue entry sequence numbers can be obtained by executing the QUE command with the /LI function specified.

NOTE

> Any attempt to delete a queued entry with an account number that is not the user's results in a privilege violation error message.

FORMAT:

```
     QUEΔ/KI:seq[:seq:...:seq]
```

where:

:seq is the sequence number for the queued entry being deleted.

Two formats for specifying sequence numbers are supported: a single defined sequence number (:seq) or a range of sequence numbers (:seql-seqn).

A single defined sequence numbers is specified by entering a colon (:) followed by the sequence number. When this format is used, the sequence number is interpreted as a command to delete a single queued entry.

A range of sequence numbers is specified by entering a colon (:) and the beginning sequence number, followed by a dash (-) and the ending sequence number. When this format is used, the sequence numbers are interpreted as a command to delete all entries whose consecutive sequence numbers start with the number specified to the left of the dash and end with the number specified to the right of the dash.

Any combination of formats can be used in the same /KI command line; however, the maximum number of individual combinations is ten. See Figure 5-2 for a sample execution of the /KI function.

```
DEV ACT  ACCOUNT    FILE SPECIFICATION     SEQ  PRI FO  CO P

LPO      [200,200]  DPO:RCVAST.MAC;45       25   55  2   1 *
LPO      [200,200]  DPO:SETBLD.CMD;3        26   55  2   1 *
LPO      [200,200]  DPO:PRTX.MAC;3          27   55  2   1 *
LPO      [200,200]  DPO:PRT.MSG;3           28   55  2   1 *
LPO      [200,200]  DPO:OPR.MAC;17          29   55  2   1 *   Listing of
LPO      [200,200]  DPO:OPRBLD.CMD;5        30   55  2   1 *   queued en-
LPO      [200,200]  DPO:SET.MAC;11           1   50  1   1 *   tires before
LPO      [200,200]  DPO:MAC12.CMD;4          3   50  1   1 *   execution of
LPO      [200,200]  DPO:SPRODT.CMD;2         4   50  1   1 *   /KI function
LPO      [200,200]  DPO:INLINE.MAC;51        5   50  1   1 *
LPO      [200,200]  DPO:FORMAT.MAC;23        7   50  1   1 *
LPO      [200,200]  DPO:SETUP.MAC;15         8   50  1   1 *
LPO      [200,200]  DPO:IOAST.MAC;25        13   50  1   1 *
LPO      [200,200]  DPO:SPR2.MAC;60         14   50  1   1 *
LPO      [200,200]  DPO:SPRBLD.CMD;10       15   50  1   1 *
LPO      [200,200]  DPO:TSTBLD.CMD;6        16   50  1   1 *
LPO      [200,200]  DPO:TEST.MAC;22         17   50  1   1 *

QUE>/KI:27-30:1:3-5:7-8:13
QUE>/LI

DEV ACT  ACCOUNT    FILE SPECIFICATION     SEQ  PRI FO  CO P

LPO      [200,200]  DPO:RCVAST.MAC;45       25   55  2   1 *   Listing of
LPO      [200,200]  DPO:SETBLD.CMD;3        26   55  2   1 *   queued en-
LPO      [200,200]  DPO:SPR2.MAC;60         14   50  1   1 *   tries after
LPO      [200,200]  DPO:SPRBLD.CMD;10       15   50  1   1 *   execution of
LPO      [200,200]  DPO:TSTBLD.CMD;6        16   50  1   1 *   /KI function
LPO      [200,200]  DPO:TEST.MAC;22         17   50  1   1 *
```

Figure 5-2
Sample Execution of /KI Function

EXAMPLES:

1. MCR>QUE /KI:2:6:10:12

   In this example, the queued entries whose sequence numbers are 2, 6, 10, and 12 are deleted from the queue manager task's queue.

2. MCR>QUE /KI:2:6-12:14

   In this example, the queued entry 2, the entries 6 through 12, and entry 14 are deleted from the queue manager task's queue.

5.15.5 Modify A Queue Entry (/MO)

The modify function (/MO) allows the user to change the option(s) originally specified for a queued entry or entries.

The following queued entry fields can be modified by the /MO function.

listing device
priority
forms type
number of copies
delete option

FORMAT:

```
QUEΔ[dev:=]/MO:seq[:seq...:seq][/switch(es)]
```

where:

dev: is the new listing device specifier. If dev: is specified, it must be followed by an equal sign (=).

:seq is the sequence number of the queue entry being modified. Sequence numbers can be entered individually or in a range of sequence numbers. See Section 5.15.3.

NOTE

When the /MO function is executed, the specified queue entry is deleted and replaced with a modified version of the original entry. Therefore, the original sequence number disappears and is replaced with a number one greater than the highest number in the queue at the time the modify function is executed. Figures 5-3 and 5-4 contain sample executions of the /MO function.

/switch(es)

is the option modification switch(es). Modification switches are described in Table 5-5.

Modification switches can be appended only to the last sequence number in the command line.

NOTE

If the /DE switch is not specified, /-DE is assumed.

Table 5-5
/MO Option Switches

SWITCH	DESCRIPTION	DEFAULT
/PR:nnn	Change the priority of the queued entry to nnn (where nnn is a decimal number ranging from 1 to 250).	none
/FO:n	Change the form type to n (where n is a decimal number ranging from 1 to 6).	none
/CO:nn	Change the number of copies to nn (where nn is a decimal number ranging from 1 to 31).	none
/DE	Change the delete/preserve indicator to delete.	/-DE

```
     QUE /LI

DEV ACT   ACCOUNT     FILE SPECIFICATION      SEQ   PRI  FO   CO P

LP0       [200,200]   DP0:BPR.ODL;7             1    50   0    1 *
LP0       [200,200]   DP0:RICK.MAC;14           2    50   0    1 *
LP0       [200,200]   DP0:BPRBLD.CMD;20         3    50   0    1 *
LP0       [200,200]   DP0:INIT.MAC;37           4    50   0    1 *   Listing of
LP0       [200,200]   DP0:TEMP.SAV;2            5    50   0    1 *   queued en-
LP0       [200,200]   DP0:HANK.BIS;1            6    50   0    1 *   tries prior
LP0       [200,200]   DP0:BPRMAC.MAC;5          7    50   0    1 *   to execu-
LP0       [200,200]   DP0:UTIL.MAC;3            8    50   0    1 *   tion of /MO
LP0       [200,200]   DP0:ERRMES.MAC;4          9    50   0    1 *   function
LP0       [200,200]   DP0:MAIN.MAC;32          10    50   0    1 *
LP0       [200,200]   DP0:MAIN.MAC:32          11    50   0    1 *
LP0       [200,200]   LP0:CLOSE.MAC;12         12    50   0    1 *

MCR>QUE TT1:=/MO:2:6:9-12/FO:2/PR:55/DE
MCR>QUE /LI

DEV ACT   ACCOUNT     FILE SPECIFICATION      SEQ   PRI  FO   CO P

TT1       [200,200]   DP0:RICK.MAC;14          13    55   2    1
TT1       [200,200]   DP0:HANK.BIS;1           14    55   2    1
TT1       [200,200]   DP0:ERRMES.MAC;4         15    55   2    1     Listing of
TT1       [200,200]   DP0:LOGGER.MAC;3         16    55   2    1     queued en-
TT1       [200,200]   DP0:MAIN.MAC;32          17    55   2    1     tires after
TT1       [200,200]   DP0:CLOSE.MAC;12         18    55   2    1     execution
LP0       [200,200]   DP0:BPR.ODL;7             1    50   0    1 *   of /MO
LP0       [200,200]   DP0:BPRBLD.CMD;20         3    50   0    1 *   function
LP0       [200,200]   DP0:INIT.MAC;37           4    50   0    1 *
LP0       [200,200]   FP0:TEMP.SAV;2            5    50   0    1 *
LP0       [200,200]   DP0:BPRMAC.MAC;5          7    50   0    1 *
LP0       [200,200]   DP0:UTIL.MAC;3            8    50   0    1 *

MCR>
```

NOTE

Note that the sequence numbers for the entries being modified were deleted and replaced with modified entries having new sequence numbers.

Figure 5-3
Sample Execution of /MO Function
(with Option Switches Specified)

```
QUE /LI

DEV ACT  ACCOUNT    FILE SPECIFICATION    SEQ  PRI FO  CO P

TT1      [200,200]  DP0:RICK.MAC;14        13   55  2   1
TT1      [200,200]  DP0:HANK.BIS;1         14   55  2   1
TT1      [200,200]  DP0:ERRMES.MAC;4       15   55  2   1
TT1      [200,200]  DP0:LOGGER.MAC;3       16   55  2   1      Listing of
TT1      [200,200]  DP0:MAIN.MAC;32        17   55  2   1      queued en-
TT1      [200,200]  DP0:CLOSE.MAC;12       18   55  2   1      tries prior
LP0      [200,200]  DP0:BPR.ODL;7           1   50  0   1 *    to execu-
LP0      [200,200]  DP0:BPRBLD.CMD;20       3   50  0   1 *    tion of /MO
LP0      [200,200]  DP0:INIT.MAC;37         4   50  0   1 *    function
LP0      [200,200]  DP0:TEMP.SAV;2          5   50  0   1 *
LP0      [200,200]  DP0:BPRMAC.MAC;5        7   50  0   1 *
LP0      [200,200]  DP0:UTIL.MAC;3          8   50  0   1 *

MCR>QUE /MO:13-18
MCR>QUE /LI

DEV ACT  ACCOUNT    FILE SPECIFICATION    SEQ  PRI FO  CO P

TT1      [200,200]  DP0:RICK.MAC;14        19   55  2   1 *
TT1      [200,200]  LP0:HANK.BIS;1         20   55  2   1 *
TT1      [200,200]  DP0:ERRMES.MAC;4       21   55  2   1 *
TT1      [200,200]  DP0:LOGGER.MAC;3       22   55  2   1 *    Listing of
TT1      [200,200]  LP0:MAIN.MAC;32        23   55  2   1 *    queued en-
TT1      [200,200]  DP0:CLOSE.MAC;12       24   55  2   1 *    tries after
LP0      [200,200]  DP0:BPR.ODL;7           1   50  0   1 *    execution
LP0      [200,200]  DP0:BPRBLD.CMD;20       3   50  0   1 *    of /MO
LP0      [200,200]  DP0:INIT.MAC;37         4   50  0   1 *    function
LP0      [200,200]  DP0:TEMP.SAV;2          5   50  0   1 *
LP0      [200,200]  DP0:DPRMAC.MAC;5        7   50  0   1 *
LP0      [200,200]  DP0:UTIL.MAC;3          8   50  0   1 *
```

NOTE

Note that the sequence numbers, as in Figure 5-3, were deleted and replaced with new ones. Note also, that the delete option for the specified entries were automatically changed to preserve (/-DE)

Figure 5-4
Sample Execution of /MO Function
(Default)

5.16 RESUME COMMAND (RES)

FUNCTION:

The RESUME command allows the user to continue execution of a previously suspended task or tasks. Suspended tasks can be resumed only from the terminal from which they were initiated.

FORMAT:

```
RES[UME]Δtaskname[/TI=dev][,taskname-1[/TI=dev],...]
```

where:

taskname is the name of the task being resumed.

/TI=dev is an optional switch (appended to the taskname) which allows the user to resume any task in the system by specifying its TI (dev is the device mnemonic for the terminal corresponding to the TI under which the task is run).

EXAMPLES:

1. `MCR>RESUME XKE`

2. `MCR>RESUME XKE, NK111,240Z`

In the examples above the requests are to:

1. RESUME Task XKE.

2. RESUME Task XKE, Task NK111, and Task 240Z.

5.17 RUN COMMAND (RUN)

FUNCTION:

The RUN command allows the user to initiate task images (output from the RSX-11D Task Builder) that have not been installed. RUN causes a selected task image to be installed temporarily, loaded and executed and removed upon termination.

The task attributes are those specified at task build time. The only exception is the task name, which is converted to the first six characters of the filename.

The task runs under the UIC that the terminal is logged in under.

FORMAT:

```
RUNΔfilename[/TIM=NX] $
    or
RUNΔ$filename $
```

where:

filename is the file specifier for the task image. Table 5-6 contains a list of defaults in task image file specifiers.

$filename instructs the system that the requested task image resides in a predetermined UFD. This UFD is defined by the system manager at system generation time.

Table 5-6
Defaults in Task Image File Specifiers

SPECIFIER	DEFAULT
dev:	SY:
[ufd]	UFD that corresponds to the UIC under which the user
logged on	
filename	Must be specified
.type	.TSK
;ver	The latest version for the file

OPTION

/TIM=nx

n is a decimal number indicating the number of time units (x) of central processor time that the task is allowed.

x indicates the type of time unit and can be one of the following.

H = hour
M = minutes
S = seconds
T = ticks

The amount of time specified must be no greater than 24 hours. For example, 24H, 1440M, and 86400S are the maximum amounts for hours, minutes, and seconds, respectively.

If /TIM is not included and accounting is specified for the task, the maximum amount of processor time allowed is five minutes. If the task is still running after five minutes of processor time has been used and /TIM was not specified, the system aborts the task.

EXAMPLE:

```
MCR>RUN DK1:RICK.TSK;3($)
```

In this example, task image RICK.TSK;3 is installed, loaded and executed, and removed upon termination. RICK.TSK;3 resides in the user's directory file on DK1:.

5.18 SYSTEM STATUS COMMAND (SYS)

The system status command (SYS) lists on the terminal the names of tasks associated with any one or all of the following system lists and queues.

- Active task list (ATL)
- Memory required list (MRL)
- Clock queue (CKQ)
- Global common directory (GCD)
- Checkpointable task list (CTL)
- I/O request queue (IRQ)
- Fixed task list (FTL)
- System task directory (STD)
- Physical unit directory (PUD)
- Task partition directory (TPD)
- SEND/RECEIVE queue (SRQ)
- Asynchronous system trap queue (ASQ)

For each list, SYS prints the list name and the names of all tasks that have entries or are entered in that list. In the case of the SRQ and ASQ, SYS prints the queue name and the names of all tasks that have nodes in that queue. The function and content of system lists and queues are described in the RSX-11D Executive Reference Manual. SYS can also print the names of shared global areas (SGAs) to which tasks in the system task directory (STD) are bound and provide the following information:

1. List all installed tasks.
2. List the device known to the system.
3. List the partition defined at system generation.

Each list contains information related to the specific category.

FORMAT

```
    SYS [/switch[:opt]]
```

where:

switch = any one of the valid switches listed in Table 5-7. /BRF is the default switch.

opt = a partition name if /MRL or /CTL is specified, or a device name and unit number if /IRQ is specified. No colon follows the unit number.

Table 5-7
SYS Switches

SWITCH	FUNCTION
/ATL	Lists the names of tasks in the active task list.
/MRL[:opt]	Lists the names of tasks in the memory required list. If the output from SYS is to be restricted to a particular partition, use the opt parameter; for example, /MRL:GEN indicates that the names of tasks installed in the GEN partition and contained in the MRL are to be printed.
/CKQ	Lists the names of tasks in the clock queue.
/GCD	Lists the names of libraries and common areas in the global common directory.
/IRQ[:opt]	Lists the names of tasks with entries in the I/O request queues. If the IRQ for a particular device is desired, the opt parameter must be included to designate the device; for example, /IRQ:DK0 causes the IRQ associated with the DK handler to be listed.
/CTL[:opt]	Lists the names of tasks in the checkpointable task list. If the output from SYS is to be restricted to a particular partition, use the opt parameter; for example, /CTL:GEN indicates that the names of tasks running in the GEN partition and contained in the CTL are to be printed.
/FTL	Lists the names of tasks in the fixed task list.
/SRQ	Lists the names of tasks that have SENDS queued for them.
/ASQ	Lists the names of tasks in the ATL that have ASTs queued for them.
/SGA	Lists the names of all tasks in the STD that are bound to one or more sharable global area and the shared global areas to which each is bound.
/BRF	Lists the names of all tasks in the ATL, MRL and CKQ. /BRF is used by default if no switches are included in the SYS command.
/FUL	Lists all of the above except that which is produced by using /SGA.
/TAS	Lists the names of and the following information for installed tasks: version number, default partition and priority, task size, and fixed and multiuser indicators. See Figure 5-5.

Table 5-7 (Cont.)
SYS Switches

SWITCH	FUNCTION
/DEV	Lists the name and PUD address of each symbolic device known to the system. On the listing, devices for which a device handler is resident are indicated by two asterisks (**). A redirected device is followed by the mnemonic of the device to which it was redirected. Spooled and mounted (MOU) devices are indicated by the words SPOOLED and MOUNTED, respectively. The number following a spooled device is its current forms type.
/PAR	List the following description of each memory partition in the system: name, base address (octal), size (octal), and partition type. Partition type can be: U (user-controlled), S (system-controlled, priority-oriented), T (system-controlled, time-scheduled).
/COM	Lists the following description of each installed shareable global area: name, base address (octal), size, UIC, access, position independent or blank, creation date. NOTE Each scan performed by SYS inhibits task switching for a significant period of time. Therefore, SYS /FUL should not be issued frequently.

```
  1       2       3       4    5     6       7          8
F11ACP   VO205   GEN     245 021500 DP 0-00000005357 FIXED
```

1 = Task name

2 = Version number of task

3 = Default partition name

4 = Priority of task

5 = Size of task (octal)

6 = Device identification

7 = Disk address (octal)

8 = Contains the word "FIXED" if the task is fixed in memory, or CHECKPOINTED" if the task is checkpointed.

Figure 5-5
Sample TASK LIST Listing and Descriptions of Listed Items

EXAMPLES

1. MCR>SYS /ATL

```
ATL:

DP....   TT....   DK....   LP....   DS....   DT....   ...MCR   ...SYS   MO....

SPR2..   DLLEX
```

2. MCR>SYS /MRL

```
MRL:

...MAC   ...TKB   CRF...
```

3. MCR>SYS /CKQ

```
CKQ:

NO ENTRIES
```

4. MCR>SYS /IRQ

```
IRQ:


TT0:

...PIP   ...FLX

CI0:

NO ENTRIES

CO0:

NO ENTRIES

CL0:

NO ENTRIES

SY1:

NO ENTRIES

IP0:

NO ENTRIES
```

```
5.  MCR>SYS /CTL

    CTL:

    ...SYS  ...PIP  DLLEX

6.  MCR>SYS /CTL:GEN

    CTL:

    ...SYS  DLLEX

7.  MCR>SYS /FTL

    FTL:

    F11ACP

8.  MCR>SYS /SRQ

    SRQ:

    NO ENTRIES

9.  NCR>SYS /ASQ

    ASQ:

    NO ENTRIES
```

```
10.  MCR>SYS /SGA
     ACCLOG SYSRES
     ERRLOG SYSRES
     MO.... SYSRES
     SPR... SYSRES
     SPR2.. SYSRES
     ...ACT PROOT.
     ...BOO SYSRES
     ...CNF SYSRES
     ...DMP PROOT. SYSRES
     ...EDI PROOT. SYSRES
     ...FLX PROOT. SYSRES
     ...FOR SYSRES
     ...INS SYSRES
     ...LBR PROOT. SYSRES
     ...MAC PROOT. SYSRES
     ...MCR PROOT.
     ...PIP PROOT. SYSRES
     ...QUE PROOT. SYSRES
     ...REM SYSRES
     ...RJE SYSRES
     ...SYS SYSRES
     ...TKB SYSRES
     ...VFY PROOT. SYSRES
```

NOTE

PROOT indicates that the task is multiuser and has a read-only segment in its root. This is the segment of the task that is shared among all versions of the task.

```
11.  MCR>SYS /BRF

     ATL:

     DP....  ...RJE  TT....  DK....  LP....  DS....  DT....  ...MCR  ...SYS

     MO....  ...EDI

     MRL:

     NO ENTRIES

     CKQ:

     NO ENTRIES
```

```
12.  MCR>SYS /FUL

     ATL:

     DP....  ...RJE  TT....  DK....  LP....  DS....  DT....  ...MCR  ...SYS

     MO....  ...EDI

     MRL:

     NO ENTRIES

     CKQ:

     NO ENTRIES

     FTL:

     F11ACP

     GCD:

     SYSRES

     CTL:

     ...SYS  ...MAC

     SRQ:

     NO ENTRIES

     ASQ:

     NO ENTRIES

     IRQ:

     TT:

     NO ENTRIES

     CI:

     NO ENTRIES

     DK:

     NO ENTRIES

     DP:

     NO ENTRIES
```

```
13.  MCR>SYS /TAS
       ACCLOG  0203     GEN      235 003200 DP 0-00000050241
       CRF...  00       GEN      050 037300 DP 0-00000045654 MULTI-USER
       DK....  06A      GEN      248 003400 DP 0-00000046654
       DLLEX   03       GEN      050 027200 DP 0-00000226525
       DP....  06       SYDISK   248 003500 DP 0-00000046775
       DS....  07       GEN      248 003400 DP 0-00000047145
       DT....  V005B    GEN      248 004300 DP 0-00000047661
       ERRLOG  0321     GEN      200 011000 DP 0-00000053135
       F11ACP  D0225    GEN      220 022700 DP 0-00000054167 FIXED
       F11MSG  V0007    GEN      200 004700 DP 0-00000051505
       LP....  05       GEN      248 003200 DP 0-00000047753
       MO....  V003F    GEN      100 003600 DP 0-00000053076
       MT....  06       GEN      248 005700 DP 0-00000050144
       POOL    1A       GEN      050 001600 DP 0-00000021630 MULTI-USER
       SPR...  V006A    GEN      100 010400 DP 0-00000017347
       SPR2..  V006A    GEN      090 011400 DP 0-00000021154
       TT....  0602     TTY      248 014100 DP 0-00000054064
       .TKTN.  V05      GEN      249 003000 DP 0-00000032265
       ...ABO  V19      MCR      221 003300 DP 0-00000021166 MULTI-USER
       ...ACT  V01      GEN      221 006300 DP 0-00000024762 MULTI-USER
       ...ALT  V05      MCR      221 003300 DP 0-00000022446 MULTI-USER
        ..BOO  V005C    GEN      098 015000 DP 0-00000006432
       ...BYE  V01      MCR      221 000600 DP 0-00000015544 MULTI-USER
       ...CDA  0302     GEN      065 057600 DP 0-00000077035
       ...CNF  00003    GEN      250 003400 DP 0-00000220017
       ...DEV  V15      GEN      221 001600 DP 0-00000016565 MULTI-USER
       ...DLL  01       GEN      050 023000 DP 0-00000173415
       ...DMO  V0011    GEN      075 005600 DP 0-00000051723
       ...DMP  D0004    GEN      055 030500 DP 0-00000015511 MULTI-USER
       ...EDI  06       GEN      060 047500 DP 0-00000057131 MULTI-USER
       ...FIX  V19      MCR      221 003700 DP 0-00000025707 MULTI-USER
       ...FLX  M04      GEN      055 061400 DP 0-00000030017 MULTI-USER
       ...FOR  DY01     GEN      050 040300 DP 0-00000104543 MULTI-USER
       ...HEL  V01      GEN      221 024000 DP 0-00000046313 MULTI-USER
       ...INI  V0006    GEN      075 023200 DP 0-00000054104 MULTI-USER
       ...INS  V005D    GEN      150 020500 DP 0-00000046341 MULTI-USER
       ...LBR  X03.4    GEN      050 064300 DP 0-00000102327 MULTI-USER
       ...LOA  V14      MCR      221 003600 DP 0-00000025767 MULTI-USER
       ...LUN  V10      GEN      221 004300 DP 0-00000032330 MULTI-USER
       ...MAC  014      GEN      050 100500 DP 0-00000030572 MULTI-USER
       ...MCR  V41      MCR      230 002400 DP 0-00000030170 MULTI-USER
       ...MOU  V0417    GEN      075 011200 DP 0-00000053014
       ...OPE  V33      GEN      221 005300 DP 0-00000045717 MULTI-USER
       ...OPR  V006A    GEN      221 011500 DP 0-00000030675
       ...PIP  D1320    GEN      055 030500 DP 0-00000012715 MULTI-USER
       ...PWD  V01      GEN      221 006500 DP 0-00000045775 MULTI-USER
       ...QUE  V006A    GEN      100 022600 DP 0-00000030137 MULTI-USER
       ...REA  V23      GEN      221 004200 DP 0-00000045727 MULTI-USER
       ...RED  V25      MCR      221 002700 DP 0-00000030773 MULTI-USER
       ...REM  V09      GEN      150 014300 DP 0-00000046236 MULTI-USER
       ...RES  V18      MCR      221 003200 DP 0-00000046052 MULTI-USER
       ...RJE  00124    GEN      250 037400 DP 0-00000227533
       ...RJP  00124    GEN      250 037300 DP 0-00000174417
       ...RUN  V004A    GEN      221 005000 DP 0-00000047442 MULTI-USER
       ...SAV  0601     GEN      221 006500 DP 0-00000006451
       ...SET  V03      GEN      221 004600 DP 0-00000046060 MULTI-USER
       ...SYS  1A       GEN      150 007300 DP 0-00000046622 MULTI-USER
       ...TIM  V15      GEN      221 002300 DP 0-00000032051 MULTI-USER
       ...TKB  19       GEN      050 075700 DP 0-00000062374 MULTI-USER
       ...UFD  V0415    GEN      075 006400 DP 0-00000051752 MULTI-USER
```

```
        ...UNF  V16     MCR     221 002700 DP 0-00000045307 MULTI-USER
        ...UNL  V11     MCR     221 002700 DP 0-00000046006 MULTI-USER
        ...VFY  D0520   GEN     055 037300 DP 0-00000021203 MULTI-USER
        ...WHO  V01     GEN     221 001500 DP 0-00000021174 MULTI-USER

14.   MCR>SYS /DEV
        TT0  141500  **
        CI0  141562  TT0
        CO0  141644  TT0
        CL0  141726  TT0
        SY1  142010  DP0
        IP0  142072
        NT0  142154
        SP0  142236  DS0
        WK0  142320  DS0
        LP1  142402
        XX0  142464
        BP0  142546
        MO0  142630  **
        DT3  142712  **
        DT2  142774  **
        DT1  143056  ** MOUNTED
        DT0  143140  **
        CT1  143222
        CT0  143304
        CR0  143366
        LP0  143450  ** SPOOLED  0
        TT20 143532
        TT17 143614  **
        TT16 143676  **
        TT15 143760  **
        TT14 144042  **
        TT13 144124  **
        TT12 144206  **
        TT11 144270  **
        TT10 144352  **
        TT7  144434  **
        TT6  144516  **
        TT5  144600  **
        TT4  144662  **
        TT3  144744  **
        TT2  145026  **
        TT1  145110  **
        MT1  145172
        MT0  145254
        DS0  145336  ** MOUNTED
        DP1  145420  **
        DP0  145502  ** MOUNTED
        DK3  145564  **
        DK2  145646  **
        DK1  145730  **
        DK0  146012  **
        SY0  146074  DP0
```

```
15.  MCR>SYS /PAR
       SYDISK 113200  003500 U
       MCR    116700  015000 S
       GEN    133700  601100 S
       TTY    735000  014400 U

16.  MCR>SYS /COM
       SYSRES 174600  016400   1,1   RO LIB PI 05/06/75

17.  MCR>SYS /ATL

     ATL:

     DP....  ...RJE  TT....  DK....  LP....  DS....  DT....  ...MCR  ...SYS

     MO....

18.  MCR>SYS /MRL

     MRL:

     NO ENTRIES
```

5.19 TASK LIST COMMAND (TAS)

Refer to the SYS /TAS command in Section 5.18.

5.20 TIME COMMAND

FUNCTION:

The TIME command (TIM) allows the user to list the time and date on the terminal.

FORMAT:

```
TIME
```

EXAMPLE:

```
MCR>TIME

11/14/74 13:45:00
```

5.21 TERMINAL STATUS COMMAND (WHO)

The terminal status command (WHO) indicates which terminals are in use, provides the UIC of the user, and indicates whether the user terminal is privileged or slave. It also indicates those terminals that were defined during system generation but which are not currently in use.

FORMAT

```
     WHO
```

EXAMPLE

```
     MCR>WHO

     TT0 [1,1]PRIVILEGED
     TT1 [101,30]
```

CHAPTER 6

PRIVILEGED USER COMMANDS

Privileged user commands that appear to duplicate nonprivileged commands are actually nonprivileged commands with enhanced capabilities for the privileged user.

A list of the privileged user commands follows.

- ABORT
- ALTER
- BAD
- BOOT
- CANCEL
- DISABLE
- DISMOUNT
- ENABLE
- FIX
- INITVOL
- INSTALL
- LOAD
- MOUNT
- MEMORY
- OPEN
- OPERATE
- PASSWORD
- REASSIGN
- REDIRECT
- REMOVE
- RUN
- SAVE
- SET
- TIME
- UFD
- UNFIX
- UNLOAD

Commands are presented in alphabetic order.

NOTE

> Please reread the Preface for explanation of the use of characters (/ [ ] etc.) that indicate command format conventions.

6.1 ABORT COMMAND (ABO)

FUNCTION:

The Abort command (ABO) allows the privileged user to terminate the execution of a specified task.

FORMAT:

```
ABO[RT]Δtaskname[/TI=dev]
```

where:

taskname — is the name of the task being aborted.

/TI=dev — is an optional switch (appended to the taskname) which allows the user to abort any task in the system, by specifying its TI. (dev) is the device mnemonic for the terminal corresponding to the TI under which the specified task is running.

NOTE

If the requesting task was initiated at the requesting terminal, or if the designated task is not a multiuser task, the /TI switch is not required.

EXAMPLES:

1. `MCR>ABO SCAN`

2. `MCR>ABO SCAN/TI=TT15`

6.2 ALTER PRIORITY COMMAND (ALT)

FUNCTION:

The Alter priority command (ALT) allows the privileged user to alter the priority of an active task or tasks.

FORMAT:

```
ALT[ER]Δtaskname[/TI=dev]/PRI=nnn[,taskname[/TI=dev]/PRI=nnn,...]
```

where:

taskname — is the name of the task being altered.

/TI=dev — is an optional switch (appended to the taskname) which allows the user to alter the priority of any task in the system by specifying its TI. (dev) is the device mnemonic for the terminal corresponding to the TI under which the task runs.

NOTE

If the requesting task was initiated at the requesting terminal, or if the designated task is not a multiuser task the /TI switch is not required.

/PRI=nnn — is the switch used to specify the new task priority. nnn is a decimal number ranging from 1 to 250.

EXAMPLE:

```
MCR>ALT SCAN/PRI=75
```

6.3 BAD BLOCKS COMMAND (BAD)

FUNCTION:

WARNING

> Any data stored on disk pack will be destroyed when the BAD command is executed.

The BAD BLOCKS command (BAD) is used to locate any unusable blocks on a disk pack. In addition the BAD command records their locations in the last good block on the disk prior to the initialization of the disk with the INITVOL command. The BAD command may be executed simultaneously with other RSX-11D tasks which do not require the disk under test. The detection of bad blocks is desirable for any type of disk pack, especially for multisurface disk packs, which, even when new, may contain several bad blocks.

The BAD command detects bad blocks by writing a special worst case pattern in each block of the disk. Then, each block is read to verify the data. As bad blocks are detected, the locations and the number of bad blocks are collected. When all blocks have been tested, the collected data is written into the last good block on the disk.

After testing, the following message is printed and BAD exits to MCR:

```
BAD--TOTAL NO.  OF BAD BLOCKS = nnnnn
```

where nnnnn is the total number (decimal)

of bad blocks found in all sets.*

* There can be up to 102 sets of bad blocks. Each set can describe from 1 to 256 contiguous bad blocks. If this maximum of 102 sets is exceeded the following error message is printed:

```
BAD--BAD BLOCK FILE OVERFLOW
```

When INITVOL is executed using the /BAD = [AUTO] option, blocks identified as bad in the bad block data are allocated so they cannot be used for data storage. At the same time, the last good block on the disk pack (which includes the bad block data) is allocated, saving the data for subsequent initialization (e.g., in case a mistake is made in entering the INITVOL information).

FORMAT:

The BAD command format is:

```
BADΔdev:
```

where:

dev: is the device specifier for the disk under test

EXAMPLE:

```
MCR>BAD DK1:

BAD--TOTAL NO.  OF BAD BLOCKS=42
```

In the above example, disk DK1 is searched for bad blocks; 42 bad blocks are detected.

ERROR MESSAGES:

Errors detected during BAD processing are reported to the user via error messages listed on the user's terminal. These error messages are detailed in Section A.2.

TECHNICAL NOTES

1. The disk must be off-line (i.e., not mounted).

2. The disk must be formatted.

3. The disk device must be a valid Files-11 supported device.

4. The user will be warned not to use the disk pack as a system disk if block 0 is found to be bad (block 0 is used to store the bootstrap record).

5. BAD first writes the worst case pattern 26 (decimal) blocks at a time. If an error occurs, it then tests each individual block of the 26. It is possible for individual block testing to be successful, thereby indicating hardware malfunctions on long data transfers. As long as the test of individual blocks is successful, BAD does not report errors.

   The user may notice this situation if error logging is running while BAD is testing the disk. When BAD reports no bad blocks and error logging reports errors, the problem is probably in the disk controller rather than the particular disk volume.

6.4 BOOTSTRAP COMMAND (BOO)

FUNCTION:

The BOOTSTRAP command (BOO) allows the privileged user to perform either of the following mutually exclusive operations.

1. Stop the present operating system and bootstrap that system or another system from any system device.

2. Copy a bootstrap block on to block 0 of the specified device from a specified RSX-11D system image file on the same device.

FORMAT:

The BOOT command can be specified in four different formats. Each format is discussed separately.

Format 1

```
        BOO[T]
```

When this format is used, BOOT issues the following prompt:

```
        BOO>
```

The user must now enter another carriage return.

When executed in this format the BOOT command causes the latest version of the file RSX.SAV (residing in the default directory file of the system disk) to be bootstrapped into memory.

Format 2

```
        BOO[T]Δfilespec
```

where:

filespec — is the file specifier for the file that contains, as its first record, the bootstrap routine to be used in re-bootstrapping the operating system. Table 6-1 contains a list of default values for BOOT file specifiers.

When this format is used, the BOOT command causes the bootstrap routine to be loaded from the file specified by "filespec". This bootstrap routine then loads the remainder of the file into memory thus starting up the new RSX-11D system.

Format 3

```
BOO[T]Δ/WB
```

where:

/WB is the write bootstrap switch.

When this format is used, a bootstrap routine is written on block 0 of the system device. The bootstrap routine is copied from virtual block one of the latest version of file RSX.SAV residing in the default directory file on the system device.

Format 4

```
BOO[T]Δfilespec/WB
```

where:

filespec — is the file specifier for the file that contains, as its first virtual block, the bootstrap routine to be used. Table 6-1 contains a list of default values for BOOT file specifiers.

When this format is used, the bootstrap routine which is contained as the first virtual block, in the specified file, is copied out to block 0 of the specified device.

Table 6-1
Defaults In BOO File Specifiers

SPECIFIER	DEFAULT
dev:	SY:
ufd	The UFD which corresponds with the UIC under which the user logged on
filename	RSX
.type	.SAV
;ver	Latest version

TECHNICAL NOTES

1. All devices, except the one from which the bootstrap operation is to be accomplished, should be dismounted.

2. When a re-bootstrap is to take place, the currently running system should be quiescent (i.e., no tasks running)

3. The device from which a task is installed is recorded in that task's STD entry. Therefore, after bootstrapping a new system, any installed tasks must be present at the same place on their original devices; e.g., a system pack cannot be moved from DK0 to DK1 and then bootstrapped using the BOOT command.

4. Only as much memory as was specified at SYSGEN time will be loaded during the bootstrap operation.

5. The file specified to BOO must have been created by SYSGEN, and must not have been moved or copied after creation.

EXAMPLES:

1.
```
MCR>BOO
```

   In this example, the system is re-bootstrapped using the latest version of the system image file RSX.SAV which resides in the default directory file on device SY:).

2.
```
MCR>BOO DK1:[10,10]SYS.SAV
```

   In this example, the system is re-bootstrapped using the latest version of the system image file SYS.SAV which resides in directory file [10,10] on device DK1:).

3.
```
MCR>BOO /WB
```

   In this example, a bootstrap routine is copied from virtual block one of the system image file RSX.SAV (which resides in the default directory file on SY:) to block 0 of device SY:.

4.
```
MCR>BOO DK1:[10,10]SYS.SAV/WB
```

   In this example, a bootstrap routine is copied, from virtual block one of the system image file SYS.SAV (which resides in directory file [10,10] on DK1:) to block 0 of device DK1:.

6.5 CANCEL COMMAND (CAN)

FUNCTION:

The CANCEL command (CAN) allows the privileged user to cancel periodic rescheduling of tasks.

FORMAT:

```
CAN[CEL]Δtaskname[/TI=dev][,taskname[/TI=dev],...]
```

where:

taskname — is the name of the task being canceled.

/TI=dev — is an optional switch (appended to the task name) which allows the user to alter the priority of any task in the system by specifying its TI. (dev) is the device mnemonic for the terminal corresponding to the TI under which the task runs.

NOTE

If the requesting task was initiated at the requesting terminal, or if the designated task is not a multiuser task the /TI switch is not required.

EXAMPLES:

1. MCR>CANCEL XKE

   Cancels all scheduled requests for task XKE (terminates periodic rescheduling).

2. MCR>CANCEL XKE,NK111

   Cancels all scheduled requests for task XKE and NK111. (Terminates periodic rescheduling.)

6.6 DISABLE COMMAND (DIS)

FUNCTION:

The DISABLE command (DIS) allows the privileged user to inhibit task execution without actually removing the task from the system. Disabled tasks cannot be initiated until they are enabled through the ENABLE command issued by a privileged user (see Section 6.8).

FORMAT:

```
DIS[ABLE]Δtaskname[,taskname,...]
```

where:

taskname is the name of the task being disabled.

EXAMPLE:

```
MCR>DIS SCAN,RICK
```

In this example, the tasks SCAN and RICK are disabled. Neither of these tasks can be initiated until a privileged user issues the ENABLE command.

6.7 DISMOUNT VOLUME COMMAND (DMO)

FUNCTION:

The DISMOUNT VOLUME command (DMO), for the privileged user, is functionally the same as for a nonprivileged user; the difference is that only a privileged user is allowed to dismount the system volume.

6.8 ENABLE COMMAND (ENA)

FUNCTION:

The ENABLE command (ENA) reverses the effects of the DISABLE command (see Section 6.6).

FORMAT:

```
ENA[BLE]Δtaskname[,taskname,...]
```

where:

taskname   is the name of the task being enabled.

EXAMPLE:

1. `MCR>ENA SCAN`

2. `MCR>ENA SKE,RICK,JEN`

   In the examples above, tasks SCAN, SKE, RICK, and JEN are enabled, thus making them available for execution.

6.9 FIX-IN-MEMORY COMMAND (FIX)

FUNCTION:

The FIX-IN-MEMORY command (FIX) allows the privileged user to FIX a task in its default partition; i.e., to dedicate memory to a task for faster response to requests for execution. The user cannot FIX a task unless it was built as a fixable task.

FORMAT:

```
FIXΔtaskname[/TI=dev][,taskname[/TI=dev],...]
```

where:

taskname — is the name of the task being fixed in memory.

/TI=dev — is an optional switch which specifies the TI under which the task is to be fixed. dev is the device mnemonic for the terminal corresponding to the TI under which the task is to run.

EXAMPLES:

1.
```
MCR>FIX XKE
```

2.
```
MCR>FIX JAG,240Z
```

In the examples above, the tasks XKE, JAG, and 240Z are fixed in their default partitions.

3. An SGA can be fixed in memory by writing a dummy task that uses that SGA. The dummy task should be built with the following options, which are described in Chapter 3 of the <u>RSX-11D Task Builder Reference Manual</u>:

```
/-CP /-FP /FX
STACK = 10
UNITS = 0
LIBR = sga name        or        COMMON = sga name
```

Example of dummy task:

```
     .MCALL EXIT$C

START:

     EXIT$C      ; in case task is executed
     END START
```

The dummy task uses 96 words; this includes the task header and minimum stack size, plus one FTL entry. The above approach is the optimal solution to fixing an SGA.

Fixing the dummy task guarantees the loading and fixing of the SGA it uses. Both the task and SGA should be installed in the same partition.

6.10 INITIALIZE VOLUME COMMAND (INI)

WARNING

Any data stored on the volume will be destroyed when INI is executed.

FUNCTION:

The INITIALIZE VOLUME command (INI) provides the privileged user with a facility for producing new Files-11 structured volumes. Files-11 structured volumes are discussed in detail in the IAS/RSX-11 I/O Operations Reference Manual.

FORMAT:

```
INI[TVOL]Δdev:[volumelabel][/keyword(s)]
```

where:

dev: is the device specifier for any file structured media.

volumelabel is the label being assigned to the volume.

NOTE

Volume labels can be from 1 to 12 alphanumeric characters in length. The only exception is magnetic tape volumes which are from 1 to 6 alphanumeric characters in length.

Volume label must be specified for magnetic tape.

/keyword(s) define the volume characteristics. If no keywords are specified, the default characteristics are used (see individual keyword descriptions). The following keyword options are provided.

NOTE

Notations listed to the left of keyword options, equate to the following:

*= Valid for disk and DECtape only

**= Valid for magnetic tape only

/UIC=[group number,member number]

Default

```
/UIC=[1,1]
```

/PRO=[system,owner,group,world]

This keyword allows the user to establish volume access privileges. Each entry consists of from one to four letters which have the following meanings:

R - for READ access

W - for WRITE access

E - for EXTEND access

D - for DELETE access

The absence of a code letter means the access right is denied to the user.

Protection code subparameters (system,owner, group,world) are positional; therefore, the location of the entry in the parameter string defines the user to whom the codes apply.

Example

```
/PRO=[RWED,RWED,RW,RW]
```

In this example, group and world are denied extend and delete access.

Default

```
/PRO=[RWED,RWED,RWED,RWED]
```

NOTES

1. On magnetic tape, the protection option delete is equated to write, since write access implies total access because of the sequential nature of magnetic tape.

2. Extend access may be specified without write access. This implies that the user may extend the last file on the volume, but not write on the volume in any other manner.

* /MXF=maximum number of files allowed on this volume. The highest maximum permitted is one half the number of blocks on the volume or 65535 decimal, whichever is less.

Example

/MXF=100

Default

One fourth the number of blocks on the volume

* /EXT=default file extension size in blocks.

Default

/EXT=5

* /FPRO=[default file protection]

This parameter is entered using the same format as the /PRO keyword.

Default

/FPRO=[RWED,RWED,RWE,R]

* /CHA=[characteristic words]

This keyword defines the device characteristics. The possible parameters for this keyword follow.

ATCH - device may be attached for exclusive use by one task.

DCF - device control functions permitted; e.g., Read/Write logical.

Example

/CHA=[ATCH,DCF]

Default

No ATCH and no DCF

* /INF=number of file headers to allocate in the initial index file.

Default

/INF=16

* /WIN=default window size for file access to this volume. Represents the number of retrieval pointers.

Default

/WIN=7

* /LRU=number of directories to keep pre-accessed while volume is in use. For optimized directory access this value should be slightly higher than the expected number of concurrent users of the volume.

Default

/LRU=3

** /DENS=magnetic tape density

Enter 800 for 800 BPI, or 1600 for 1600 BPI.

Default

/DENS=800

* /INDX=Option - Index file position.

Options are:

BEG - beginning of volume

MID - middle of volume

END - end of volume

BLK:number - logical block number

Example

/INDX=BLK:100

Default

/INDX=MID

/BAD=[options]-Initialize bad block file. Options are:

AUTO - use bad block data left on volume by BAD command (see Section 6.3).

MAN - enter bad blocks in format described in section 6.10.1.

If both options are used, they must be separated by a comma.

Example

```
/BAD=[AUTO]
     or
/BAD=[AUTO,MAN]
```

Default

none

6.10.1 MAN Option for /BAD Keyword

When "MAN" is specified as the /BAD Keyword option, INI will issue the following prompt:

```
INI>BAD=
```

This prompt is issued after the INI command line is terminated, and before INI begins processing (initializing) the requested volume.

After INI issues the prompt, the user can begin entering bad block location data. Bad block data is entered immediately following the equal sign (=) in the following format.

```
INI>BAD=u[,n]
```

where:

u — is the logical block number, in octal, of the initial bad block in the group.

n — is the number, in octal, of consecutive blocks contained in the group. If omitted, a value of one is assumed.

NOTE

If a decimal number is used for either u or n, it must be followed by a decimal point (.).

After the first group of bad blocks is entered, INI reissues the prompt. The user can, if necessary, enter more bad block data by simply repeating the above procedure.

To terminate, the user need only enter a carriage return immediately following the equal sign (=) without entering data.

Upon termination, the initial INI command line, including the bad block data, is processed, and the requested volume is initialized.

TECHNICAL NOTES

Because of the large number of options available with the INI command, it may not be possible to enter the entire command string on a single line. For this reason, INI has been modified to accept multiline commands. To accomplish this, the user must follow the procedure outlined below.

1. MOU must be initiated as follows:

   ```
   MCR>INI
   ```

   When INI is ready to accept command lines, the following prompt is issued.

   ```
   INI>
   ```

2. At this point, multiple lines can be typed in to INI. Each line except the last must contain a hyphen (-) as the last character entered. When INI recognizes the hyphen, it assumes that the command line is incomplete, and reissues the INI> prompt for more parameters. When the last line is entered (no hyphen), INI begins processing the command. See example number 3.

EXAMPLES:

1.
```
MCR>INI DK1:TESTPACK/CHA=[ATCH,DCF]/BAD=[AUTO]
```

In this example, a disk volume on DK1: is initialized with the following characteristics:

Volume label	- TESTPACK
UIC	- [1,1]

Volume Protection	- System = RWED
	Owner = RWED
	Group = RWED
	World = RWED

Maximum number of files	- 1200
Default file extension	- 5

File protection	- System = RWED
	Owner = RWED
	Group = RWE
	WORLD = R

Characteristic word	- ATCH and DCF
Index file position	- MID
Bad blocks	- Automatically accounted for.
Number of headers in index file	- 16
Window size	- 7
Number of pre-accessed directories	- 3

2.
```
MCR>INI DK1:TESTPACK/PRO=[RWED,RWED,RW,R]/CHA=[ATCH,DCF]/BAD=[MAN]

INI>BAD=6.,10.

INI>BAD=12.

INI>BAD=
```

In the above example, a disk volume on DK1: is initialized with the following characteristics.

Volume label	- TESTPACK
UIC	- [1,1]

Volume protection	- System = RWED
	Owner = RWED
	Group = RW
	World = R

Maximum number of files	- 1200
Default file extension	- 5

File protection	- System = RWED
	Owner = RWED
	Group = RWE
	World = R

Characteristic word	- ATCH and DCF
Index file position	- MID
Bad blocks	- 6 through 17, and 12
Number of headers in index file	- 16
Window size	- 7
Number of pre-accessed directories	- 3

3.

```
MCR>INI

INI> DK2:FRED/UIC=[301,275]/LRU=2-

INI> /PRO=[RWED,RWED,R,]/INDX=BEG-

INI> /MXF=1000.

MCR>
```

In the above example, a disk volume DK2: is initialized with the following characteristics:

Volume label	- FRED
UIC	- [310,275]
Volume protection	- System = RWED
	Owner = RWED
	Group = R
	World = No access
Maximum number of files	- 1000.
Default file extension	- 5
File protection	- System = RWED
	- Owner = RWED
	- Group = RWE
	- World = R
Characteristic Word	- No ATCH and no DCF
Index file position	- BEG
Bad blocks	- Not accounted for
Number of headers in index file	- 16
Window size	- 7
Number of pre-accessed directories	- 2

6.11 INSTALL TASK COMMAND (INS)

FUNCTION:

The INSTALL TASK command (INS) installs tasks and sharable global areas (SGAs) in the system. It can also override or extend some of the attributes assigned to the task by the task builder. These overrides are specified in the form of keyword parameters appended to the task image file specifier(s) in the INS command line.

FORMAT:

```
INS[TALL]Δfilespec[/keyword(s)][,filespec[/keyword(s)],...]

                    or

INS[TALL]Δ@indirect
```

where:

filespec — is the file specifier for the image (task or sharable global area) being installed. Table 6-2 contains a list of defaults in INS file specifiers.

/keyword(s) — option parameter(s) used to override or extend arguments assigned at task-build time. The following keyword options are provided:

/PAR=partition name

This option specifies the partition into which the task or global area is to be installed.

On the installation of a sharable global area (SGA) INSTALL will recognize the special pseudo partition name $$$EXT, and will align that SGA on the external page in real memory. Furthermore, the SGA will be flagged as if it were loaded but no actual load will occur. This function allows a task to bind to a common area, for example, and reference locations symbolically in the external page.

Example

```
/PAR=GEN
/PAR=$$$EXT     (for SGAs only)
```

Default
In order:

The partition specified when building the task or SGA, or if none, The partition specified as the default during system generation.

/PRI=priority-number (decimal)

This option specifies the execution priority to be assigned to the task. Priority ranges from a low of 1 to a high of 250.

Example

```
/PRI=200
```

Default

```
/PRI=50
```

/TASK=taskname

This option allows the user to assign a name to the task or SGA being installed. This name overrides the one assigned by the Task Builder. Task names can be from 1 to 6 alphanumeric characters in length.

Example

```
/TASK=RICK
```

/POOL=pool-limit (decimal)

This option allows the user to assign a new pool limit to the task being installed.

The pool limit value can be from 0 to 255 decimal, and represents the maximum number of 8-word nodes that the task is allowed to use at one time.

Example

```
/POOL=100
```

Default

```
/POOL=40 (established by task builder)
```

/UIC=[uic]

This option allows the user to change the task's UIC, or the owning UIC of an SGA.

Example

```
/UIC=[11,11]
```

/RUN[=REM] (run-remove)

This option is different from the above mentioned options, in that it does not change any task attributes. When /RUN is specified, it directs INS to install the specified task and run it.

If the optional subparameter "=REM" is appended to /RUN, it specifies that the named task is to be removed after execution.

/TIM=allowable run time expressed as nH, nM, nS, or nT.

This option allows the user to specify the length of time the task is allowed to run. n is the decimal number of hours, minutes, seconds, or ticks the task is allowed to run. Hours, minutes, seconds, or ticks are represented as H, M, S, or T, respectively.

Example

```
/TIM=20M   (task is to run 20 minutes)
```

NOTE

This keyword has no effect if system accounting is not running, or if the task being installed was task built as a non-accountable task. In the latter case, however, a diagnostic message is issued. See Chapter 2 of the RSX-11D System Manager's Guide.

/ACC=non-owner access expressed as RO, RW, or NA

This switch allows the user to specify the access privileges afforded non-owners of a specified SGA. A non-owner of a SGA is one whose UIC does not exactly match that of the SGA. RO, RW, and NA equate to read only, write only, no access, respectively.

Example

```
/ACC=RO
```

Default

```
/ACC=NA
```

/LI and /CM

These options are used to specify that the entity being installed is either a library (/LI) or a common area (/CM). By definition, a common area has read/write access. Application of the /CM switch ensures that such data is written back to the SGA file when all binding tasks become inactive. See Chapter 5 of the RSX-11D System Manager's Guide.

Default

/CM

NOTE

Install determines that the file being installed is a sharable global area. The switches /ACC, /LI, and /CM have no effect on the installation of ordinary tasks.

/INC=task size increment

This switch overrides the EXTTSK task builder option. It specifies the decimal number of words by which the upper read/write area of the task is to be extended. The value specified will be rounded up to the next 32 word boundary. This option is illegal for shareable global area.

Example

/INC=2048

Allocates as additional 2K to the task's address space.

Default

Either 0 or the EXTTSK value.

TECHNICAL NOTES:

1. Chapter 7 or the RSX-11D Task Builder Reference Manual describes the building of a shareable global area.

2. All SGAs to which a task is bound must be installed before the task is installed. They need not be installed before building the task.

Table 6-2
Defaults in INS File Specifiers

SPECIFIER	DEFAULT
dev:	If omitted in the first or only file specifier, SY: is used. If omitted in the second through n file specifiers, the device specified or defaulted from the previous file specifier is used.
[ufd]	If omitted in a file specifier, the UFD that corresponds to the UIC under which the user logged on is used.
filename	Must be specified.
.type	.TSK
;ver	The latest version for the file.

EXAMPLES:

1. MCR>INS SCAN

   In this example, the latest version of the task image file SCAN.TSK is selected, and the task image which it contains is installed. None of the task attributes that were assigned to the task at task-build time are changed.

2. MCR>INS DK1:[11,2]SCAN.;3/PRI=103,RICK/TASK=HELP

   In this example, the tasks to be installed reside in directory file [11,2] on DK1:. Version ;3 of task image file SCAN.TSK, and the latest version of task image file RICK.TSK are selected. The task images contained in these files are installed into the system, with the following modifications being made to them. Task image SCAN will have its priority changed to 103, and task image RICK will have its name changed to "HELP".

6.12 LOAD COMMAND (LOA)

FUNCTION:

The LOAD command (LOA) allows the user to specify that an indicated device handler be made resident in memory and ready for service. I/O requests to a device are not honored unless the requested device handler is resident.

FORMAT:

```
LOA[D]Δhandler-name[:]
```

where:

handler-name   is the task name of the handler to be loaded.

EXAMPLE:

```
MCR>LOAD LP
```

The line printer device handler is loaded into memory. I/O requests to the LUN assigned to the line printer can now be honored.

6.13 MEMORY UNLOCK (MEM)

FUNCTION:

The MEMORY UNLOCK command allows the user to unlock tasks that the Executive locked in memory as a result of a main memory parity error. Parity error handling is discussed in The <u>RSX-11D System Manager's Guide</u>.

FORMAT:

```
MEM[ORY]Δ[taskname][/switches][,taskname[/switches]][,...]
```

taskname — is optional. If specified, it is the name of the task to be freed. MEM searches the ATL and unlocks the first locked task encountered with the specified name. The task name must be included if a series of specifications are included in the command.

switches — can be either or both of the following options.

/TI=xxn — indicates that the first locked task encountered in the ATL with the specified TI is to be unlocked. The value xxn is the device mnemonic; for example, TT3.

If both taskname and TI are specified in the MEM command, the first locked task encountered in the ATL with the specified taskname and TI assignment is freed.

/AL — indicates that all locked tasks are to be freed. If both taskname and /AL are specified, all locked tasks with the specified taskname are freed.

If both /TI and /AL are specified, all locked tasks with the specified TI are freed.

If taskname, /TI, and /AL are specified, all locked tasks with the specified taskname and TI are freed.

EXAMPLES:

```
MCR>MEM  /AL

MCR>MEM ABC/TI
```

6.14 MOUNT VOLUME COMMAND (MOU)

The MOUNT volume command (MOU) for the privileged user has the same format as MOU for the nonprivileged user. The privileged user does, however, have three switch options that are not available to the nonprivileged user. These options are as follows:

/UIC=[uic]

This option specifies the volume UIC to be used for the duration that the volume is mounted.

Example

```
/UIC=[11,11]
```

/OVR

This switch allows the user to override the volume label check.

NOTE

This option must be specified when mounting magnetic tapes with blank volume labels.

/PRO=[system, owner, group, world]

This option allows the user to change the volume access privileges. Each entry consists of from one to four letters which have the following meanings.

R - for READ access

W - for WRITE access

E - for EXTEND access

D - for DELETE access

The absence of one of these letters in an entry signifies that the access right is denied to the user.

Protection code subparameters (system, owner, group, world) are positional; therefore, the location of an entry in the parameter string defines the user to whom the codes apply.

Example

```
/PRO=[RWED,RWED,RW,RW]
```

In this example, group and world are denied extend and delete access.

EXAMPLE:

```
MCR>MOU DK1:SYS004/UIC=[200,200]
```

In this example, a request is made to mount disk volume SYS004 on DK1:. A request is also made to replace the volume's default UIC with "[200,200]".

6.15 OPEN REGISTER COMMAND (OPE)

FUNCTION:

The OPEN Register command (OPE) allows the privileged user to open a memory address (gain access to it) for examination and optional modification.

When a location is opened the specified memory address and the contents of the address are listed on TI:. The format of this message is shown in Figure 6-1.

FORMAT:

```
OPE[N]Δmemory address[+or-n][/option(s)]
```

where:

memory address address (octal) is the location to be opened.

n is an optional number to be added to the memory address, if positive, or subtracted from the memory address, if negative.

/option(s) The following options specify the address space to which the memory address applies:

NOTE

If no option is specified, the address is interpreted as an absolute memory address. If an option is specified, the address is interpreted as a virtual address within the specified address space.

/TASK=task name[/TI=dev]

/PAR=partition name

/KNL=specifies kernel virtual address space

FORMAT

```
     memory-address contents/new-value or line-terminator
```

where:

memory-address — (printed by the system) is the 6-digit (octal) virtual address

contents/ — (printed by the system) is the 6-digit (octal) value at the memory-address, followed by a slash (/)

new-value — (typed in by user) is the new value entered into the memory-address

line-terminator — (typed in by user) is one of the optional line terminators; see list of line terminators below

Line Terminator Options

ESC - ESCape or ALTMODE: end of command. The ESC (or ALTMODE) key is the only means of exit from the OPEN MCR function.

CR - carriage return: the next sequential location is opened.

^CR - up-arrow return: the previous location is opened.

*CR - asterisk carriage return: the location pointed to by the final contents of the opened location is opened.

Figure 6-1
Format of Open Register Command Output

EXAMPLES:

1.

```
MCR>OPEN 4 10/KNL

000014 060014/350

000016 071005/$
```

In this example, the value of the memory address is 4, "n" equals 10 (a positive number that is added to the memory address yielding the effective address "000014"). The option specified is Kernel Virtual. The System responds by printing the address (000014) and the contents of the address (060014), followed by a slash. The user responds by entering the new value "350", followed by a carriage return. Terminating the line by a carriage return causes the next sequential location to be displayed. The dialogue is then terminated by entering ALTMODE.

2.

```
MCR>OPEN 4/TASK=ABC

000004 111111/
```

In this example, memory location 4 of task "ABC" is opened. The memory address (000004) and its contents (111111) are printed. After examining the contents of this address, the operator can either enter a new value or examine other locations by typing one of the line terminator options.

3.
```
MCR>OPE 4

000004 0305/20054$
```

In this example, absolute location 4 is displayed, and modified to contain "20054".

6.16 OPERATE COMMAND (OPR)

The OPERATE command (OPR), for the privileged user, has the same format as OPR for the nonprivileged user. The privileged user does, however, have three switch options that are not available to the nonprivileged user. These options are described in Table 6-3.

Table 6-3
Operate Option Switches

SWITCH	DESCRIPTION
/CHG	Set the forms type in the PUD of the specified device to 7. When printing of the current file has completed, the user can mount new forms on the device and queue a test pattern file to test for forms alignment (see QUE command option /TE, section 5.15.)
/FO:n	Set the forms type in the PUD of the specified device to n (where n is a decimal number ranging from 0 to 6). Processing of files requiring the specified forms type and device type begins automatically.
/RE	Recycle all active orders and requeue them as inactive. Automatically begin processing queued files, directed to the specified device, having matching forms types.

6.17 ENTER PASSWORD COMMAND (PWD)

The ENTER PASSWORD command (PWD) is functionally the same for the privileged user as it is for the nonprivileged user, with this exception: the privileged user can change the password for any UFD on any volume that is not protected against system access.

6.18 QUEUE COMMAND (QUE)

FUNCTION:

The QUEUE command for the privileged user has the same format as the QUE command for the nonprivileged user. The privileged user does, however, have the option of deleting (/KIL) or modifying any queued entry in the Queue Manager tasks queue.

The format of the delete or modify command is identical to the nonprivileged delete or modify command format. The only important distinction is that any privileged user that modifies a queued entry becomes the owner of the modified entry.

6.19 REASSIGN COMMAND (REA)

FUNCTION:

The REASSIGN command (REA) allows the operator to reassign logical unit numbers (LUNs) of a particular task from one physical device unit, and assign them to another.

FORMAT:

```
REA[SSIGN]ΔtasknameΔlunlistΔdevunit:
```

where:

taskname — name of the task whose LUNs are being reassigned.

lunlist — list of one or more logical unit numbers separated by commas.

devunit — new device unit symbol followed by the unit number.

EXAMPLES:

```
1.   MCR>REASSIGN JOE 3 TT0:

2.   MCR>REA BILL 2,3,4 TT1:

3.   MCR>REA SAM 5 NONE
```

In the examples above, the requests are to:

1. REASSIGN, in task JOE, logical unit number 3 to device TT0.
2. REASSIGN, in task BILL, logical unit numbers 2, 3 and 4 to device TT1.
3. REASSIGN, in task SAM, logical unit number 5 to NONE. If 5 was previously assigned to LP0, for example, nothing would be printed on LUN 5.

TECHNICAL NOTES:

1. The REASSIGN command performs reassignments in the indicated task's disk image.
2. The REASSIGN command does not affect a memory-resident task.
3. Removing and re-installing a task will restore its LUNs to their original assignments specified at task build time.

6.20 REDIRECT COMMAND (RED)

FUNCTION:

The REDIRECT command (RED) allows the operator to redirect all I/O requests from one physical device unit to another. This command can be used if one of the I/O units needed for a task is inoperable for some reason.

FORMAT:

```
RED[IRECT]Δnew-dev:=old-dev:
```

where:

new-dev: is the new device unit symbol, followed by an optional unit number (zero is assumed).

= is notation signifying the redirect action.

old-dev: is the old device unit symbol, followed by an optional unit number (zero is assumed).

NOTE

TI: cannot be designated as old-dev, but it can be designated as new-dev.

EXAMPLES:

```
1.   MCR>REDIRECT TT3:=TT6:

2.   MCR>RED TT=LP:

3.   MCR>RED LP:=LP:
```

In the examples above the requests:

1. Redirect all I/O requests from device TT6 to device TT3;

2. Redirect all I/O requests from device LP0 to device TT0.

3. Reverse the redirection specified in example 2.

TECHNICAL NOTES:

1. The RED command does not redirect any I/O already in the queue. Previous I/O requests are not transferred.

2. If, through a sequence of Redirect commands, the user establishes a redirect chain which returns to old-dev, the following message is issued, and the RED command is rejected:

```
RED -- CIRCULAR REDIRECT CHAIN
```

Example:

```
     MCR>RED TT2:=TTl:

     MCR>RED TT3:=TT2:

     MCR>RED TTl:=TT3:

     RED -- CIRCULAR REDIRECT CHAIN
```

6.21 REMOVE COMMAND (REM)

FUNCTION:

The REMOVE command (REM) allows the privileged user to take an inactive, unfixed task out of the system.

It also allows the removal of sharable global areas whose binding tasks are not installed.

FORMAT:

```
REM[OVE]Δname1[/switch][,name2[/switch],...]

                    or

REM[OVE]Δ@indirect
```

where:

name — is the name of a task or sharable global area being removed.

/switch — an optional switch used to identify sharable global areas to be removed. The following switch values are available for use:

- /LI identifies the named global area as being a library.
- /CM identifies the named global area as being a common area.

@indirect — is an indirect command file specifier for a file which contains a list of REMOVE commands.

NOTE

Three levels of indirection are supported.

EXAMPLES:

1. MCR>REM SCAN

   In this example, the task SCAN is removed from the system.

2. MCR>REM SCAN,XKE,JAG

   In this example, tasks SCAN, XKE, and JAG are removed from the system.

3. MCR>REM SOMLIB/LI

   In this example, the library "SOMLIB" is removed, providing that none of its binding tasks are installed.

6.22 RUN COMMAND (RUN)

FUNCTION:

The RUN command allows the privileged user to initiate tasks in any of six different forms as follows:

1. Request the execution of a task when memory becomes available.

2. Execute a task, contingent upon immediate memory availablity

3. Run a task at a specified future time and, optionally, rerun it periodically

4. Schedule the task for running at a specified future time and, optionally, rerun it periodically

5. Synchronize the running of a task with a time unit and, optionally, rerun it periodically

6. Run a task that is not installed in the system. This form of the RUN command is described in section 5-17.

NOTE

1. If the user is running an uninstalled task, the only switch options available for use are

   /UIC=, /PAR=, /PRI=.

2. The default UIC for the general user is the UIC under which the user logged on. For the privileged user, the default UIC of the task being run is used.

REQUEST form of command

The "Request" form of the RUN command allows the operator to request the execution of tasks.

NOTE

If memory is not available, the task will be queued until memory becomes available.

FORMAT:

```
RUNΔtaskname[options][,taskname[options],...]($)
```

where:

taskname   is the name of the task being run

options   are any of the following:

/PRI = priority number (decimal)

/PAR = partition name

/UIC = [user identification code]

EXAMPLES:

```
MCR>RUN SCAN($)

MCR>RUN SCAN/PAR=XYZ/PRI=58($)

MCR>RUN XKE,NK111/PRI=10,SCAN($)

MCR>RUN XKE/UIC=[11,1]($)
```

TECHNICAL NOTE:

1. Execution is contingent upon priority and memory availability.

2. See Section 4.2 for a description of the UIC under which the task will run.

EXECUTE form of command

The "Execute" form of the RUN command allows task execution contingent upon priority.

NOTE

> If no memory is available in the specified or default partition, the request is rejected.

FORMAT:

```
        RUNΔtaskname/MEM[/options][,taskname/MEM[/options],...](Ⓢ)
```

where:

taskname — is the name of the task being run

options — are any of the following:

/PRI = priority number

/PAR = partition name

/UIC = [user identification code]

EXAMPLES:

```
     MCR>RUN SCAN/MEM(Ⓢ)

     MCR>RUN SCAN/MEM/PRI=100/PAR=XXX/UIC=[1,3](Ⓢ)

     MCR>RUN SCAN/MEM/PRI=25,240Z/PAR=Z/PRI=99(Ⓢ)
```

TECHNICAL NOTES:

1. Execution is contingent upon priority.
2. The "Execute" form of the RUN command does not cause checkpointing to occur.

RUN form of command

The "Run" form of the RUN command allows the operator to schedule tasks in terms of "delta time from now" with the option to specify periodic reschweuling, run priority, and memory partition.

FORMAT

```
RUNΔtasknameΔtime[options][,tasknameΔtime[options],...] $
```

where:

taskname — is the name of the task being run

time — is the "delta time from now" the task is to begin execution. Delta time is expressed as nH, nM, nS, or nT where n is the decimal number of hours, minutes, seconds, or ticks.

options — are any of the following:

/RSI = reschedule interval expressed as nH, nM, nS, or nT

/PRI = priority number (decimal)

/PAR = partition name

/UIC = [user identification code]

EXAMPLES:

```
1.   MCR>RUN XKE 15M $

2.   MCR>RUN XKE 15M/RSI=90S/UIC=[3,1] $

3.   MCR>RUN XKE 15M/RSI=GOS/PRI=150/PAR=XYZ $

4.   MCR>RUN XKE 15M,NK111 20M,240Z 25M $
```

In the examples above, the requests are to:

1. Run task XKE 15 minutes from now

2. Run task XKE 15 minutes from now, rescheduling it every 90 seconds. The user identification code is [3.1]

3. Run task XKE in 15 minutes, rescheduling it every 90 seconds, assigning it a priority of 150, and running it in partition XYZ.

4. Run task XKE in 15 minutes, run task NK111 in 20 minutes, and run task 240Z in 25 minutes.

SCHEDULE form of command

The "Schedule" form of the RUN command allows the operator to schedule tasks in terms of absolute time of day, with the option to specify periodic rescheduling, run priority, and memory partition.

FORMAT:

```
RUNΔtasknameΔtime[options] ($)
```

where:

taskname   is the name of the task being run

time   is the absolute time of day the task is to begin execution. Time is expressed as hours: minutes: seconds [:ticks]

options   are any of the following:

/RSI = reschedule interval, expressed as nH, nM, nS, or nT.

/PRI = priority number

/PAR = partition name

/UIC = [user identification code]

EXAMPLES:

```
1.  MCR>RUN XK1 12:23:15($)

2.  MCR>RUN XKE 12:00:00/RSI=6M/PRI=99/PAR=XYZ ($)

3.  MCR>RUN SCAN 23:30:00/RSI=10M/PRI=200($)

4.  MCR>RUN XKE 1:00:00,240Z 1:30:00/UIC=[111,1] ($)
```

In the examples above the requests are to:

1. Schedule task XK1 at 12:23:15.

2. Schedule task XKE at 12:00, reschedule it every six minutes, assign it a priority of 99, and execute it in partition "XYZ".

3. Schedule task SCAN at 23:30, reschedule it every ten minutes, assign it a priority of 200.

4. Schedule task XKE at 1:00, task 240Z at 1:30; (with a user identification code of [111.1]).

SYNCHRONIZE form of command

The "Synchronize" form of the RUN command allows the operator to synchronize the scheduling of a task in terms of delta time from "clock unit synchronization", with the option to specify periodic rescheduling, run priority, and memory partition.

FORMAT:

```
RUNΔtasknameΔstartimeΔ+time[options]($)
```

where:

taskname   is the name of the task being run

startime   is the synchronization clock unit as follows:

H = synchronize on the next hour.

M = synchronize on the next minute.

S = synchronize on the next second.

T = synchronize on the next tick.

time   is the delta time increment that is added to the synchronization clock unit yielding the actual start time. Time is expressed as nH, nM, nS, or nT where n is the number of hours, minutes, seconds, or ticks.

options   are any of the following:

/RSI = rescheduling interval expressed as nH, nM, nS, or nT

/PRI = priority number

/PAR = partition name

/UIC = [user identification code]

EXAMPLES:

```
MCR>RUN NK111 H+10M($)

MCR>RUN NK111 S+10M/PRI=10,240Z M+0S/RSI=15S/PRI=90($)
```

In the example above the requests are to:

1. Synchronize task NK111, schedule for 10 minutes after the next hour.

2. Synchronize task NK111, schedule for 10 minutes after the next second, assign it a priority of 10; Synchronize task 240Z, schedule it for 0 seconds after the next minute, rescheduling it every 15 seconds, and assign it a priority of 90.

6.23 SAVE COMMAND (SAVE)

WARNING

The SAVE command should be executed only when the system is quiescent.

FUNCTION:

THE SAVE command is used to record the core image of an RSX-11D system on the disk from which it was originally bootstrapped, so that a bootstrap can reload it and start up the system.

FORMAT:

```
SAV[E]Δ[/switch]
```

where:

/switch is one of the following optional switches:

/MOU=dev:[dev:...:dev:]

This switch is used to allow the system to be saved with the specified devices mounted. See Technical Note 3.

/LOG=TTn:[TTn:...:TTn:]

This switch is used to allow the system to be saved with the specified terminals logged on.

/NO, /NOX, /NOT, /NOXT

These switches all inhibit automatic memory expansion or truncation when the saved image is rebooted. If none of these switches is used, SAVE determines exactly how much memory exists and expands or truncates the highest partition in the system.

NOTE

If the user attempts to save a system that still has volumes mounted or terminals logged on, and the /MOU or /LOG switches have not been specified, the save will not occur.

EXAMPLE:

1.

```
MCR>SAVE
```

In this example, the current status of the system is saved on the disk from which it was originally bootstrapped. System changes made by the REDIRECT or other MCR commands are also saved with the system core image.

2.

```
MCR>SAVEΔ/MOU:DF0:
```

In this example, the system is saved with DF0: mounted.

TECHNICAL NOTES

1. Because the SAVE command provides a copy of a completed system configuration, it must be used only when the system is quiescent. SAVE ensures that the system is quiescent by searching the system data base for any of the following conditions:

   - Mounted devices,
   - Users logged onto terminals other than the one from which the SAVE command was issued,
   - Tasks with I/O in progress,
   - Tasks being loaded or checkpointed,
   - Shareable global areas being loaded or, in the case of read/write common areas, being recorded on disk,
   - Tasks loaded or fixed beyond the end of the SAVE file,
   - Shareable global areas including the read-only root of multiuser tasks loaded beyond the end of the save file.

   If any of these conditions is detected, SAVE issues an appropriate error message.

2. The SAVE command will attempt to record (on the system device) all memory specified at SYSGEN time. If more memory exists than was declared at SYSGEN, only the declared memory will be saved. If, however, less memory exists than was declared at SYSGEN, only the amount that exists will be saved. For a PDP-11/70, no more than 124K will be saved.

3. SAV will not permit the system to be saved with volumes mounted or terminals logged on, unless the appropriate /MOU or /LOG switches are specified.

   In the case where the /MOU switch is specified, for Files-11 volumes the user must take the following information into account:

   a. When a volume is mounted, volume control data is established in memory to reflect the volume's current file status. This data is updated with every file operation.

   b. When a volume is dismounted, the volume's control data is reset.

   c. If the user selects to save the system with volumes mounted, the volume control data for each mounted volume is saved reflecting the current status of the volume. It is imperative that no file activity occur on the volume between the time the system was saved, and the next bootstrapping of the system. Otherwise, the integrity of the volume will be destroyed.

d. When the system is re-bootstrapped, the status of the volume must be exactly the same as it was when the system was saved, or the integrity of the volume will be destroyed on the very first file operation.

NOTE

After the system is re-bootstrapped, and before any file activity has occurred, the user can execute the following commands to ensure the integrity of any volume:

```
MCR>DMO dev:

MCR>MOU dev:
```

This will reset the volume's control data to reflect the current volume status.

In the case where the /LOG switch is specified when the system is re-bootstrapped, the specified terminals are automatically logged on having the same UIC and privilege they had when the system was saved.

4. When a saved system is rebooted or restarted, the system is expanded or truncated automatically to the amount of physical memory available unless the /NOXT switch was used when the system was saved.

Because SAVE is active when a save is performed, it is active when the reboot/restart occurs. It is, in fact, the task SAVE that restarts an RSX-11D system and performs the memory size calculations.

To expand the system, SAVE determines how much physical memory responds by testing memory in 4K word increments on a PDP-11/40, 45, or 50 or by reading the memory size register on PDP-11/70. If the address at the top of memory is greater than that at the last save, or at system generation if no save was performed, SAVE performs the following steps:

. Increases the size of the last (highest addressed) partition,

. Adjusts the hole pointer,

. Prints a explanatory message on the terminal from which the SAV was initiated.

If the actual memory is less than the amount contained in the save file, SAVE attempts to truncate one or more partitions starting from the highest end of memory. Truncation fails and an appropriate message is printed under the following circumstances:

. Any occupied partition size reduces to zero,

. Any truncation occurs in an occupied user-controlled partition, Truncation in a system-controlled partition is so extensive that a previous occupied area of memory does not exist.

In summary, unoccupied partitions can be truncated to size zero. Unoccupied parts of system-controlled partitions also can be truncated. Successful truncation results in a descriptive message printed at the terminal.

5. Irrespective of any previous memory expansion or truncation, the save file is always the length specified at system generation and only that amount of memory is saved. This fact is the reason for the checking described in technical note 1 on entities loaded beyond the end of the save file.

6. Chapter 5 of the RSX-11D System Generation Reference Manual contains more details about SAVE.

6.24 SET COMMAND (SET)

FUNCTION:

The SET command provides the privileged user with a facility for changing the terminal default UIC, the MCR timeout, the terminal's characteristics and privileges, for setting or clearing output spooling on a device and for controlling cache memory on PDP-11/70.

FORMAT:

```
SETΔ[/keyword(s)]
```

where:

/keyword — is one or more optional keywords which, when specified, alter or set the default values they represent. The following keyword options are provided:

/UIC=[group,member]

This option allows the user to change the therminal default UIC to the one specified to the right of the equal sign(=).

Example: /UIC=[200,200]

/TMO=nnnnn

This option allows the user to change the MCR timeout value. The number nnnnn is a decimal value (any number up to 32767) representing the number of seconds MCR is to wait for a command before timing out.

Example: /TMO=90

/LA30S=dev:[dev:...:dev:]

This option defines the specified device(s) as having LA30S characteristics.

Example: /LA30S=TT1:TT2:

/LA30P=dev:[dev:...:dev:]

This option defines the specified device(s) as having LA30P characteristics.

Example: /LA30P=TT1:TT2:

/KSR33=dev:[dev:...:dev:]

This option defines the specified device(s) as having KSR33 characteristics.

Example: /KSR33=TT1:TT2:

/PRV=dev:[dev:....:dev:]

This option defines the specified device(s) as privileged terminal device(s).

Example: /PRV=TT1:TT2:

/SLV=dev:[dev:....:dev:]

This option defines the specified device(s) as slave terminal device(s).

Example: /SLV=TT1:TT2:

/-PRV=dev:[dev:....:dev:]

This option defines the specified device(s) as nonprivileged terminal device(s).

Example: /-PRV=TT1:TT2:

/-SLV=dev:[dev:....:dev:]

This option defines the specified device(s) as nonslave terminal device(s).

Example: /-SLV=TT1:TT2:

/SP=dev:[dev:....:dev:]

This option defines the specified device(s) as being available to the output despooler for the spooling of spooled output files.

Example: /SP=LP:TT1:

/-SP=dev:[dev:...:dev:]

This option defines the specified device(s) as no longer being available to the output despooler for the spooling of output files.

Example: /-SP=TT1:TT2:

/INB=dev:[dev:...:dev:]

This option defines a terminal(s) as having intermediate buffering.

Example: /INB=TT3:

/-INB=dev:[dev:...:dev:]

This option discontinues intermediate buffering on the specified terminal(s).

/CAC=n

This option specifies the number of parity errors that can occur in a PDP-11/70 cache group in one minute before the Executive turns the group off. The value n can be from 1 to 32,767 (decimal). The default value for n is 50. Parity error handling is discussed in the RSX-11D System Manager's Guide.

/CAC=ONn

This options allows the user to turn on a PDP-11/70 cache memory group that had previously been turned off. The value n indicates the number of the group.

/CAC=OFn

This option allows the user to turn off a PDP-11/70 cache memory group that is on. The value n indicates the number of the group.

EXAMPLE:

1. MCR>SET /UIC=[20,20]/TMO=90/LA30S=TT1:TT2:

   In this example, the following actions are performed:

   a. The system UIC is set to [20,20].

   b. The MCR timeout is set to 90 seconds.

   c. TT1: and TT2: are defined as having LA30S characteristics.

2. MCR>SET /LA30S=TT1:TT2:TT3:/PRV=TT1:TT2:/SP=TT3:

   In this example, the following actions are performed.

   a. TT1:, TT2:, and TT3: are defined as having LA30S characteristics.

   b. TT1: and TT2: are defined as being privileged terminals.

   c. TT3: is made available to the output despooler for the spooling of output files.

6.25 TIME COMMAND (TIM)

FUNCTION:

The Time command (TIM) allows the operator to list the time and date, or to alter the time and date values in the system clock calendar.

FORMAT:

```
TIM[E]Δ[time]Δ[date]
```

where:

time is specified in the format hour:minute:second

date is specified in the format month/day/year

NOTE

If the TIME command is executed without parameters, the time and date that the system is currently running under will be listed. Also, if either or both of the parameters (time or date) are specified, the corresponding field in the system will be altered to the value specified by the parameter.

EXAMPLE:

1. `MCR>TIM 9:10:20 3/21/75`

   In this example the time and date in the system clock calendar are changed to 9:10:20 and 3/21/75, respectively.

6.26 USER FILE DIRECTORY COMMAND (UFD)

FUNCTION:

The User File Directory command (UFD) creates a user file directory (UFD) on the specified volume and enters its file name into the master file directory (MFD). The UFD command accepts the [group,owner] number as both the UFD name and the file owners' UIC.

FORMAT:

```
UFDΔdevice:[uic][/switch]
```

where:

device: specifies the device that contains the volume to be acted upon.

[uic] is the UIC of the UFD being created.

NOTE

The brackets which enclose the UIC are a required part of the parameter.

/switch is one or more of the optional UFD switches described in Table 6-4.

EXAMPLE:

```
MCR>UFD DK1:[1,1]/ALLOC=100.
```

In the example above, the request creates the UFD on disk device DK1 with the UIC [1,1] as the directory name; space is allocated for 100 directory entries.

Table 6-4
UFD Switches

SWITCH	DESCRIPTION
/PRO	This switch allows the owner of the directory to selectively permit access to his file. This switch is specified in the following format: /PRO=[system,owner,group,world] NOTE If the /PRO switch is not specified, the default file protection for the volume is used.
/ALLOC	This switch allows the owner of the directory to pre-allocate space for a specified number of directory entries. This switch is specified in the following format: /ALLOC=number-of-entries NOTE The number specified is assumed to be octal; the number may be specified with a trailing decimal point to represent a decimal value. The default is 32 octal.

6.27 UNFIX COMMAND (UNF)

FUNCTION:

The Unfix command (UNF) allows the privileged user to free "Fixed" tasks from memory.

FORMAT:

```
UNF[IX]Δtaskname[/TI=dev][,taskname[/TI=dev],...]
```

where:

taskname	is the name of the task being unfixed.
/TI=dev	is an optional switch (appended to the taskname) which allows the user to unfix any task in the system by specifying its TI (dev is the device mnemonic for the terminal corresponding to the TI under which the task is run).

EXAMPLES:

1. MCR>UNFIX XKE

2. MCR>UNFIX 240Z,NK111

In the example above the requests:

1. Unfix task XKE, freeing it from memory; and,

2. Unfix task 240Z and Task NK111.

6.28 UNLOAD COMMAND (UNL)

FUNCTION:

The UNLOAD command (UNL) causes the indicated device handler to exit, releasing its memory and causing the device to become inaccessible.

FORMAT:

```
UNL[OAD]Δhandler-name[:]
```

EXAMPLE:

1. MCR>UNLOAD LP

   In the example above the request unloads the line printer handler task.

CHAPTER 7

MAGNETIC TAPE VOLUMES

The magnetic tape file system is a privileged task named MTAACP that provides file primitive services to FCS. The file system also uses an associated message task named F11MSG that must be installed before a magnetic tape volume can be mounted.

The magnetic tape file system processes tape volumes that have been initialized through the INITVOL MCR function. The mounting and dismounting of a volume set is accomplished using the MOU and DMO MCR functions.

Once a volume set is mounted, all volume or reel switching is transparent to the FCS level of the system.

NOTE

> It is not necessary that all reels in a volume set be mounted on tape units simultaneously.

7.1 INITIALIZING TAPE VOLUMES

Initializing a magnetic tape volume is an important step in preparing a magnetic tape file system for use. All protection is at the volume level; therefore, to ensure proper use of volume sets, each volume must have a unique volume identifier (label). The system manager should establish a method of assigning unique volume labels and external identification.

7.2 MOUNTING A VOLUME SET

The MOU MCR command is used to mount the volume set. The following information should be kept in mind when mounting a volume set.

1. Any number of tape units can be assigned to the volume set. All units must be compatible with each other (i.e., they must be either 9- or 7-track and 1600 or 800 bpi).

2. All volume labels or only the first volume label in the set can be specified. See the examples below.

The number of volume labels specified in the MOUNT command affects the form of the operator mount message printed on the console. The system assigns relative volume numbers sequentially, starting with volume 1 and continuing through the list of volumes. The system assumes that the volumes, in the volume set are mounted physically as indicated in the MOU command.

However, only relative volume 1 is verified at mount time; the verification of subsequent volumes is performed when these volumes are needed.

3. Volumes produced by the magnetic tape file system of RSX-11D versions before version 6 (.FL300) are recognized automatically. The volumes produced under previous versions can only be read; they never can be written by the magnetic tape file system.

The file set identifier, the file section number, and the file sequence number in HDR1 file labels are used by the file system to identify individual volumes in a set of volumes.

The following are examples of MOU commands for volume sets.

```
MCR>MOU MT0:1STVOL
```

1STVOL is the first volume of a volume set. It is mounted on MT0 which is the unit on which all subsequent volumes are to be mounted.

```
MCR>MOU MT(0,1):(1STVOL,2NDVOL)
```

1STVOL is mounted on MT0. 2NDVOL either is or will be mounted on MT1. If 2NDVOL is not mounted when required, a message requesting operator action is printed on the console.

7.3 DISMOUNTING THE VOLUME

The DMO MCR command is used to dismount a volume set. The following information should be kept in mind when dismounting a volume set.

1. All units assigned to the volume set are dismounted by the dismounting of any unit assigned to the volume set.

2. Individual DISMOUNT COMPLETE messages are printed on the operator's console for each unit assigned to the volume set.

3. All units are rewound and placed off-line unless the unit is currently off-line or rewinding.

7.4 OPERATOR INTERACTION WITH THE MAGNETIC TAPE FILE SYSTEM

The magnetic tape file system issues messages to the console output device (CO) through the file system message task (F11MSG). These messages are requests for operator action. The operator is expected to perform the requested action for the specified tape unit and then place the unit on-line.

The magnetic tape file system automatically detects that the requested unit has been placed on-line and restarts the process that was attempted before the request was issued.

CHAPTER 8

NODE POOL STATUS PROGRAM

The node pool status program (POOL) provides the privileged user with statisitics about pool usage, and prints these statistics on the terminal. Once the POOL program is initiated, it executes every two seconds and prints the number of nodes available (unused) in the pool along with the largest contiguous amount of node space. All numbers (decimal) are in terms of 8-word nodes.

Type the following command to cause POOL to execute:

```
MCR>RUN POOL($)
```

See Figure 8-1.

To terminate POOL execution, the user must press CTRL C to obtain the MCR> prompt and enter the following command:

```
RES POOL($)
```

This causes POOL to print summary information and exit. The summary information consists of the following:

1. The largest contiguous amount of node space
2. The smallest amount of node space
3. The largest number of available nodes
4. The smallest number of available nodes.

This summary information is accumulated from the time POOL is initiated, to the time it is terminated. See Figure 8-1.

```
MCR>RUN POOL($)

NODES = 745   LARGEST   HOLE = 113
NODES = 759   LARGEST   HOLE = 113
NODES = 751   LARGEST   HOLE = 113
NODES = 753   LARGEST   HOLE = 113
NODES = 746   LARGEST   HOLE = 113
NODES = 742   LARGEST   HOLE = 113

MCR>RES POOL($)

             NODES = 728   LARGEST   HOLE = 113
LARGEST NUMBER OF NODES = 759
SMALLEST NUMBER OF NODES = 728
LARGEST HOLE = 113
SMALLEST HOLE = 113
```

Figure 8-1
Sample POOL Execution And Termination

APPENDIX A

MCR COMMAND SUMMARY AND ERROR MESSAGES

A.1 MCR COMMAND SUMMARY

ABORT (Non-privileged)

```
ABOΔtaskname[,taskname,...]
```

ABORT (Privileged)

```
ABOΔtaskname[/TI=dev]
```

ACTIVE TASK LIST (Non-Privileged)

```
ACTΔ[taskname][/switch(es)]
```

Switches:

```
/FU - Full listing
/SH - Short listing
/TI=dev - All tasks for specified TI
```

ALTER PRIORITY (Privileged)

```
ALTΔtaskname[/TI=dev]/PRI=nnn[,taskname[/TI=dev]/PRI=nnn,...]
```

BAD BLOCKS (Privileged)

```
BADΔdev:
```

BOOTSTRAP (Privileged)

Format 1

```
BOO

BOO>
```

Format 2

```
BOOΔfilespec
```

Format 3

```
BOOΔ/WB
```

Format 4

```
BOOΔfilespec/WB
```

BYE (Non-Privileged)

```
BYE
```

CANCEL (Privileged)

```
CANΔtaskname[/TI=dev][,taskname[/TI=dev],...]
```

CONTINUE (Non-Privileged)

```
CONΔtaskname[,taskname,...]
```

DISABLE (Privileged)

```
DISΔtaskname[,taskname,...]
```

DISMOUNT VOLUME (Non Privileged)

```
DMOΔdev:[volumelabel][/UIC=[uic]][/LOCK]
```

DISMOUNT VOLUME (Privileged)

Same as non-privileged with this exception: only a privileged user is allowed to dismount the system volume.

ENABLE (Privileged)

```
ENAΔtaskname[,taskname,...]
```

FIX IN MEMORY (Privileged)

```
FIXΔtaskname[/TI=dev][,taskname[/TI=dev],...]
```

HELLO (Non-Privileged)

```
HELΔ[uic]

PASSWORD>
```

INITIALIZE VOLUME (Privileged)

```
INIΔdev:[volumelabel][/keyword(s)]

     Keywords

     /UIC=[uic]

     /PRO=[system,owner,group,world]
```

/MXF=maximum number of files allowed on this volume

/EXT=default file extension size in blocks

/FPRO=[default file protection]

/CHA=[characteristic word]
ATCH or DCF or both

/INF=number of file headers in file index

/WIN=default window size for file

/LRU=number of directories to keep accessed

/DENS=magnetic tape density

800 or 1600

/INDX=index file position option

BEG - beginning of volume

MID - middle of volume

END - end of volume

BLK:nnn - logical block number

/BAD=option (Initialization of bad block file)

AUTO - use bad block data left on volume by BAD command task.

MAN - enter bad blocks manually.

```
INI>BAD=u[,n]
```

INSTALL TASK (Privileged)

```
INSΔfilespec[/keyword(s)][,filespec[/keyword(s)],...]

            or

INSΔ@indirect
```

Keywords

/PAR = partition name

/PRI = priority-number (decimal)

/TASK = taskname

/POOL = pool-limit (decimal)

/UIC = [uic]

/RUN = [REM] (run-remove)

/TIM = allowable run time expressed as nH, nS, nM, or nT.

```
          /ACC = non-owner access expressed as

               RO - read only
               RW - write only
               NA - no access (default)

          /LI = library file

          /CM = common area

          /INC = task size increment

LOAD (Privileged)

     LOAΔhandler-name

LOG COMMENT (Non-Privileged)

     LOGΔcomment-line

             or

     ;Δcomment-line

             or

     !Δcomment-line

LOGICAL UNIT NUMBERS (Non-Privileged)

     LUNΔtaskname[,taskname,...]

MEMORY UNLOCK (Privileged)

     MEMΔtaskname[/options][,taskname,...]

          Keywords

          /AL - all such tasks

          /TI = TTn - all tasks with this TI

MOUNT VOLUME (Non-Privileged)

     MOUΔdev:volumelabel[keyword(s)]

          Keywords

          /CHA=[characteristics word]

               FOR - foreign volume

               ATCH - device can be attached exclusively
                      by one task

               DCF - device control functions allowed

          /UNL (volumes index file is unlocked)

          /DENS = magnetic tape density
```

```
                800 or 1600

          /ACP = taskname (file processor for the volume)

          /EXT = default file extend increment in blocks

          /FPRO = [default file protection]

          /LRU = number of directories to keep pre-accessed

          /OVRFSID (override the set identifier check)

          /OVREXP (override the expiration date check)

     Format for Mounting Multi-Volume Magnetic Tape

     MOUΔMT(n1[,n2,...,nn]):(label1[,label2,...,labeln])[/keyword(s)]

MOUNT (Privileged)

     MOUΔdev:volumelabel[/keyword(s)]

          Keywords

          All of the above-mentioned keywords, plus the following:

          /UIC =[uic]

          /OVR (override the volume label check)

          /PRO = [system,owner,group,world]

OPEN REGISTER (Privileged)

     OPEΔmemory address[+ or -n][/options]

          Options

          /TASK = taskname[/TI=dev]

          /PAR = partition name

          /KNL (specifies kernel virtual address)

OPERATE (Non-Privileged)

     OPRΔdev:/switch[,dev:/switch,...]

          Switches

          /SP - Start printing

          /AB - Abort printing

          /ST - Stop printing

          /RS - Resume printing

          /RS:T - Resume printing from last
                    encountered form feed.
          /RS:TOF - Resume printing from top-of-file

OPERATE (Privileged)

     OPRΔdev:/switch[,dev:/switch,...,dev:/switch]

          Switches

          All of the above mentioned switches plus the following:

          /CHG - Change forms

          /FO:n - Set the form type to n

          /RE - Recycle all active orders and
                requeue them as inactive

PASSWORD (Non-Privileged)

     PWDΔ[ufd]

     PASSWORD>password

PASSWORD (Privileged)

     Same as non-privileged with this exception;  the privileged  user
     can  change  the password for any UFD on any volume not protected
     against system access.

QUEUE (Non-Privileged)

     Queue a File For Printing

     QUEΔ[dev:=]infile-1[,infile-2,...,infile-n][/switch(es)]

                                  or

     QUEΔ@indirect

          Switches

          /PR:nnn - Set  the  priority  of  output  selection  to  nnn
                    (default /PR:50)

          /FO:n   - Set form type to n (default /FO:0)

          /CO:nn  - Set number of copies to nn (default /CO:1)

          /DE     - Delete file after printing (default /-DE)

          /TE     - Test for forms alignment

     List Elements in Queue

     QUEΔ/LI

     Kill (Delete) Elements in Queue

     QUEΔ/KI:seq[:seq:...:seq]

     Modify Elements in Queue
     QUEΔ[dev:=]/MO:seq[:seq:...:seq][/switch(es)]

          Switches

          /PR:nn  - Change priority to nnn.

          /FO:n   - Change form type to n.

          /CO:nn  - Change number of copies to n.

          /DE     - Change delete/preserve indicator to delete
                    (default /-DE)
```

REASSIGN (Privileged)

```
     REAΔtasknameΔlunlistΔdev:
```

REDIRECT (Privileged)

```
     REDΔnew-dev:=old-dev:
```

REMOVE (Privileged)

```
     REMΔname1[/switch][,name2[/switch],...]

                              or

     REMΔ@indirect
```

RESUME (Non-Privileged)

```
     RESΔtaskname[/TI=dev][,taskname[/TI=dev],...]
```

RUN (Non-Privileged)

```
     RUNΔfilename($)

          or

     RUNΔ$filename($)
```

RUN (Privileged)

```
     (Request)

     RUNΔtaskname[/option][,taskname[/option],...]($)

          Options

          /PRI=priority number (decimal)

          /PAR=partition name

          /UIC=[user identification code]

     (Execute)

     RUNΔtaskname/MEM[/option][,taskname/MEM[/option],...]($)
```

Options

/PRI=priority number (decimal)

/PAR=partition name

/UIC=[user identification code]

(Run)

```
RUNΔtasknameΔtime[/options][,tasknameΔtime[/options],...] $
```

Options

/RSI=reschedule interval expressed as nH, nM, nS, or nT

/PRI=priority number (decimal)

/PAR=partition name

/UIC=[user identification code]

(Schedule)

```
RUNΔtasknameΔtime[/options] $
```

Options

/RSI = reschedule interval expressed as nH, nM, nS, or nT

/PRI = priority number (decimal)

/PAR = partition name

/UIC = [user identification code]

(Synchronize)

```
RUNΔtasknameΔstartimeΔ+time[/options] $
```

Options

/RSI = rescheduling interval expressed as nH, nM, nS, or nT

/PRI = priority number (decimal)

/PAR = partition name

/UIC = [user identification code]

/TIM = timeout limits for accountable tasks

SAVE (Privileged)

```
SAVΔ[/switch]
```

Switches

/MOU=dev:[dev:...dev:] - save system with dev: mounted

/LOG=TTn:[TTn:...:TTn:] - save system with specified terminals logged on.

/NOXT - inhibit memory expansion/truncation on reboot.

SET (Privileged)

SETΔ[/keyword(s)]

Keywords

/UIC=[group,member]

/TMO=nnnnn - change MCR timeout value to nnnnn.

/LA30S=dev:[dev:...:dev:] - device has LA30S characteristics

/LA30P=dev:[dev:...:dev:] - device has LA30P characteristics

/KSR33=dev:[dev:...:dev:] - device has KSR33 characteristics

/PRV=dev:[dev:...:dev:] - device is defined as a privileged terminal device

/SLV=dev:[dev:...:dev:] - device is defined as a slave terminal device

/-PRV=dev:[dev:...:dev:] - device is defined as a non-privileged terminal device

/-SLV=dev:[dev:...:dev:] - device is defined as a non-slave terminal device

/SP=dev:[dev:...:dev:] - devices available to the Output Despooler

/-SP=dev:[dev:...:dev:] - devices no longer available to the Output Despooler

/INB=dev:[dev:...:dev:[- device is defined as having intermediate buffering.

/-INB=dev:[dev:...:dev:] - device is defined as haveing intermediate buffering recended.

SYS (Non-Privileged)

SYSΔ[/switch[:opt]]

Switches

/ATL - List names of tasks in the active task list

/MRL[:opt] - List names of tasks in memory required list

/CKQ - List name of task in the clock que

/GCO - Lists names of Libraries and comman area in the global common directory

/IRQ[:opt] - List names of tasks with entries in the I/O request queue

/CTL[:opt] - List name of tasks in the checkpointable task list

/FTL - List names of tasks in the fixed tasks list

/SRQ - List names of tasks that have sends queued for them

/ASQ - List names of tasks in the ATL that have ASTs queued for them

/SGA - List names of tasks in the STD and the SGAs to which each is bound

/BRF - Lists names of tasks in the ATL, MRL, or CKQ

/FUL - Lists all the above except /SGA

/TAS - Same as TAS command

/DEV - Same as DEV command

/PAR - Same as PAR command

/COM - Same as COM command

TIME (Non-privileged)

TIM

TIME (Privileged)

TIMΔ[time]Δ[date]

USER FILE DIRECTORY (Privileged)

UFDΔdev:[uic][/switch]

Switches

/PRO=[system,owner,group,world]

/ALLOC=number-of-entries

UNFIX (Privileged)

UNFΔtaskname[/TI=dev][,taskname[/TI=dev],...]

UNLOAD (Privileged)

UNLΔhandler-name[:]

WHO (Non-Privileged)

WHO

A.2 MCR ERROR MESSAGES

The error messages in this appendix are listed in alphabetic order by issuing task. The only exceptions are the messages issued by the task MCRERR. It prints the user-entered command that caused the error and then, on the following line, prints the error message. Messages with no 3-letter prefix can be identified as coming from MCRERR. These messages are presented in Section A.2.1

ACT -- INVALID ADDRESS IN A.TI

The active task list contains a TI that has an invalid address.

Notify the DEC software specialist.

ACT -- INVALID DEVICE

The device mnemonic, specified by the user, is either invalid or specified in a nonstandard format.

ACT -- INVALID SWITCH

The user specified a switch that ACT does not recognize. Either the switch is not an ACT switch, or it was specified in the wrong format.

ACT -- INVALID SWITCH COMBINATION

The user specified switches that are mutually exclusive; see the description of ACT in Chapter 5 to determine which switches can be used together.

ACT -- INVALID TASK NAME

The user specified a task name that either contains more than 6 alphanumeric characters or one that contains a non-alphanumeric character.

ACT -- NO TASKS ACTIVE ON TERMINAL

or

ACT -- TASK NOT ACTIVE

Self-explanatory.

ACT -- TOO MANY TASKS FOR BUFFER

ACT can only process a maximum of 10 active tasks, and the active task list contains more than 10. Only the first 10 tasks in the queue are listed at the user terminal.

To make ACT handle more than 10 active tasks, the system manager must task build ACT to accommodate more tasks. This is accomplished by modifying ACT's size, and the size of its processing buffer (MAXTSK).

ACT -- THE /FU SWITCH REQUIRES A TASK NAME

Self-explanatory.

BAD -- BAD BLOCK FILE OVERFLOW

More than 102 groups of bad blocks (each group consisting of from 1 to 256 bad blocks) were detected. The disk pack is defective.

BAD -- BLOCK 0 BAD - DO NOT USE AS SYSTEM DEVICE

This is a warning message; it requires no user action. BAD has terminated normally; and all bad block information is available for INITVOL.

BAD -- DEVICE HANDLER MISSING

The required disk handler task is not loaded.

Load the required handler and retry the command.

BAD -- DEVICE NOT IN SYSTEM

The specified device was not included as a usable device when the system was generated. That is, there is no PUD entry for the device. The user most likely typed the device mnemonic incorrectly.

BAD -- DISK ATTACH FAILED

The specified disk is currently attached by another task.

Wait until the attaching task releases the disk, then retry the command.

NOTE

If the specified disk is currently attached by another task, there is a good chance the user specified the wrong device mnemonic.

BAD -- DISK ERROR

An unaccountable error occurred on the specified device; most likely a hardware error.

Insure that the device is properly cycled up and on line, then retry the command. If the error reoccurs, notify field service.

BAD -- DISK MOUNTED

The specified disk is logically mounted. That is, a MOU command was issued for the device. BAD cannot process bad blocks for a logically mounted disk.

Execute the DMO command for the device, then retry the command.

BAD -- DISK NOT READY

The specified disk device is inaccessible; probably due to a hardware error.

Insure that the device is properly cycled up and on-line, then retry the command. If the error reoccurs, notify field service.

BAD -- DISK WRITE LOCKED

The specified disk device is inaccessible for write operations.

Set the write enable switch and retry the command.

BAD -- FAILED TO READ MCR COMMAND BUFFER

BAD failed to obtain the command line with which it was invoked. This will occur if BAD is invoked using the RUN command, as opposed to the prescribed method.

Retry the command using the prescribed format. If the failure reoccurs, notify the DEC software specialist.

BAD -- FAILED TO WRITE BAD BLOCK FILE

Either an error occurred when BAD attempted to write the bad block information, or the last 256 blocks of the disk are unusable.

BAD -- ILLEGAL DEVICE

The device mnemonic, specified by the user, is either invalid or specified in a non-standard format.

BAD -- SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

BOO -- ASSIGN LUN ERROR -- CODE -nn

The system is unable to assign a LUN to the device being booted. The reason for the failure is determined by -nn, where -nn is a system standard error code. See Appendix N.

BOO -- CANNOT FIND SYSTEM DEVICE PUD

The system device (SY) PUD cannot be found. This will occur if SY was redirected to itself, or if the PUD entry itself is corrupted.

Redirect SY to a valid system device, then retry the command. If the error reoccurs, call the DEC software specialist.

BOO -- COMMAND I/O ERROR CODE, CODE = -nn

The TTY handler is unable to obtain a command line from the user terminal.

Notify the DEC software specialist.

BOO -- COMMAND SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

BOO -- ERROR CLOSING FILE filename CODE = -nn

The file system is unable to close the specified file. The reason for the failure is determined by -nn, where -nn is a system standard error code. See Appendix N.

BOO -- FILE NOT AN RSX IMAGE

Self-explanatory.

BOO -- FILE NOT CONTIGUOUS

The file is either corrupted, or not an RSX system image file.

BOO -- GET LUN ERROR -- CODE = -nn

The system is unable to get a LUN for the device being booted. The reason for the error is determined by -nn, where -nn is a system standard error code. See Appendix N.

BOO -- INDIRECT COMMAND SYNTAX ERROR

The user specified an indirect command file specifier in a format that does not conform to syntax rules.

Refer to the command description for the correct syntax.

BOO -- INDIRECT FILE DEPTH EXCEEDED

The indirect command file, specified by the user, has an indirect command file specification as one of its commands. BOO can only process one indirect command file at a time.

Recreate the indirect command file, or enter the commands manually.

BOO -- INDIRECT FILE OPEN FAILURE, CODE = -nn

BOO is unable to open the specified indirect command file. The reason for the failure is determined by -nn, where -nn is a system standard error code. See Appendix N.

BOO -- INVALID SWITCH

The user specified a un-recognizable switch. The only switch recognized by BOO is /WB.

BOO -- ILLEGAL DEVICE/VOLUME

The device mnemonic, specified by the user, does not conform to device syntax.

BOO -- ILLEGAL DEVICE/UNIT NUMBER

Either the device mnemonic is illegal; not recognized by the system, or the unit number exceeds the prescribed range for the device.

BOO -- ILLEGAL FILENAME

The user specified a file name that either contains a non-alphanumeric character, or one that is greater than 9 alphanumeric characters in length.

BOO -- ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

Self-explanatory.

BOO -- ILLEGAL GET COMMAND LINE ERROR, CODE -nn

The system, while processing a get command line directive, encountered an unexplainable error condition. Either the system software is corrupted, or the hardware has failed. -nn is the get command line error code.

Notify the DEC software specialist.

BOO -- ILLEGAL UIC

The UFD specified by the user is either illegally formatted or nonexistent.

BOO -- I/O ERROR RSX IMAGE FILE filename, CODE = -nn

The file system encountered an error while reading the specified file. The reason for the failure is determined by -nn, where -nn is a system standard error code. See Appendix N.

BOO -- I/O ERROR (CODE = -nn) WRITING BLOCK 0

Probable hardware error; the QIO issued by BOO was valid, but the write to block 0 failed. The reason for the failure is determined by -nn, where -nn is a system standard error code. See Appendix N.

BOO -- NO WILD CARDS ALLOWED

Self-explanatory.

BOO -- NOT BOOTABLE DEVICE

The specified device is not a mass storage device.

BOO -- ONLY ONE FILESPEC (OUTPUT) ALLOWED

Self-explanatory.

BOO -- OPEN FAILURE FILE filename, CODE = -nn

BOO failed to open the specified file. The reason for the failure is determined by -nn, where -nn is a system standard error code. See Appendix N.

BOO -- QIO FAILURE (CODE = -nn) WRITING BLOCK 0

A QIO issued by BOO was rejected; BOO may be corrupted. The reason for the failure is determined by -nn, where -nn is a system standard error code. See Appendix N.

BOO -- SYSTEM IS NOT QUIESCENT

The user attempted to execute BOO while tasks were still active in the system.

Execute ACT to determine which tasks remain active. Terminate the active tasks or wait for them to finish executing, then retry the BOO command.

CON -- COMMAND SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

CON -- ILLEGAL ERROR/SEVERITY CODE n1 n2 n3

CON failed in its attempt to issue a message to the user terminal. n1 contains the value in R1 at the time of the failure; n2 contains the value in R2; and n3 contains the return address after calling the error message routine.

NOTE

The high order byte in R1 reflects the severity code between 0 and 2.

The low order byte in R1 reflects the error message CON was trying to issue.

Notify the DEC software specialist.

CON -- TASK taskname NOT ACTIVE

or

CON -- TASK taskname NOT IS SYSTEM

Self-explanatory.

CON -- UNABLE TO READ MCR COMMAND LINE

CON failed to obtain the command line with which it was invoked. This will occur if CON is invoked using the RUN command, as opposed to the prescribed method.

Retry the command using the prescribed format. If the failure reoccurs, notify the DEC software specialist.

DMO -- COMMAND I/O ERROR

The DMO task is unable to obtain a command line from the user terminal.

Notify the DEC software specialist.

DMO -- DEVICE NOT IN SYSTEM

The user specified a device that was not included as a usab'e device when the system was generated. That is, there is no PUD entry for the device. The user most likely typed the device mnemonic incorrectly.

DMO -- DEVICE NOT MOUNTABLE

The user specified a device that cannot be mounted; e.g., card reader.

DMO -- HANDLER NOT RESIDENT

The user specified a device for which the appropriate device handler is not resident.

Load the required device handler task and retry the command.

DMO -- ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

Self-explanatory.

DMO -- INVALID UIC

The UIC specification, entered by the user, either does not match the volume's UIC; or it was entered in the wrong format.

DMO -- INVALID VOLUME LABEL

The volume label, specified by the user, does not match the volume's label.

DMO -- SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

DMO -- VOLUME BUSY - TRY AGAIN

There are open files on the volume.

Wait for the files to be properly closed, then retry the command.

DMO -- VOLUME NOT MOUNTED

Self-explanatory.

HEL -- DIRECTIVE FAILURE

An undefinable error occurred during the execution of HEL.

Retry the command. If the error reoccurs, notify the DEC software specialist.

HEL -- ILLEGAL PASSWORD

System access is denied. The specified password does not match the password stored in the requested UFD.

HEL -- READ FAILURE ON DIRECTORY ENTRY

Either the specified UIC does not correspond to a UFD on the system device. Either the system device is not mounted or the hardware has failed.

Notify the system manager.

HEL -- SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

INI -- ALLOCATION FOR SYS FILE EXCEEDS VOLUME LIMIT

Either the user specified /INDX=BLK:n where the value for n was too high; or the user specified /INDX=END and the value specified in /INF=n was too large.

Retry the command with a lower value specified.

INI -- BAD BLOCK HEADER I/O ERROR

INI is unable to retrieve the bad block data area from the specified disk volume.

Retry the command. If the problem persists, perform the following:

1. Rerun BAD

2. Rerun INI with /BAD=[AUTO] specified.

If the problem still persists, the hardware has failed. Notify field service.

INI -- BAD BLOCK FILE FULL - TASK ABORTED

Either the bad block file was full from a previous BAD execution, or the user specified more bad blocks than the bad blocks file is able to store. The disk is not usable.

INI -- BLOCK(S) EXCEED VOLUME LIMIT

The user specified a range of bad blocks that exceed the range of blocks available on the desired volume.

INI -- BOOT BLOCK WRITE ERROR

INI has failed in its attempt to format the disks bootstrap block. One of the following conditions exists:

1. The disk device is off line
2. The bootstrap block (block 0) is bad
3. The device is write locked
4. The hardware has failed.

INI -- CHECKPOINT FILE HEADER I/O ERROR

INI is unable to write the volume's checkpoint file header area. One of the following conditions exists:

1. The device is off line
2. The device is not write enabled
3. The volume has a bad block
4. The hardware has failed.

INI -- DATA ERROR

The user entered bad block data in an incorrect format.

Re-enter the command using the correct format.

INI -- DEVICE NOT IN SYSTEM

The specified device was not included as a usable device when the system was generated. That is, there is no PUD entry for the device. The user most likely typed the device mnemonic incorrectly.

INI -- DEVICE NOT READY

The requested device is not completely cycled up.

Wait for the device to become ready, then retry the command.

INI -- DEVICE WRITE LOCKED

Set the device write enable switch, then retry the command.

INI -- DUPLICATE BLOCK(S) FOUND

The user specified bad block data for blocks that already appear in the bad blocks file.

INI -- FILE CORRUPT -- DATA IGNORED

The volume's bad block data is inconsistent.

Re-execute BAD, then retry the INI command.

INI -- HANDLER NOT RESIDENT

The required device handler is not resident.

Load the required device handler and retry the command.

INI -- HOME BLOCK ALLOCATE WRITE ERROR

INI has failed in its attempt to allocate space for the volume's home block. One of the following conditions exists:

1. The device is off line
2. The device is not write enabled
3. The volume has bad blocks
4. The hardware has failed.

INI -- HOME BLOCK WRITE ERROR

INI has failed in its attempt to write the volume's home block data. One of the following conditions exists.

1. The device is off line
2. The device is not write enabled
3. The volume has bad blocks
4. The hardware has failed.

INI -- INDEX FILE BIT MAP I/O ERROR

INI has failed in its attempt to write the volume's index bit map. One of the following conditions exists:

1. The device is off line
2. The device is not write enabled
3. The volume has bad blocks
4. The hardware has failed.

INI -- INDEX FILE HEADER I/O ERROR

INI has failed in its attempt to write the index file header. One of the following conditions exists:

1. The device is off line
2. The device is not write enabled
3. The volume has bad blocks
4. The hardware has failed.

INI -- ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

or

INI -- ILLEGAL KEYWORD VALUE

Self-explanatory.

INI -- ILLEGAL UIC

The user specified a UIC that either does not conform to UIC syntax, or is not in the range of valid UICs.

INI -- MAGTAPE DEVICE ERROR

The specified magnetic tape device cannot be accessed for one of the following reasons:

1. The device is off line
2. The device is not write enabled
3. The hardware has failed.

INI -- MAGTAPE LABEL MUST BE SPECIFIED

Self-explanatory.

INI -- MAGTAPE WRITE ERROR

INI is unable to initialize a magnetic tape volume for one of the following reasons:

1. The device is off line
2. The device is write protected
3. The hardware has failed.

INI -- MFD FILE HEADER I/O ERROR

INI is unable to write the MFD file header for one of the following reasons:

1. The device is off line
2. The device is not write enabled
3. The volume has bad blocks
4. The hardware has failed.

INI -- MFD WRITE ERROR

INI is unable to write the MFD file for one of the following reasons:

1. The device is off line
2. The device is not write enabled
3. The volume has bad blocks
4. The hardware has failed.

INI -- NOT FILES-11 DEVICE

Self-explanatory.

INI -- NULL FILE HEADER I/O ERROR

The hardware has failed. Notify field service.

INI -- STORAGE BIT MAP FILE HEADER I/O ERROR

INI is unable to format the bit map for one of the following reasons:

1. The device is off line
2. The device is not write enabled
3. The volume has bad blocks
4. The hardware has failed.

INI -- SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

INI -- VOLUME MOUNTED

INI cannot process a logically mounted volume.

If the device mnemonic was correctly specified, execute DMO to dismount the volume, then retry the command; otherwise, retry the command with the device mnemonic correctly specified.

INS or INV -- COMMAND I/O ERROR

INI is unable to get a command line from the user terminal.

Notify the DEC software specialist.

INS or INV -- COMMAND SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

INS -- DATE MISMATCH GLOBAL AREA area-name FILE filename

The task being installed is linked with a global area that is a different version than the one currently installed.

Remove the current version; replace it with the latest version, then retry the command.

INS -- FLAGWORD MISMATCH GLOBAL AREA area-name FILE filename

Either the flag word field of the global area is corrupted, or the corresponding field in the task image is corrupted.

Rebuild.either the offending task or global area, and retry the command.

INS -- GLOBAL AREA ACCESS REQUEST VIOLATION FILE filename

The task being installed is denied access to the specified global area. One of the following conditions exists:

1. The task contains an illegal reference or link to the global area

2. The global area is protected against access.

Rebuild the task correctly and retry the command.

INS -- GLOBAL AREA NAME ALREADY IN USE

The user attempted to install a global area whose name matches one that is already installed.

Either remove the global area already installed, or rebuild the new one, giving it a new name.

INS or INV -- UNKNOWN DEVICE dev: FILE filename

A device assigned to a LUN at task build time is not one that was generated into the system. This is a diagnostic message; the task image was successfully installed. The user must, however, reassign the LUN to a valid device before running the task.

INS or INV -- ILLEGAL DEVICE/VOLUME dev:

The user specified a device that is not recognized by the system. It is either an illegal device, or the device was improperly specified.

INS or INV -- ILLEGAL DIRECTORY

The user specified a UFD in a format that does not conform to syntax rules.

INS or INV -- ILLEGAL ERROR/SEVERITY CODE n1 n2 n3

INS failed in its attempt to issue a message to the user terminal. n1 contains the value in R1 at the time of the failure; n2 contains the value in R2; and n3 contains the return address after calling the error message routine.

NOTE

The high order byte in R1 reflects the severity code between 0 and 2.

The low order byte in R1 reflects the error message CON was trying to issue.

Notify the DEC software specialist.

INS or INV -- ILLEGAL FILENAME filename

The specified filename contains non-alphabetic characters, or it is more than 9 characters in length.

INS or INV -- ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

Self-explanatory.

INS or INV -- ILLEGAL GET COMMAND LINE ERROR CODE -nn

The system, while processing a get command line directive, encountered an unexplainable error. Either the system software is corrupted, or the hardware has failed. -nn is the error return code.

Notify the DEC software specialist.

INS or INV -- ILLEGAL HEADER DATA FILE filename

The task image header block is corrupted; the task is unusable.

Rebuild the task; then retry the command.

INS or INV -- ILLEGAL LABEL DATA FILE filename

The task image contains inconsistent information; the file is corrupted and, therefore, unusable.

Rebuild the task; then retry the command.

INS -- ILLEGAL OPTION ON SHARABLE GLOBAL AREA filename

The user specified the /INC keyword option for a sharable global area.

Retry the command without the /INC switch specified.

INS or INV -- ILLEGAL POOL LIMIT FILE filename

The pool limit established in the task image file header is illegal; either file header is corrupted, or the value specified for the /POOL switch exceeds 255.

Rebuild the task; then retry the command.

INS or INV -- ILLEGAL PRIORITY FILE filename

The task image file header is corrupted; the file is not usable.

Rebuild the file; then retry the command.

INS or INV -- ILLEGAL SWITCH

The user specified a switch that INS does not recognize. Either the switch is not an INS switch, or it was specified in the wrong format.

INS or INV -- ILLEGAL TASK STATUS WORD FILE filename

The task status word of the task image file's header is corrupted; the file is not usable.

Rebuild the file; then retry the command.

INS or INV -- ILLEGAL UIC FILE filename

Either the UIC in the task image file's header is corrupted, or the UIC specified in the INS command line is bad.

Either rebuild the task and retry the command, or retry the command with the UIC correctly specified.

INS or INV -- IMPROPER SIZE DATA FILE filename

The task image file is corrupted.

Rebuild the task, then retry the command.

INS or INV -- INDIRECT COMMAND SYNTAX ERROR

The user entered an indirect command file specifier in a format that does not conform to syntax rules.

Retry the command with the indirect command file correctly specified.

INS or INV -- INDIRECT FILE DEPTH EXCEEDED

The indirect command file, specified by the user, has an indirect command file specification as one of its commands. INS can only process one indirect file at a time.

Recreate the originally specified indirect command file, replacing the erroneous command file specifier with the actual commands it represents.

INS or INV -- INDIRECT FILE OPEN FAILURE filename

The requested indirect command file cannot be located on the specified or implied device.

Insure that the indirect file specifier was correctly entered, and retry the command.

INS or INV -- I/O ERROR TASK IMAGE FILE

INS encountered an error while processing the specified task image file. One of the following conditions exists:

1. The device is off line
2. The device is not write enabled
3. The volume has bad blocks
4. The hardware has failed.

INS -- LENGTH MISMATCH GLOBAL AREA area-name FILE filename

Either the task image file's label block is corrupted, or the GCD in a global area is corrupted.

Rebuild either the task image or global area, then retry the command.

INS or INV -- MEMORY MAPPING CONFLICT FILE filename

Either the task's memory mapping registers or memory limits are invalid.

Rebuild the file, then retry the command.

INS or INV -- NO NODE AVAILABLE FOR FILE filename

The system is saturated with active tasks; there are no pool nodes available to run INS.

Either wait for pool nodes to become available, or notify the system manager.

INS or INV -- NO ROOM IN STD FOR NEW TASK taskname

The number of allowable installed tasks, specified at system generation, has been reached. The named task cannot be installed until some of the previously installed tasks are removed. In the case of INV, the system data base or SGN1/INV have been corrupted.

Notify the system manager, if using INS. If using INV, notify the DEC software specialist.

INS or INV -- NON CONTIGUOUS TASK IMAGE FILE

Either the file is not a task image file, or the file is corrupted.

Either retry the command with the correct file specified; or rebuild the task, then retry the command.

INS or INV -- OPEN FAILURE FILE filename

INS failed to open the specified task image file for an update. One of the following conditions exists:

1. The requested file does not exist in the specified UFD

2. The UFD does not exist

3. The file is protected against write access

4. The volume is not mounted

5. The volume is off line

6. The device is not write enabled

7. The hardware has failed

INS -- SHARABLE GLOBAL AREA HAS NO OWNING UIC filename

All SGAs must be built or installed having a owning UIC.

Either rebuild the SGA specifying a UIC, or retry the install command with the /UIC switch specified.

INS or INV -- SPECIFIED PARTITION IS TOO SMALL FILE filename

Self-explanatory

INS -- STARTING APR MISMATCH GLOBAL AREA area-name FILE filename

Either the APR field in the global area is corrupted, or the corresponding field in the task image is corrupted.

Rebuild either the offending task image or global area and retry the command.

INS or INV -- TASK NAME ALREADY IN USE FILE filename

The user is attempting to install a task whose name is assigned to a previously installed task.

Either remove the previously installed task replacing it with this one, or assign a different name to this task via the /TASK keyword.

INS -- TASK REQUIRES MORE THAN 7 GCD'S

The specified task image is linked to more than 7 GCD'S.

Rebuild the task changing the number of GCD'S to 7 or less.

INS -- TICKS PER SECOND IN SCOM INVALID

This is a system software problem; notify the DEC software specialist.

INS or INV -- TIMEOUT SPECIFIED ON NON ACCOUNTABLE TASK

Self-explanatory. This is a diagnostic message; the install was successfully completed.

INS -- UNDEFINED GLOBAL AREA area-name FILE filename

The specified task is linked to a global area that has not been installed.

Install the required global area, then retry the command.

INS or INV -- UNDEFINED PARTITION FILE filename

The user specified the /PAR switch either at task build time, or at install time. The specified partition does not exist.

Either rebuild the task, or retry the INS command with the partition name correctly specified.

INS or INV -- UNKNOWN KEYWORD IDENTIFIER

Self-explanatory.

INS -- WATCHDOG TIMER OVERFLOW

The value specified in the /TIM switch exceeds 24 hours in hours, minutes, seconds, or ticks.

INV -- CANNOT INSTALL GLOBAL REGIONS

Sharable Global areas cannot be installed during SYSGEN phase 1. Defer installation until phase 2 or later.

INV -- CANNOT INSTALL MU TASKS WITH RO AREAS

The user is attempting to install a multi-user task that has a read only root, which cannot be done during phase 1 of system generation. Defer installation of these tasks until phase 2 or later.

INV -- CANNOT INSTALL TASKS BOUND TO GLOBALS

During phase one of SYSGEN, tasks that are bound to sharable global areas cannot be installed. Defer installation of these tasks until phase 2 or later.

LUN -- CAN'T READ MCR COMMAND BUFFER

LUN failed to obtain the command line with which it was invoked. This will occur if LUN is invoked using the RUN command, as opposed to the prescribed method.

Retry the command using the prescribed format. If the failure reoccurs, notify the DEC software specialist.

LUN -- FATAL I/O ERROR

The device, on which the specified task is installed, cannot be accessed. One of the following conditions exists:

1. The requested file does not exist in the specified UFD

2. The UFD does not exist

3. The file is protected against write access

4. The volume is not mounted

5. The volume is off line

6. The device is not write enabled

7. The hardware has failed

LUN -- NO LUN ASSIGNMENT

or

LUN -- NO LUNS

Self-explanatory.

LUN -- SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

MEM -- GET LINE ERROR

MEM failed to obtain the command line with which it was invoked. This situation can occur if MEM is invoked using the RUN command.

Retry the command by typing the following line.

```
MCR>MEM   options
```

If the failure reoccurs, notify the DEC software specialist.

MEM -- ILLEGAL SWITCH

An illegal switch or a nonexistant device for TI= was specified, or a series of specifications were included with one or more tasknames omitted.

Retry the command.

MEM -- PRIVILEGED USER FUNCTION ONLY

Self-explanatory.

MEM -- taskname SPECIFIED VERSION NOT LOCKED

Either the task specified in the MEM command was not locked in memory or a typing error was made when typing the command.

MEM -- SYNTAX ERROR

Self-explanatory.

MOU -- ACP TASK NOT INSTALLED

Self-explanatory. Install task the desired ACP task and/or verify that the ACP task name was correctly specified, then retry the command.

MOU -- ALL UNITS MUST HAVE SAME NUMBER OF TRACKS

All units dedicated to a magnetic tape volume set must be the same; that is, all 800 BPI or all 1600 BPI.

Retry the command with the correct device types specified.

MOU -- BAD BLOCK ON DEVICE

Either the hardware has failed, or there is an inconsistency with the medium; for example, a seven track tape on a nine track drive.

MOU -- COMMAND I/O ERROR

The TTY handler is unable to obtain a command line from the user's terminal.

Notify the DEC software specialist.

MOU -- DEVICE IS NOT MOUNTABLE

Self-explanatory.

MOU -- DEVICE NOT READY

The requested device is not completely cycled up (in a ready state).

Wait for the device to become ready, then retry the command.

MOU -- DEVICE NOT IN SYSTEM

The specified device was not included as a usable device when the system was generated. That is, there is no PUD entry for the device. The user most likely typed the device mnemonic incorrectly.

MOU -- FILE SYSTEM MESSAGE TASK (F11MSG) MUST BE INSTALLED

Self-explanatory; install task F11MSG, then retry the command.

MOU -- HANDLER TASK NOT RESIDENT

The handler task for the device being mounted is not resident.

Load the required device handler task, then retry the command.

MOU -- OTHER VOLUME STILL ON LINE

The user is attempting to mount a volume on a device that previously contained a volume that was never dismounted. The user most likely removed the volume from the device and replaced it with a new volume, without having issued a DMO command for the previous volume.

Replace the current volume with the previous one. Issue the appropriate DMO command, then replace the dismounted volume with the next volume to be mounted. Retry the MOU command.

MOU -- ILLEGAL FUNCTION FOR NON PRIVILEGED TERMINAL

Self-explanatory.

MOU -- HOME BLOCK NOT FOUND

MOU is unable to obtain the required mount information from the volume's home block. One of the following conditions exist:

1. The volume is not a Files-11 volume

2. The hardware is malfunctioning. For example, the disk drive's read/write heads are misaligned.

MOU -- I/O SYSTEM ERROR

An undefinable error occurred during the execution of MOU.

Retry the command. If the error reoccurs, notify the DEC software specialist.

MOU -- NO SPACE FOR VCB

There is no space available in the system node pool for the volume control block.

Retry the command. If the failure persists, the system should be rebuilt with a larger node pool.

MOU -- PARITY ERROR ON DEVICE

This is a hardware failure; notify field service.

MOU -- SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

MOU -- UNIT DOES NOT SUPPORT 1600 BPI

The user specified 1600 BPI for a tape drive that can only support 800 BPI. MOU defaults to 800 BPI and continues processing the command.

MOU -- VOLUME IS NOT ANSI FORMAT

Either the hardware is malfunctioning or the volume is not an ANSI standard volume. In either case, the volume cannot be mounted.

MOU -- WRONG VOLUME

The volume's label does not match the label specified in the MOU command line. The volume is not mounted.

Retry the command with the correct volume label specified.

OPE -- BYTE ADDRESS

The user is attempting to open an odd numbered address location.

OPE -- CAN'T READ MCR COMMAND BUFFER

OPE failed to obtain the command line with which it was invoked. This will occur if OPE is invoked using the RUN command, as opposed to the prescribed method.

Retry the command using the prescribed format. If the failure reoccurs, notify the DEC software specialist.

OPE -- ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

Self-explanatory.

OPE -- INPUT I/O ERROR

The TTY handler is unable to obtain a command line from the user terminal.

Notify the DEC software specialist.

OPE -- INVALID ADDRESS

The address specified by the user either is not in the range of valid memory addresses, or was incorrectly specified.

OPE -- NUMBER TOO LARGE

Self-explanatory.

OPE -- OUTPUT ERROR

The TTY handler is unable to issue a message to the user's terminal.

Notify the DEC software specialist.

OPE -- PARTITION NOT IN SYSTEM

Self-explanatory.

OPE -- SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

OPE -- TASK NOT FIXED IN CORE

or

OPE -- TASK NOT IN SYSTEM

Self-explanatory.

OPR -- ILLEGAL ERROR CODE RECEIVED FROM SPR2..

An internal error has occurred. The multiple device despooler task has delivered an error code that is not within the range of acceptable error codes.

Notify the DEC software specialist.

OPR -- ILLEGAL SWITCH COMBINATION

The user specified a stop switch that conflicts either with a begin spooling switch, and/or a resume switch, for the same device.

OPR -- INVALID DEVICE SPECIFICATION

The user has specified a device that is not a system generated device or the device specified is not a carriage controlled unit record device; e.g., line printer or teleprinter.

OPR -- INVALID OR NONEXISTENT FORM TYPE SPECIFIED

The user specified a form type that is not in the range of 0 to 6.

OPR -- INVALID SWITCH OR SWITCH VALUE

Self-explanatory.

OPR -- NO COMMAND LINE SPECIFIED

The user typed the OPR command without following it with a command specification.

OPR -- NO DEVICE SPECIFIED

or

OPR -- NO SWITCH SPECIFIED

Self-explanatory.

OPR -- SEND/REQUEST TO SPOOLER TASK FAILED

The send/request directive to either the queue manager task (SPR) or the multiple device despooler task (SPR2) has failed. Either or both of the tasks are not installed.

OPR -- SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

OPR -- UNKNOWN ARGUMENT TO /RS: SWITCH

Self-explanatory.

PWD -- DIRECTIVE FAILURE

An undefinable error occurred during the execution of PWD.

Retry the command. If the error reoccurs, notify the DEC software specialist.

PWD -- NONEXISTENT ACCOUNT

The user is attempting to assign a password to a nonexistent UFD.

PWD -- NON SYSTEM ACCOUNT MAY ONLY MODIFY OWN PASSWORD

Self-explanatory.

PWD -- READ OR WRITE FAILURE ON DIRECTORY ENTRY

One of the following conditions exists:

1. The UFD does not exist

2. The file is protected against write access

3. The volume is not mounted

4. The volume is off line

5. The device is not write enabled

6. The hardware has failed

PWD -- SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

QUE -- DEVICE dev: DOES NOT CONTAIN A DIRECTORY

The user has queued a file to be printed, however the device designated as the input device does not contain a directory. All queue orders up to the device in error have been processed.

Re-enter the command from the point of error specifying a correct input device.

QUE -- DEVICE dev: IS NOT SPOOLABLE

The user has queued a file to be printed onto a device that is not a spoolable device. Spoolable devices are unit-record carriage controlled devices; e.g., line printer or teleprinter.

QUE -- ERROR OPENING filename -nn

An error has occured opening the file designated by "filename". The reason for the failure is determined by -nn, where -nn is a system standard error code. See Appendix N.

All file specifications up to the file in error have been processed. Correct the error and reinput the command from the point of failure.

QUE -- ERROR READING SPRQUEUE.SYS FILE -nn

An error has occurred while reading the queue file. The reason for the error is determined by -nn, where -nn is a system standard error message. See Appendix N.

Notify the system manager.

QUE -- GET COMMAND LINE ERROR -1

An error occurred while reading the indirect command file. The device has cycled down or the file is corrupted.

Obtain a listing of the queue file to determine which commands of the indirect file have been processed. Correct the errant indirect file condition and resume processing from the point of failure.

QUE -- GET COMMAND LINE ERROR -2

An open failure has occurred on the specified indirect command file. One of the following conditions exists:

1. The user directory does not permit read access to the file.
2. There is a problem with the physical device (e.g., device cycled down).
3. The volume is not mounted.
4. The specified file directory does not exist.
5. The file does not exist as specified.

QUE -- GET COMMAND LINE ERROR -3

The user specified an indirect file in a format that does not conform to syntax rules.

QUE -- GET COMMAND LINE ERROR -4

The maximum depth of three indirect files has been exceeded.

Locate the point within the indirect files where the fourth level of indirection has been specified. Delete this level of indirection. All of the commands up to this point have been processed. Specify the fourth level of indirection as a separate command to QUE. Resume processing of the original indirect command files at the command following the fourth level of indirection.

QUE -- NO FILES QUEUED

The user has requested a list of the queue file, but no files are queued.

This message will also occur if the file [1,4]SPRQUEUE.SYS is locked. If the file is locked, notify the system manager who will take the appropriate action.

QUE -- SEQUENCE NUMBER NOT SUPPLIED FOR MODIFY OR KILL

The user has entered a modify or kill command omitting the required sequence number.

QUE -- SYNTAX ERROR [text string in error]

Self-explanatory; refer to the command description for the correct syntax.

If the command was a modify or kill command, obtain a listing of the queued file to determine what part of the command has been processed. Re-enter the modify or kill command from the point of failure, correcting the syntax error. For all other queue commands no processing has occurred. Re-enter the command correctly.

REA -- DEVICE NOT IN SYSTEM

The user specified a device mnemonic that has not been generated into the system.

REA -- FATAL I/O ERROR

REA, either while reading or writing the task header, encountered an error. One of the following conditions exists:

1. The requested file does not exist in the specified UFD

2. The UFD does not exist

3. The file is protected against write access

4. The volume is not mounted

5. The volume is off line

6. The device is not write enabled

REA -- ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

OR

REA -- ILLEGAL LUN NUMBER

Self-explanatory.

REA -- SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

REA -- TASK NOT IN SYSTEM

Self-explanatory.

RED -- CAN'T READ MCR COMMAND BUFFER

REA failed to obtain the command line with which it was invoked. This will occur if RED is invoked using the RUN command, as opposed to the prescribed method of invoking RED.

Retry the command using the prescribed format. If the failure reoccurs, notify the DEC software specialist.

RED -- CIRCULAR REDIRECT CHAIN

The user, through a sequence of Redirect commands, has established a chain of redirection which returns to old-dev.

RED -- DEVICE NOT KNOWN TO SYSTEM

The user specified a device mnemonic that has not been generated into the system.

RED -- HANDLER NOT RESIDENT

This message informs the user that the device handler for the redirected device is not resident. The redirect has been successfully complected.

RED -- ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

Self-explanatory.

RED -- SOURCE DEVICE, OR ITS EXISTING REDIRECT, IS ATTACHED

Self-explanatory; wait until the device becomes unattached, or abort the process that has it attached.

RED -- SOURCE DEVICE, OR ITS EXISTING REDIRECT IS ATTACHED

Self-explanatory; dismount the device and retry the command.

RED -- SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

REM -- COMMAND SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

REM -- FAILED TO OPEN @ FILE

The requested indirect command file cannot be found on the volume indicated. Either the file does not exist of the file was specified incorrectly.

REM -- ILLEGAL ERROR/SEVERITY CODE n1 n2 n3

REM failed in its attempt to issue a message to the user terminal. n1 contains the value in R1 at the time of the failure; n2 contains the value in R2; and n3 contains the return address after calling the error message routine.

NOTE

The high order byte in R1 reflects the severity code between 0 and 2.

The low order byte in R1 reflects the error message REM was trying to issue.

Notify the DEC software specialist.

REM -- ILLEGAL FUNCTION FOR NON-PRIVILEGED USER

Self-explanatory.

REM -- ILLEGAL SWITCH

The user specified a switch that REM does not recognize. Either the switch is not a REM switch, or it was specified in the wrong format.

REM -- INCONSISTENT SWITCH SPECIFIED

The user specified a switch combination that is confusing to REM.

REM -- MAX @ FILE DEPTH EXCEEDED

The indirect command file, specified by the user, has an indirect command file specification as one of its commands. REM can process only one indirect file at a time.

Either recreate the originally specified file replacing the offending indirect file specifier with the commands if represents, or enter the required commands manually.

REM -- TASK taskname FIXED IN MEMORY.

A task that is fixed in memory cannot be removed.

Unfix the task, then retry the command.

REM -- TASK taskname HAS NONZERO POOL USAGE COUNT

The user is attempting to remove a task that has initiated an activity, and that activity is on-going; e.g. initiated an other task, or an I/O operation.

Wait for the initiated activity to complete, then retry the command.

REM -- TASK taskname IS ACTIVE

or

REM -- TASK taskname NOT IN SYSTEM

or

REM -- taskname NOT IS SYSTEM

Self-explanatory

REM -- library-name HAS NON-ZERO INSTALL COUNT

The user is attempting to remove a library that has installed tasks linked to it.

Remove the installed tasks, then retry the command.

RUN -- CAN'T READ MCR COMMAND BUFFER

Run failed to obtain the command line with which it was invoked. This will occur if RUN is invoked using the RUN command, as opposed to the prescribed method.

Retry the command using the prescribed format. If the failure reoccurs, notify the DEC software specialist.

RUN -- HANDLER NOT RESIDENT TO LOAD TASK

The device handler task, required to load the requested from the volume on which it resides, is not in the system.

Load the required handler task, then retry the command.

RUN -- INSUFFICENT POOL NODES

There are not enough available pool nodes to run the desired task.

Either wait for a node to become available, or notify the system manager. If this has been a persistent problem, the system manager may want to reconfigure the system.

RUN -- INVALID KEYWORD

The user specified a keyword that either is not a valid keyword, or was specified in an illegal format.

RUN -- INVALID TIME PARAMETER

The value assigned to the /RSI switch is either invalid, or specified in the wrong format.

RUN -- INVALID UIC

The UFD specified by the user either does not exist, or was specified in an illegal format.

RUN -- INVALID PRIORITY

The priority specified in the /PRI switch either exceeds the maximum allowable (250), or was specified in an illegal format.

RUN -- MEMORY UNAVAILABLE

or

RUN -- PARTITION NOT IN SYSTEM

Self-explanatory.

RUN -- SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

RUN -- TASK ALREADY ACTIVE

or

RUN -- TASK DISABLED

or

RUN -- TASK NOT INSTALLED

Self-explanatory.

RUN -- UNRECOGNIZED DIRECTIVE ERROR

An undefinable error occured during the execution of RUN.

Retry the command. If the error reoccurs, notify the DEC software specialist.

SAV -- *DIAG* SYSTEM CLOCK CHANGED TO A KW11 [L or P]

The system was generated as having a KW11L or KW11P clock. SAV found that the clock present in the system is opposite from the one specified. SAV continues processing and the system is started with the clock found as the system clock.

SAV -- *FATAL* BAD BOOT IN LOW MEMORY

The first 32-word block of memory has been corrupted; most likely by hardware. One of the following conditions exists:

1. The bootstrap routine was written by a user who failed to adhere to the rules.

2. The hardware failed.

3. Someone or some running task corrupted real address 0.

SAV -- *FATAL* FAILED TO READ MCR BUFFER

SAV failed to obtain the command line with which it was invoked. This will occur if SAV is invoked using the RUN command, as opposed to the prescribed method.

Retry the command using the prescribed format. If the failure persists, notify the DEC software specialist.

SAV -- *FATAL* ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

Self-explanatory.

SAV -- *FATAL* dev: IS STILL LOGGED ON

The user attempted to execute SAV with the specified terminal(s) still logged on.

Either re-enter the command with the /LOG switch specified, or log the terminals off, and retry the command.

SAV -- *FATAL* dev: IS STILL MOUNTED

The user attempted to execute SAV with the specified device(s) still mounted.

Either re-enter the command with the /MOU switch specified, or dismount the devices and retry the command.

SAV -- *FATAL* NO SYSTEM CLOCK RESPONSE, RSX-11D CANNOT RUN

Neither the line frequency clock nor the programable clock will respond to SAV. Either there is no clock on the system, or the clock does not work.

Notify your field service engineer.

SAV -- *FATAL* RO ROOT OF taskname IS LOADED BEYOND END OF SAVE FILE

The read-only root of the named multiuser task is either active or fixed at an address beyond that which is to be saved. All active versions of the task must be deactivated to unload the read-only root of the task.

SAV -- *FATAL* RO ROOT OF taskname IS LOADING

The Executive has initiated the I/O operations to load the read-only root of a multiuser task. This is a separate operation from that used for the impure copies of a multiuser task.

SAV -- *FATAL* SHAREABLE GLOBAL sga IS LOADED BEYOND END OF SAVE FILE

The named SGA is either active or fixed at an address beyond that which is to be saved. One of the following steps must be performed:

1. Unload all active or fixed tasks that use the SGA to allow the SGA to be unloaded.

2. Reload the SGA at a lower addresss.

SAV -- *FATAL* SHAREABLE GLOBAL sganame IS LOADING OR EXITING

The Executive has initiated the I/O operations to load an SGA or write a read/write SGA to disk. The SGA has one of the following states: load request pending (GS.LRP), load request queued (GS.LRQ), and record request queued (GS.RRQ).

The system is not quiescent and a SAVE cannot be performed until these states have disappeared.

SAV -- *FATAL* SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

SAV -- *FATAL* SYSTEM HAS INSUFFICIENT MEMORY
PARTITION parname WAS OCCUPIED BEYOND AVAILABLE MEMORY
THE SYSTEM IS NOT RUNNABLE

This situation is caused by an attempt to truncate a system in which some entity (task or SGA) was loaded beyond the available memory.

This error can be caused by bad memory that results in SAVE'S assuming that such memory does not exist. If so, call your field service engineer. If not, the newly booted system was saved too large to run on this configuration.

SAV -- *FATAL* TASK taskname HAS I/O IN PROGRESS

The named task has a nonzero I/O in progress count; therefore, the system is not quiescent. Wait for the I/O to complete or abort the task.

SAV -- *FATAL* TASK taskname IS LOADED BEYOND END OF SAVE FILE

The named task is either active or fixed at an address beyond that which is to be saved. One of the following must be done:

1. Unfix the task, if it is fixed,

2. Unload the handler, if the task is a handler, and load it at a lower address,

3. Terminate the active task and perhaps run it at a lower address,

4. Wait for the task to terminate normally.

SAV -- *FATAL* TASK taskname IS LOADING OR CHECKPOINTING

The Executive has initiated the I/O operations to load or checkpoint the named task. The task has one of the following states: load request pending (TS.LRP), load request queued (TS.LRQ),

The system is not quiescent and a SAVE cannot be performed until these states have disappeared.

SAV -- *FATAL* 11/40 TRAP ECO (KT11-D) NECESSARY TO RUN RSX-11D

SAVE detected that a PDP-11/40 hardware modification (ECO) necessary for operation of RSX-11D is either not present or is failing to operate correctly.

Notify your field service engineer.

SAV -- PARTITION parname EXPANDED BY nnnn*32 (DEC) WORD BLOCKS

SAVE found more memory than was specified at the last save of the system or at system generation if no previous save occurred. The extra amount of memory is nnnn (decimal) 32-word blocks. The total memory size has been printed in a previous message. The named partition has been expanded to use the extra memory.

SAV -- PARTITION parname TRUNCATED BY nnnn*32 (DEC) WORD BLOCKS

SAVE found less memory than was specified at the last save or at system generation if no previous save occurred. The difference in memory size is indicated by nnnn (decimal) 32-word blocks. The total memory size has been printed in a previous message. The named partition has been truncated to fix the memory available.

```
SAV -- THE FOLLOWING EMPTY PATRITION(S) TRUNCATED TO ZERO SIZE:
            parname1
            parname2
               .
               .
               .
```

The system has been successfully reduced in size but all of the named partitions are completely beyond the available memory. Any attempt to run a task already installed in one of these partitions causes the task to be queued on that partition's memory required list (MRL). The task remains on the MRL until it is aborted. Any attempt to install a task in such a partition results in an INS error message to the effect that the task is larger than the partition.

SET -- CAN'T READ MCR COMMAND BUFFER

SET failed to obtain the command line with which it was invoked. This will occur if SET is invoked using the RUN command, as opposed to the prescribed method.

Retry the command using the prescribed format. If the failure reoccurs, notify the DEC software specialist.

SET -- CAN'T SET CHARACTERISTICS FOR THIS TYPE TERMINAL

Self-explanatory; the characteristics specified are incompatible.

SET -- ILLEGAL CACHE VALUE

A number that was not in the range 1 through 32,767 (decimal) was specified in the /CAC=n switch.

Retry the command using a number within the specified range.

SET -- ILLEGAL FUNCTION FOR MACHINE TYPE

The /CAC switch was specified while running on a computer other than a PDP-11/70.

SET -- ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

or

SET -- PARAMETER ERROR - BAD DEVICE NAME

or

SET -- SPECIFIED DEVICE IS NOT A TERMINAL TYPE DEVICE

OR

SET -- SPOOLING NOT SUPPORTED IN THIS RELEASE FOR THIS DEVICE

Self-explanatory.

SET -- SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

SPR -- DEVICE dev: NOT FOUND IN PUD TABLE

A file has been queued via a PRINT$ macro to a nonexistent system device. The queue entry is automatically dequeued.

SPR -- ERROR ISSUING SEND REQUEST TO SPR2.. -nn

An error occurred when SPR issued a Send/Request directive to the despooler task (SPR2..). The reason for the error is determined by -nn, where -nn is the directive status word (DSW) error code.

SPR -- ERROR ON RECEIVE -nn

An error occurred while SPR was receiving a Send/Request directive. The reason for the error is determined by -nn, where -nn is the directive status word (DSW) error code.

```
SPR -- ERROR READING SPRQUEUE.SYS FILE -nn
                     or
       ERROR WRITING SPRQUEUE.SYS FILE -nn
                     or
       ERROR OPENING SPRQUEUE.SYS FILE -nn
                     or
       ERROR CLOSING SPRQUEUE.SYS FILE -nn
```

An error occurred when SPR attempted to access the queue file. The reason for the error is determined by -nn, where -nn is the directive status word (DSW) error code.

NOTE

If the error is persistent, the queue file is corrupted. Perform the queue error procedure documented in the RSX-11D System Managers Guide.

SPR -- INVALID SEQUENCE NUMBER

The sequence number for a kill or modify command does not correspond to any sequence number in the queue.

Retry the command with the correct sequence number specified.

SPR -- SPRQUEUE.SYS FILE CAPACITY EXCEEDED

The SPRQUEUE.SYS file has a capacity of 2047 queue orders. This capacity has been exceeded, and the last order has not been queued.

Avoid adding more orders to be despooled. Allow the despooler task to process sufficient orders to create space in the queue. Then, retry the command.

SPR2.. -- DEVICE dev:, FAILED TO OPEN FILE

The despooler was unable to open an output file on device (dev:). The open order is automatically aborted by the despooler. Either the file to be opened was deleted after a request to print it was queued, or the volume on which the file resides was dismounted.

SPR2.. -- DEVICE dev: TIMED OUT

Three minutes have elapsed since a QIO output order was issued to device (dev:). The multiple device despooler has received no notification from the handler that the order has completed. This error normally occurs when the line printer is not in a ready state, and will be accompanied by a not ready message issued by the device handler. This message requires no user action.

SPR2.. -- GET$ ERROR - DEVICE dev: ERROR CODE -nn

An input error occurred on the device that contains the file being transfered to device dev:. The reason for the error is determined by -nn, where -nn is a system standard error code. See Appendix N. This message does not require user action.

SPR2.. -- GET$ ERROR - DEVICE dev: - ERROR CODE 40

The despooler encountered an input record size, on device dev:, that is greater than 256 bytes. Printing of the input file is automatically aborted.

See Appendix N.

SPR2.. -- INTERNAL ERROR - BAD SEND/REQUEST FROM ...OPR

OPR, as a result of an OPR command issued by the user, issued a send/request directive to the despooler. The request code does not match the appropriate send/request assigned codes.

Document the occurrence along with the OPR command issued, and notify the DEC software specialist.

SPR2.. -- INTERNAL ERROR -- DUPLICATE DEVICES ON DEVICE dev:

The queue manager task directed the despooler task to transfer data to a device (dev:) that is currently being used. Perform the corrective action detailed in the RSX-11D System Managers Guide.

SPR2.. -- INTERNAL ERROR -- FAILED TO ASSIGN LUN

The despooler task was unable to assign the LUN for a given device. This message will be issued as a result of a system generation in which the required device was deleted.

Perform the corrective action detailed in the RSX-11D System Manager's Guide.

SPR2.. -- INTERNAL ERROR -- TOO MANY SIMULTANEOUS DEVICES

The despooler task notified the queue manager task that it is capable of processing n devices simultaneously. The queue manager task, however, delivered n+1 process orders.

Perform the corrective action detailed in the RSX-11D System Manager's Guide.

SPR2.. -- INTERNAL ERROR - ZERO COPIES SPECIFIED

The queue manager task requested a despooler operation on a file, but it also requested that no copies are to be printed. The queue file may be corrupt.

Perform the corrective action detailed in the RSX-11D System Manager's Guide.

SPR2.. -- I/O ERROR ON DEVICE dev: - ERROR CODE -nn

An I/O error occurred during an output operation on device (dev:). The reason for the error is determined by -nn, where -nn is a system standard error code. See Appendix N. This message does not require user action.

SPR2.. -- QIOW$ FAILURE ON DEVICE dev: DSW=-nn - DEVICE IN WAIT

A QIO directive error occurred on device dev:. The reason for the error is determined by -nn, where -nn is the directive status word (DSW) error code.

Correct the error condition, then issue the appropriate OPR command:

OPR dev:/RS	to continue
OPR dev:/AB	to abort the operation

SPR2.. - SEND/REQUEST TO SPR... FAILED

Task SPR... is not installed; install the task.

SYS -- ILLEGAL SWITCH

The user specified a switch that SYS does not recognize. Either the switch is not a SYS switch, or it was specified in the wrong format.

SYS -- SYNTAX ERROR

The user entered a SYS command in a format that does not conform to syntax rules.

TIM -- CAN'T READ MCR COMMAND BUFFER

TIM failed to obtain the command line with which it was invoked. This will occur if TIM is invoked using the RUN command, as opposed to the prescribed method.

Retry the command using the prescribed format. If the failure reoccurs, notify the DEC software specialist.

TIM -- ILLEGAL FUNCTION FOR NON-PRIVILEGED USER

A non-privileged user is not allowed to modify the time or date.

TIM -- SYNTAX ERROR

Self-explanatory; refer to the command description of the correct syntax.

UFD -- DEVICE NOT IN SYSTEM

The user specified a device mnemonic that has not been generated into the system.

UFD -- DIRECTORY ALREADY EXISTS

Self-explanatory.

UFD -- FAILED TO CREATE DIRECTORY

The device is write locked; write enable it and retry the command.

UFD -- FAILED TO ENTER IN MFD

The UFD command task is unable to establish a pointer in the master file directory to the newly created UFD.

Either the hardware has failed, or the master file directory is corrupted.

UFD -- ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

or

UFD -- NOT FILES-11 DEVICE

Self-explanatory.

UFD -- PRIVILEGE VIOLATION ON MFD

The UFD command task, by virtue of the UIC under which it is running, is not allowed access to the volume MFD.

UFD -- SYNTAX ERROR

Self-explanatory; refer to the command description of the correct syntax.

UFD -- VOLUME NOT MOUNTED

Self-explanatory.

UFD -- WRITE ATTRIBUTES FAILED

The hardware has malfunctioned; notify field service.

UNL -- CAN'T READ MCR COMMAND BUFFER

UNL failed to obtain the command line with which it was invoked. This will occur if UNL is invoked using the Run command, as opposed to the prescribed method.

Retry the command using the prescribed format. If the failure reoccurs, notify the DEC software specialist.

UNL -- HANDLER NOT RESIDENT

or

UNL -- ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

Self-explanatory.

UNL -- NODE FOR EXIT UNAVAILABLE

A required pool node is unavailable; the system is saturated with active tasks.

Either wait for a node to become available or notify the system manager. The system manager may want to reconfigure the system.

UNL -- SYNTAX ERROR

Self-explanatory; refer to the command description for the correct syntax.

UNL -- TASK NOT IN SYSTEM

Self-explanatory.

A.2.1 Messages from Task MCRERR

Nine of the RSX-11D MCR functions are performed by a single multifunction task named MFT:

ABO	(Abort task)
ALT	(Alter priority)
CAN	(Cancel request)
FIX	(Fix task in memory)
UNF	(Unfix task)
DIS	(Disable task)
ENA	(Enable task)
LOA	(Load handler task)
RES	(Resume task execution)

MCRERR is the error handling task for MFT. The advantage of MCRERR is that it needs to be resident only when error handling is required. The messages issued by MCRERR are described below.

Each message consists of two lines. The first line printed is the command that caused the error. The second line is the actual error message.

Message descriptions are followed in the manual by a message number. If MCRERR is not installed, this number is printed in the following format on the terminal instead of the message.

```
xxx -- ERRnn
```

xxx = the function called, e.g., LOA.

nn = the message number.

CAN'T READ MCR COMMAND BUFFER 9

The requested function failed to obtain the command line with which it was invoked. This occurs when the function is requested using the RUN command as opposed to the prescribed method.

Retry the command using the prescribed format. If the failure reoccurs, notify the DEC software specialist.

DEVICE NOT IN SYSTEM 3

The specified device was not defined during system generation. The user most likely typed the device mnemonic incorrectly.

HANDLER NOT RESIDENT 17

The user specified a device for which the required handler is not resident.

Load the required device handler task and retry the command.

ILLEGAL FUNCTION OR SWITCH FOR NON-PRIVILEGED TERMINAL 2

Self-explanatory.

ILLEGAL SWITCH OR SWITCH COMBINATION FOR THIS FUNCTION 24

Self-explanatory; refer to the description of the command.

INSUFFICIENT POOL NODES TO PERFORM RQST 25

Required pool nodes are unavailable; the system is saturated with active tasks.

Either wait for nodes to become available, or notify the system manager. The system manager may want to reconfigure the system.

INVALID KEYWORD SPECIFIED 32

Self-explanatory.

INVALID PRIORITY 29

Self-explanatory.

INVALID TIME PARAMETER SPECIFIED 31

Self-explanatory.

INVALID UIC SPECIFIED 27

Self-explanatory.

MEMORY UNAVAILABLE 16

Self explanatory; remove unnecessary tasks, or wait for memory to become available and then retry the command.

NODE FOR FIXING UNAVAILABLE 20

A required pool node is unavailable; the system is saturated with active tasks.

Either wait for a node to become available, or notify the system manager. The system manager may want to reconfigure the system.

NOT ALLOWED IN TIME-SCHEDULED PARTITION 33

Self-explanatory.

NUMERIC CONVERSION ERROR 30

Self-explantory.

PARTITION TOO SMALL FOR TASK 26

Self-explanatory; install the task in a larger partition or reduce the size of the task.

PLEASE LOGON WITH "[UIC]" 8

The user has not logged onto the system using the HELLO command.

Log on and retry the command.

SPECIFIED PARTITION DOES NOT EXIST 28

Self-explanatory; if the error was not caused by mistyping the partition name, notify the system manager.

SYNTAX ERROR 1

Self-explanatory.

TASK ALREADY FIXED 21

Self-explanatory.

TASK ALREADY (OR STILL) ACTIVE 18

Self-explanatory

TASK CAN'T BE DISABLED 12

The task is not enabled.

TASK CHECKPOINTABLE 15

The task is checkpointable and, therefore, cannot be fixed in memory.

TASK DISABLED 14

The specified task must be enabled before it can be fixed.

TASK EXITING 10

The requested function cannot be performed because the specified task is in the process of exiting from the system.

TASK LOADING OR EXITING 5

The requested function cannot be performed because the specified task is either being loaded or is exiting.

TASK NOT ABORTABLE 6

An attempt was made to abort a task that was task built as not abortable using the /-AB switch. This task cannot be aborted.

TASK NOT ACTIVE 7

The requested function cannot be performed because the specified task is not active.

TASK NOT DISABLED 13

Self-explanatory.

TASK NOT ENABLED 11

Self-explanatory.

TASK NOT FIXABLE 19

The specified task was not declared fixable at task build time. The task must be built with the /FX switch specified.

TASK NOT FIXED 23

Self-explanatory.

TASK NOT IN SYSTEM 4

Self-explanatory.

TASK NOT SUSPENDED 22

Self-explanatory.

UNKNOWN DIRECTIVE ERROR 0

An unidentifiable error occurred during execution of the requested MFT function.

Retry the command. If the error reoccurs, notify the DEC software specialist.

APPENDIX B

PIP COMMAND SUMMARY AND ERROR MESSAGES

B.1 PIP COMMAND SUMMARY

APPEND

```
   outfile=infile-1                         Opens an existing file
     [,infile-2,...,infile-n]/AP[/FO]       (outfile) and merges
                                            the input file(s)
     where /FO is File Owner                onto the end of it.
```

COPY

```
   outfile[/switch]=infile-1                Creates a copy of a
     [,infile-2,...,infile-n]               file on the same or
     [/switch]                              another device.
   /switch=BL:n[.]  Block allocated.
           CO       Contiguous output
           -CO      Non-contiguous output
           FO       File Owner
           NV       New version             See Table 2-3 for a
           SU       Supersede               complete description
                                            of these switches
```

DEFAULT

```
   dev:[ufd]/DF                             Changes PIP's default
                                            device and/or ufd.
```

DELETE

```
   infile-1[,infile-2,...,infile-n]         Deletes files,
     /DE
```

ENTER

```
   outfile=infile-1                         Enters a pseudonym for
     [,infile-2,...,infile-n]/EN[/NV]       a file in a directory
     where /NV is New Version               with an option to
                                            force the version
                                            number of "outfile" to be one
                                            greater than the latest
                                            version for the file.
```

FREE

```
   dev:/FR                                  Prints out the available
                                            space on a volume.
```

IDENTIFY

```
   /ID
```

Causes the version of PIP currently in use to be displayed on the terminal.

LIST

```
(1) listfile=infile-1[,...,infile-n]/LI
                   or
(2) infile-1[,...,infile-n]/LI
```

where TI: is listfile in format 2

Lists one or more directories with an option to specify directory listing formats.

Alternate Mode Switches

```
    /BR    Brief Format
    /FU[n] Full Format
    /TB    Total Blocks Format
```

For a complete description of these switches, see the <u>RSX-11D Utility Programs Procedures Manual</u>.

PROTECT

```
   Infile-1[/PR[/SY[:RWED][/OW[:RWED]]

     [/GR[:RWED]][/WO[:RWED]]/FO
```

where SY is system access rights.
OW is member access rights.
GR is group access rights.
WO is world access rights.
RWED is read, write, extend, delete privilege.
FO is File Owner subswitch

PURGE

```
   infile-1[,infile-2,...,infile-n]/PU[:n]
```

Deletes a specified range of obsolete versions of a file.

REMOVE

```
   infile-1[,infile-2,...,infile-n]/RM
```

Removes an entry from a directory file.

RENAME

```
   outfile=infile-1
   [,infile-2,...,infile-n]/RE
   [/NV]
```

where NV is New Version.

Changes the name of a file with an option to force the version number of "outfile" to be one greater than the latest version for the file.

SPOOL

```
   infile-1[,infile-2,...,infile-n]
     /SP
```

Specifies a list of files to be printed and then deleted.

UNLOCK

```
   infile-1[,infile-2,...,infile-n]
       /UN
```

Unlocks a file which was locked as a result of being improperly closed.

UPDATE

outfile=infile-1 [,infile-2,...,infile-n]/UP[/FO] where FO is File Owner	Opens an existing file(s) (infile) and writes it, from the beginning, onto outfile.

B.2 PIP ERROR MESSAGES

Errors encountered by PIP, during processing, are reported to the user in the following format:

```
PIP --<main error message>
<filename or filespec>-<secondary error message>
```

The filename or filespec, if present, identifies the file or set of files being processed when the error occurred. If the error was detected or caused by the operating system, file system, or device handler, the secondary error message is included to explain the cause of the error.

PIP error messages are contained in message files on the system device. If PIP cannot access the message files, because the system device (SY:) is not mounted or MO is not loaded, errors are reported in the following format:

```
PIP --ERROR CODE nn.
<filename or filespec>-<HANDLER CODE -mm.>

                                 or

                           <QIO ERROR CODE -qq.>
```

where

nn — is one of the PIP error codes contained in Table 2-5.

-mm — is one of the system standard, file primitive, or file control service codes listed in Appendix I of the IAS/RSX-11 I/O Operations Reference Manual.

-qq — is one of the directive error codes listed in Appendix I of the IAS/RSX-11 I/O Operations Reference Manual.

The PIP error messages, their description and suggested user actions are as follows.

PIP -- ALLOCATION FAILURE - NO CONTIGUOUS SPACE

Contiguous space available on the output volume is insufficient for the file being copied.

Delete contiguous files that are no longer required on the output volume, and retry the command.

PIP -- ALLOCATION FAILURE ON OUTPUT FILE

or

PIP -- ALLOCATION FAILURE - NO SPACE AVAILABLE

Space available on the output volume is insufficient for the file being copied.

Delete files that are no longer required on the output volume, and reenter the command.

PIP -- BAD USE OF WILD CARDS IN DESTINATION FILE NAME

The user has specified a wild card "*" for an output file name where use of a wild card is explicitly disallowed.

Reenter the command with the proper output file explicitly specified.

PIP -- CANNOT FIND DIRECTORY FILE

The user has specified a ufd that does not exist on the specific volume.

Reenter the command specifying the correct ufd.

PIP -- CANNOT FIND FILE(S)

The file(s) specified in the command were not found in the designated UFD and device.

Check the filespec and retry.

PIP -- CANNOT SPOOL BY FILE ID

The /FI switch cannot be used with the spool command.

Retry command with appropriate filename.

PIP -- CANNOT RENAME FROM ONE DEVICE TO ANOTHER

The user has attempted to rename a file across devices.

Reenter the command renaming the file on the input volume; then, enter another command to transfer the file to the originally intended volume.

PIP -- CLOSE FAILURE ON INPUT FILE

or

PIP -- CLOSE FAILURE ON OUTPUT FILE

For some reason, the input or output file cannot be properly closed. The file will be locked to indicate possible corruption.

Retry the command. If the error reoccurs, run the verify utility VFY, against the volume in question to determine if it is corrupted. VFY is described in Chapter 9.

PIP -- COMMAND SYNTAX ERROR

The user has entered a command in a format that does not conform to syntax rules.

Reenter the command with the correct syntax.

PIP -- DEVICE NOT MOUNTED

The message is self-explanatory.

Mount the device, and retry the command.

PIP -- DIRECTORY WRITE PROTECTED

PIP could not remove an entry from a directory because the unit was write protected, or because of privilege violations.

Write enable the unit, or have the owner of the directory change its protection.

PIP -- ERROR FROM PARSE

The specified directory file does not exist.

Reenter the command with the correct ufd specified.

PIP -- FAILED TO ATTACH OUTPUT DEVICE

An attempt to attach a record-oriented output device has failed. This is usually caused by the device handler not being resident.

Load the device handler and reenter the command.

PIP -- FAILED TO DELETE FILE

or

PIP -- FAILED TO MARK FILE FOR DELETE

The user has attempted to delete a protected file.

Request PIP under the correct UIC and reenter the command.

PIP -- FAILED TO DETACH OUTPUT DEVICE

Occurs if device handler is not loaded or if some system failure occurs.

Verify that the device handler is loaded and retry. If problem persists, contact software support.

PIP -- FAILED TO ENTER NEW FILE NAME

The user has specified a file that already exists in the directory file, or the user does not have the necessary privileges to make entries in the specified directory file.

Reenter the command insuring that the file name and ufd are specified correctly or request PIP under the correct UIC and reenter the command.

PIP -- FAILED TO FIND FILE(S)

The file(s) specified in the command were not found in the designated UFD and device.

Check the filespec and retry.

PIP -- FAILED TO GET TIME PARAMETERS

An internal system failure occurred while PIP was trying to obtain the current date and time.

Retry command. If the problem persists, consult software support.

PIP -- FAILED TO OPEN STORAGE BITMAP FILE

PIP could not read the specified volume's storage bit map, usually because of a privilege violation.

Retry by running PIP under a system UIC, or have the system manager change the protection on the storage bit map.

PIP -- FAILED TO READ ATTRIBUTES

The user's volume is corrupted or the user does not have the necessary privileges to access this file.

Ensure that PIP is running under the correct UIC. If the UIC is correct, then run the verify utility, VFY, against the volume in question to determine where and to what extent the volume is corrupted.

PIP -- FAILED TO REMOVE DIRECTORY ENTRY

PIP could not remove an entry from a directory because the unit was write protected, or because of privilege violations.

Write enable the unit, or have the owner of the directory change its protection.

PIP -- FILE IS LOST

PIP has removed a file from its directory, failed to delete it, and failed to restore the directory entry.

Run the lost file scan in VFY to recover the file name.

PIP -- FAILED TO SPOOL FILE FOR PRINTING

Insufficient system dynamic memory is available, or there are too many files in the spooler queue.

Wait and reenter command again.

PIP -- FAILED TO WRITE ATTRIBUTES

The user volume is corrupted or the user does not have the necessary privileges to write the file.

Ensure that PIP is running under the correct UIC. If the UIC is correct, then run the verify utility, VFY, against the volume in question to determine where and to what extent the volume is corrupted. VFY is described in Chapter 9.

PIP -- FILE NOT LOCKED

The user issued an unlock command for a file that was not locked.

Reenter the command specifying the correct file.

PIP -- ILLEGAL COMMAND

The user has entered a command that is not recognized by PIP.

Reenter the command with the PIP command correctly specified.

PIP -- ILLEGAL SWITCH

The user has specified a switch that is not a legal PIP switch.

Reenter the command with the correct switch specified.

PIP -- ILLEGAL "*" COPY TO SAME DEVICE AND DIRECTORY

The user has attempted to copy all versions of a file into the same directory that is being scanned for input files. This would result in an infinite number of copies of the same file.

Reenter the command renaming the files, or copying them into a different directory.

PIP -- ILLEGAL USE OF WILD CARD VERSION

The use of a wild card version number in the attempted operation would result in inconsistent or unpredictable output.

Retry the command with different options or with explicit or default version number.

PIP -- I/O ERROR ON INPUT FILE

or

PIP -- I/O ERROR ON OUTPUT FILE

One of the following conditions exists:

1. The device is not on line,
2. The device is not mounted,
3. The hardware has failed.
4. The volume is full

1. Determine which condition exists.
2. Rectify the condition.
3. Reenter the command.

PIP -- EXPLICIT OUTPUT FILE NAME REQUIRED

This message is self explanatory.

Reenter the command with the output file name explicitly specified.

PIP -- NOT A DIRECTORY DEVICE

The user has issued a directory-oriented command to a device (such as a printer) that does not have directories.

Reenter the command without specifying a ufd

PIP -- NOT ENOUGH BUFFER SPACE AVAILABLE

PIP has insufficient I/O buffer space to perform the requested command.

Have the system manager rebuild PIP with more buffer space.

PIP -- NO SUCH FILE(S)

The file(s) specified in the command were not found in the designated UFD and device.

Check the filespec and retry.

PIP -- ONLY [*,*] IS LEGAL AS DESTINATION UIC

The user has specified a ufd other than [*,*] as the output file UIC for a copy.

Reenter the command line with [*,*] specified as the output ufd.

PIP -- OPEN FAILURE ON INPUT FILE

or

PIP -- OPEN FAILURE ON OUTPUT FILE

An I/O or file system error has occurred (i.e., a privilege violation, an attempted write to a locked unit, parity error, etc.).

Determine the cause of the problem, rectify or circumvent the problem, and retry command.

PIP -- OUTPUT FILE ALREADY EXISTS-NOT SUPERSEDED

An output file of the same name, type, and version as the file already exists.

Retry the copy with the /NV switch to assign a new version number or the /SU switch to supersede the output file.

PIP -- TOO MANY COMMAND SWITCHES - AMBIGUOUS

The user has specified too many switches, or the switches conflict.

Reenter the command specifying the correct set of switches.

PIP -- VERSION MUST BE EXPLICIT OR "*"

The version number of the specified file must be expressed explicitly, or as a wild card "*".

Reenter the command with the version number correctly expressed.

B.2.1 PIP Error Codes

Table B-1 identifies the error codes PIP issues when it doesn't have access to the message files. Message descriptions and suggested user actions are identical to the information contained in Section B.2.

Table B-1
PIP Error Codes

Error Code	PIP Error Message is:
1.	COMMAND SYNTAX ERROR
2.	ILLEGAL SWITCH
3.	TOO MANY COMMAND SWITCHES - AMBIGUOUS
4.	ONLY [*,*] IS LEGAL AS DESTINATION UIC
5.	ILLEGAL COMMAND
6.	ILLEGAL "*" COPY TO SAME DEVICE AND DIRECTORY
7.	BAD USE OF WILD CARDS IN DESTINATION FILE NAME
8.	EXPLICIT OUTPUT FILE NAME REQUIRED
9.	ALLOCATION FAILURE - NO CONTIGUOUS SPACE
10.	ALLOCATION FAILURE - NO SPACE AVAILABLE
11.	ALLOCATION FAILURE ON OUTPUT FILE
12.	I/O ERROR ON INPUT FILE
13.	I/O ERROR ON OUTPUT FILE
14.	ILLEGAL USE OF WILD CARD VERSION
15.	OPEN FAILURE ON INPUT FILE
16.	OPEN FAILURE ON OUTPUT FILE
17.	CLOSE FAILURE ON INPUT FILE
18.	CLOSE FAILURE ON OUTPUT FILE
19.	FAILED TO DETACH OUTPUT DEVICE
20.	DEVICE NOT MOUNTED
21.	OUTPUT FILE ALREADY EXISTS - NOT SUPERSEDED
22.	FAILED TO MARK FILE FOR DELETE
23.	FILE IS LOST
24.	VERSION MUST BE EXPLICIT OR "*"
25.	ERROR FROM PARSE
26.	FAILED TO DELETE FILE
27.	CANNOT FIND DIRECTORY FILE
28.	FAILED TO ATTACH OUTPUT DEVICE
29.	FAILED TO GET TIME PARAMETERS
30.	NOT A DIRECTORY DEVICE
31.	FAILED TO WRITE ATTRIBUTES
32.	FAILED TO READ ATTRIBUTES
33.	FILE NOT LOCKED
34.	FAILED TO ENTER NEW FILE NAME
35.	FAILED TO RESTORE ORIGINAL DIRECTORY ENTRY - FILE IS LOST
36.	CANNOT RENAME FROM ONE DEVICE TO ANOTHER
37.	FAILED TO SPOOL FILE FOR PRINTING
38.	CANNOT SPOOL BY FILE ID
39.	FAILED TO OPEN STORAGE BITMAP FILE
40.	FAILED TO FIND FILE(S)
41.	CANNOT FIND FILE(S)
42.	NO SUCH FILE(S)
43.	FAILED TO REMOVE DIRECTORY ENTRY
44.	DIRECTORY WRITE PROTECTED
45.	NOT ENOUGH BUFFER SPACE AVAILABLE

APPENDIX C

FLX COMMAND SUMMARY AND ERROR MESSAGES

C.1 FLX COMMAND SUMMARY

The FLX utility has only one command.

outfile=infile-1[,infile-2,...,infile-n] /switch	Performs file conversion between DOS-11, RT-11 and Files-11 formats.
where switch = BL:n	Indicates the number of contiguous blocks to be allocated to the output file.
BS:n	Specifies the block size for cassette tape output.
CO	Indicates that the output file is to be contiguous.
DE	Deletes files from a DOS-11, RT-11, DECtape or disk.
DI	Causes a directory listing of DOS volumes; or DOS or RSX cassette tape volumes to be listed on TI:.
DO	Identifies the file as a DOS-11 formatted file
FA:n	Formatted ASCII.
FB:n	Formatted binary.
FC	Indicates that FORTRAN carriage control conventions are to be used.

ID	Requests the current version number of FLX.
IM:n	Image mode.
LI	Same as DI.
NU:n	Used with /ZE and /RT switches to specify the number of directory blocks to allocate.
RS	Indicates the file is a Files-11 formatted file.
RT	Indicates the specification as an RT-11 format.
SP	Indicates that the converted file is to be spooled to L.P.
UI	Indicates that the output file is to have the same UIC as the input file.
VE	Verify after write (for cassette only).
ZE	Initiates DKn:, DTn:, or MTn: for DOS-11 format or CTn: for DOS or RSX format.

C.2 FLX ERROR MESSAGES

Errors encountered by FLX, during processing, are reported on the user's terminal.

The FLX error messages, their descriptions and suggested user actions are as follows.

FLX -- BAD LIST FILE SPEC

The user has specified one of the following:

1. More than one output file for an /LI or /DI operation.

2. Wild cards in the output file for an /LI or /DI operation.

Reenter the command correctly.

FLX -- CAN'T OPEN FILE

The specified file cannot be found in the requested directory.

Check file name, and retry.

FLX -- CO FILES TO MT: NOT ALLOWED

The user has appended the /CO switch to a magnetic tape output file specifier.

Retry without the /CO switch specified.

FLX -- CASSETTE ERROR I/O TERMINATED

An unexpected hardware error has occurred during the end-of-volume sequence on a cassette volume. The transfer is aborted.

Retry the previously executed command.

FLX -- COMMAND SYNTAX ERROR

The user has entered a command in a format that does not conform with syntax rules.

Reenter the command with the correct syntax.

FLX -- CONFLICTING TRANSFER MODES SPECIFIED

The user has specified conflicting transfer mode swithces.

Example:

```
SY:=DT:FOO.OBJ/IM/FB
```

Retry with switches correctly specified.

FLX -- DOS or RT-11 DEVICE NOT VALID FORMAT

The device specified with the /DO switch has an incorrect DOS file structure; or the device specified with the /RT switch has an incorrect RT file structure.

Correct the device problem, and retry.

FLX -- DT: UFD FULL

The DECtape directory is full - there is no room for the file under the specified ufd.

1. Clean up the directory by deleting unnecessary files.

2. Retry under a different ufd.

FLX -- END OF VOLUME ON CASSETTE

MOUNT NEW CASSETTE? (Y or CR)

Physical end-of-tape has been encountered during a cassette transfer. The tape rewinds, and the user is asked to mount the next cassette.

The user has two choices of action. These choices are as follows:

1. The user can mount the new cassette and type Y to signify that the cassette is mounted, and that FLX can continue transferring data.

2. The user can enter a carriage return. This informs FLX that there is no next cassette. EOF is assumed and the transfer is terminated.

FLX -- ERROR DURING DIRECTORY I/O

One of the following conditions exists:

1. The device is not write enabled.

2. /DO, /RT, or /RS switches were incorrectly specified.

3. The volume is not of the proper format.

4. A hardware error occurred during a directory I/O operation (i.e., bad tape).

1. Turn on the write enable switch.

2. Respecify the /DO, /RT, or /RS switches correctly.

3. Mount a volume which is in the proper format.

4. Retry.

FLX -- FILE ALREADY EXISTS

The user specified an output file that already exists on the (DOS) device specified.

Reenter the file specifier using a new or corrected filename.

FLX -- FILE NESTING EXCEEDED

More than one level of indirect files was specified.

Retry with only one level of indirect file specified.

FLX -- FILE NOT FOUND

The named file does not appear in the requested directory, as specified.

Retry with the file name and directory correctly specified.

FLX -- FILE OUT OF SEQUENCE

A cassette multivolume file is being accessed out of sequence.

This is a warning message. The transfer will continue unless stopped by the user.

FLX -- @FILE SYNTAX ERROR

A syntax error was encountered during indirect command file processing.

1. Edit the indirect command file using either EDI or SLP.

2. Rerun FLX using the corrected command file.

FLX -- FMTD ASCII RECORD FORMAT BAD

or

FLX -- FMTD BINARY RECORD FORMAT BAD

Either the file is corrupted, or the file is not of the specified type.

If the file is corrupted, there is no recovery possible. If the file type is incorrect, retry with the correct type.

FLX -- INCORRECT # IN/OUT SPECS

The user specified more than one output specification.

Retry the command with the proper syntax; i.e., only one output specification.

FLX -- INVALID DEVICE

The user specified a device that cannot be utilized as an input or output device; i.e., trying to read from a line printer.

Retry with a legal device specified.

FLX -- INVALID DOS OR RT-11 FILE SPEC

or

FLX -- INVALID RSX FILE SPEC

The file specifier does not conform to proper syntax; or the specified device is not supported for the desired operation.

Reenter the file specifier with the proper syntax.

FLX -- INVALID SWITCH

The user has specified a switch that is not a FLX switch, or does not conform to proper syntax.

Reenter the command specifying the correct switch.

FLX -- I/O ERROR

One of the following conditions exists.

1. The specified device is not on-line.

2. The specified volume is not mounted.

3. A hardware error has occurred (i.e., bad tape).

1. Ensure that the device is on-line, and the volume is mounted and in good condition.

2. Reenter the command.

FLX -- I/C ERROR DELETING LINKED FILE

An uncorrectable error occurred while a DOS DK or DT file was being deleted.

No action required - the file is effectively deleted.

FLX -- I/O ERROR INITIALIZING DIRECTORY

/DO, /RT, or /RS switch not specified for a DOS, RT, or RSX volume initialization command; or

One of the following conditions exists:

1. The specified device is not on-line.

2. The specified volume is not mounted.

3. A hardware error has occurred (i.e., bad tape).

1. Ensure that the device is on-line, and the volume is mounted and in good condition.

2. Reenter the command with the required switch specified.

FLX -- I/O ERROR ON COMMAND INPUT

An unexpected error in command input was encountered from either an indirect command file or TI:; FLX exits.

Restart FLX.

FLX -- I/O ERROR ON LIST FILE

An error occurred on the output device during a /DI or /LI sequence. A hardware problem exists in the output device (e.g., device powered down).

1. Rectify the condition.

2. Retry the command.

FLX -- OUTPUT DEVICE FULL

The DOS or RT-11 output volume does not contain enough space for the output file.

Delete any unnecessary files, and rerun the job.

FLX -- OUTPUT FILE SPEC NOT ALLOWED

The user specified an output file specifier for a command that does not allow an output file specifier.

Reenter the command without an output file specifier.

FLX -- RECORD TOO LARGE

FLX has detected an input record that is larger than the specified or implied record size for the file - the file is corrupted.

The file in question cannot be used with FLX.

FLX -- SPECIFIED RECORD SIZE BAD, 512. USED

The record size specified by the /FA or /FB switch is not acceptable. A record size of 512 (decimal) bytes is assumed.

This is a warning message - no action is required.

FLX -- UNABLE TO ALLOCATE FILE

The output volume for the specified file has no available space - the volume is full.

Delete any unnecessary files, and rerun the job.

FLX -- UNABLE TO OPEN FILE

The requested file cannot be located in the specified directory.

Reenter the command with the correct file specified.

FLX -- UNABLE TO OPEN LIST FILE

The list file cannot be opened under the specified file name and ufd; specified device may not be a valid RSX volume.

Reenter the command specifying the correct file name, device and ufd.

FLX -- UNDIAGNOSABLE REQUEST

FLX does not recognize the command line syntax.

Reenter the command using the proper syntax.

FLX -- /CO FILES FROM MT: NOT ALLOWED UNLESS BL: SPEC

When transferring files from MT:, the /CO switch can be specified only when the /BL switch is also specified.

Reenter the command specifying the /BL switch.

FLX -- * IN OUTPUT UIC NOT ALLOWED

The ufd field of the output file contains a wild card.

Reenter the command with the output file ufd explicitly specified.

FLX -- * IN VERSION NUMBER NOT ALLOWED

A wild card was detected in the version number field of a file specifier.

Reenter the command with all version numbers explicitly specified.

APPENDIX D

DMP COMMAND SUMMARY AND ERROR MESSAGES

D.1 DMP COMMAND SUMMARY

The DMP utility has one command.

outfile=infile/switch		Dump a file onto the printer.
where switch =	AS	Data should be dumped in ASCII mode.
	BA:n:m	Specifies a base block address.
	BL:n:m	Specifies the first and last logical blocks to be dumped.
	BY	Data should be dumped in byte octal format.
	HD	Includes the file header in the data dumped.
	ID	Causes the current version of DMP to be printed on the listing.
	LB	Causes starting (logical) block number and a contiguous or non-contiguous indication for the file to be printed.
	MD[:n]	Controls line number sequencing during a memory image dump.

D.2 DMP ERROR MESSAGES

DMP -- BAD DEVICE NAME

The user has specified an illegal device in a filespecifier.

Reenter the command specifying the correct device.

DMP -- BLOCK SWITCH REQUIRED IN LOGICAL BLOCK MODE

Self-explanatory -- /BL switch must be specified.

Reenter the command with the /BL switch specified.

DMP -- CANNOT FIND INPUT FILE

The requested file cannot be located in the specified directory.

Reenter the command with the correct filename and ufd specified.

DMP -- COMMAND SYNTAX ERROR

The user has entered a command in a format that does not conform with syntax rules.

Reenter the command with the correct syntax.

DMP -- FAILED TO ASSIGN LUN

The user has specified an illegal device in a file specifier.

Reenter the command with the correct device specified.

DMP -- FAILED TO READ ATTRIBUTES

The user has attempted to access a file for which he does not have read access privileges.

Rerun DMP under the correct UIC.

DMP -- ILLEGAL SWITCH

The user has specified a switch that is not a valid DMP switch.

Reenter the command with the correct switch specified.

DMP -- I/O ERROR ON INPUT FILE

or

DMP -- I/O ERROR ON OUTPUT FILE

An error has occurred during I/O to the file, e.g., the device is off-line.

Correct the situation and retry the command.

DMP -- NO INPUT FILE SPECIFIED

The user has terminated a command without entering an input file specifier.

Reenter the command with an input file specified.

DMP -- NO LISTS OR WILD CARDS ALLOWED

The user either entered a command with more than one input or output filename or entered a wild card in a file specifier.

Reenter the command with only one input file specifier and one output file specifier -- No wild card specifiers allowed.

DMP -- OPEN FAILURE ON INDIRECT FILE

The requested indirect command file does not exist as specified. One of the following conditions exists.

1. The user directory area is protected against an open. (The user's UIC is not recognized by the RSX-11D Executive as being valid for that user directory area.)

2. A problem exists on the physical device (e.g., device cycled down).

3. The volume is not mounted.

4. The specified file directory does not exist.

5. The file does not exist as specified.

1. Determine which of the above conditions exists.

2. Rectify the condition.

3. Reenter the command.

DMP -- OPEN FAILURE ON INPUT FILE

or

DMP -- OPEN FAILURE ON OUTPUT FILE

One of the following conditions exists:

1. The named file doesn't exist as specified.

2. The volume hasn't been mounted.

3. The specified file directory does not exist.

4. A problem exists on the physical device (e.g., device cycled down).

1. Determine which condition exists.

2. Rectify the condition.

3. Rerun DMP.

APPENDIX E

EDI COMMAND SUMMARY AND ERROR MESSAGES

E.1 EDI COMMAND SUMMARY

ADD
A[DD]string
Add the text specified by "string" to the end of the current line.

ADD AND PRINT
AP string
Same as ADD, except the new current line is printed out.

BEGIN
B[EGIN]
Sets the current line pointer to the top of the block buffer or input file.

BLOCK ON or OFF
BL[OCK][ON] or [OFF]
Switch editing modes.

BOTTOM
BO[TTOM]
Sets the current line pointer to the bottom of block buffer or input file.

CHANGE
[n]C[HANGE]/string-1/string-2
Search for string-1 and replace it with the text specified in string-2. n allows the user to repeat the command, thus allowing string-2 to be substituted for string-1 n times.

CLOSE
CL[OSE] filespec
Transfer the remaining lines in the block buffer and the input file into the output file, then close both the input file and the output file.

CLOSES
CLOSES
Close secondary input file, and begin selecting lines from the input file.

CLOSE AND DELETE
CDL filespec

Same as the CLOSE command except that the input file is deleted.

CONCATENATION CHARACTER
CC character

Change command concatenation character to the specified character (default is &).

CTRL/Z
^Z

Same as EXIT if in Edit Mode; otherwise, it causes an immediate exit of EDI.

DELETE
D[ELETE][n] or [-n]

Delete the current and next n-l lines, if n is positive; delete n lines preceding the current line, but not the current line, if n is negative.

DELETE AND PRINT
DP[n] or [-n]

Same as DELETE except that the new current line is printed out.

END
E[ND]

Same as the BOTTOM command.

ERASE
ERASE [n]

Erase the entire block buffer, the current line, and the next n blocks.

EXIT
EX[IT]

Same as CLOSE command except that when files are closed, EDI exits.

EXIT AND DELETE
EDX filespec

Exit from the editing session, close the output file, delete the input file.

EXIT TO BATCH
EB

Input and output files are closed. Output file is delivered to Batch as an input job.

FORM FEED
FF

Insert form feed into block buffer.

FILE
FI[LE] filespec

Transfer lines from the input file to the file specified by filespec.

FIND
   [n]F[IND] string

Find the line starting with "string" or, if n is specified the nth line starting with "string".

INSERT
   I[NSERT][string]

Insert "string" immediately following the current line. If "string" is null, EDI enters Input Mode.

KILL
   KILL

Terminate this editing session; close input and output files; delete the output file.

LINE CHANGE
   [n]LC/string-1/string-2

Same as CHANGE except that all occurrences of string-1 in the current line are changed to string-2.

LIST ON LINE PRINTER
   LP

List the text in the block buffer or input file on the line printer, starting with the current line.

LIST ON TERMINAL
   LI[ST]

Print on user terminal all lines in block buffer or all remaining lines in input file, starting at +n current line.

LOCATE
   [n]L[OCATE] string

Search the block buffer for "string" or, if n is specified, the nth occurrence of "string".

MACRO
   MA[CRO] x definition

Define macro x to be "definition".

MACRO CALL
   MC[ALL]

Define macros from the file MCALL.

MACRO EXECUTE
   nM[x]a

Execute Macro x for n executions passing it the numeric argument a.

MACRO IMMEDIATE
   n<definition>

Immediate Macro - this allows the user to define and execute a macro in one step.

NEXT
N[EXT][n] or [-n]
Establish a new current line + or - n lines from the current line.

NEXT PRINT
NP[n] or [-n]
Next Print; same as Next command, but the new current line is printed out.

OLD PAGE
OL[DPAGE] n
Back up to page n.

OPENS
OPENS filespec
Open secondary input file.

OUTPUT ON or OFF
OU[TPUT][ON] or [OFF]
Turn output on or off.

OVERLAY
O[VERLAY] [n]
Delete the current line and the next n-1 lines, and enter Input Mode.

PAGE
PAG[E][n]
Enter Block Data Mode, if not already in Block Data Mode, and read page n into the block buffer.

PAGE FIND
[n]PF[IND]string
Identical to FIND command except that it searches successive pages until the nth occurrence of "string" is found.

PAGE LOCATE
[n]PL[OCATE] string
Same as LOCATE command, except that successive pages are searched for the value specified by "string".

PASTE
PA[STE]/string-1/string-2
The same as the LINE CHANGE command except that all lines in the remainder of the input file or block buffer are searched for string-1. Wherever found, string-1 is replaced with string-2.

PRINT
P[RINT][n]
Print out the next line, and the next n-1 lines, on the terminal.

READ
REA[D][n]
Read the next n pages into the block buffer.

RENEW
REN[EW][n]

Write the current buffer, and read in the next page.

RETYPE
R[ETYPE][string]

Replace the current line with the text of "string". If "string" is null; the line is deleted.

SAVE
SA[VE][n][filespec]

Save the current line, and the next n-1 lines, in the file specified by filespec.

SEARCH & CHANGE
SC/string-1/string-2

Search for string-1, in the block buffer or input file starting with the line following the current line. When string-1 is found, replace all occurrences in line with string-2.

SELECT PRIMARY
SP

Select primary input file.

SELECT SECONDARY
SS

Select secondary input file.

SIZE
SIZE n

Specify maximum number of lines to be read into the block buffer on a sngle READ.

TAB ON or OFF
TA[B][ON] or [OFF]

Turn automatic tabbing on or off.

TOP
T[OP]

Same as BEGIN command.

TOP OF FILE
TOF

Returns to the top of the input file, in Block Data Mode, and saves all pages previously edited.

TYPE
TY[PE]n

Same as PRINT command except that the current line pointer does not change.

UNSAVE
UNS[AVE][filespec]

Retrieve the lines which were previously saved on filespec and insert them immediately following the current line.

VERIFY ON or OFF V[ERIFY][ON] or [OFF]	Allows user to select whether or not search and change commands are to be verified.
WRITE W[RITE]	Write the current block to the output file, and erase the contents of the buffer.

E.2 EDI ERROR MESSAGES

The four classes of EDI error messages are:

- Command level informational and error messages
- File access warning messages
- Error messages that result in restarting the editing session
- Fatal error messages that result in EDI closing all files and terminating its execution

The following sections describe all the messages that may be displayed in each class. If the recovery procedure is not evident, a suggested user action is supplied.

E.2.1 Command Level Informational and Error Messages

Messages in this class indicate either information that is designed to be helpful to the user or errors that were encountered on the previous command. All messages in this class are enclosed within square brackets and are followed by a prompt for a new command. For example, the following output occurs if a delete command encounters an end-of-buffer in block mode.

```
[EDI -- *EOB*]
*
```

Note that immediately following the message, EDI outputs an asterisk to prompt for the next command.

The messages in this class follow.

[EDI -- ALREADY PASSED THAT PAGE!]

The user has attempted to access a page number that is less than the current page. Prior pages can be accessed only via the OLDPAGE command.

If the PAGE command has been incorrectly entered, retype the command with the proper page number. Else use an OLDPAGE command to access the desired page.

[EDI -- BUFFER CAPACITY EXCEEDED BY]

offending line

[LINE DELETED]

A READ, UNSAVE, INSERT, or OVERLAY command has caused the capacity of the block buffer to be exceeded. The line that caused the overflow is displayed and deleted.

If a new file is being created, empty the buffer with a WRITE command and continue the editing session.

If an existing file is being edited, it may be possible to continue via a RENEW or WRITE command. Otherwise use the CLOSE command to close the output file and save all edits. Reopen the output file as the input file and using the SIZE command, reduce the number of lines read into each buffer; then, using the PAGE LOCATE command, search to the position in the file where editing is to continue.

Occasionally a file that was not created by EDI causes this message (i.e., an attempt to open the file for input produces this message). If this occurs, the following procedure may be used to successfully edit the file.

1. Start the editing session by specifying a filename that does not correspond to any file in the current directory. This causes EDI to create a new file and enter input mode.

2. Type carriage return to enter edit mode.

3. Using the SIZE command, reduce the number of lines read into each buffer.

4. Use the KILL command to terminate the creation of the file.

5. When EDI prompts for a new file specification, enter the name of the desired file.

[EDI -- CONCATENATING CHAR CHANGED TO "&"]

The user has changed the command concatenation character and an OLDPAGE, TOF, or TYPE command has caused it to be changed back to "&".

Use the CC command to reestablish the desired command concatenation character.

[EDI -- CREATING NEW FILE]

The input file specified in the command does not exist and EDI has created a new file. EDI automatically enters Input mode and awaits the input of text lines.

If the intent is to create a new file, continue the editing session entering new lines as required. Otherwise, enter Edit mode by typing carriage return; use the KILL command to terminate the creation of the new file; and when EDI prompts for a new file specification, enter the correct file specification.

[EDI -- ILL CMD]

A command that is not recognized by EDI has been entered; or a command that is not compatible with the current edit mode has been attempted (e.g., a READ command in line-by-line data mode).

[EDI -- ILL NUM]

A non-numeric character has been specified in a numeric field or a negative number has been entered where only positive numbers are allowed.

[EDI -- ILL STRING CONST]

A search string specified in a CHANGE, LC, PASTE, or SC command contains only one command concatenation character, or does not contain a matching string termination character (e.g., PASTE/ALPHABETA whereas PASTE/ALPHA/BETA is correct).

[EDI -- ILLEGAL IN BLOCK ON MODE]

An attempt has been made to execute a command that is illegal in block mode (e.g., an OUTPUT ON or OFF command).

[EDI -- ILLEGAL FILE NAME GIVEN IN CLOSE OR EXIT]

[EDI -- FILE WAS NOT RENAMED]

A syntactically incorrect file specifier has been given in a CLOSE or EXIT command or the attempt to rename the output file has failed.

The output file is closed under the name of the input file without any loss of information. The Peripheral Interchange Program (PIP) can be used to rename the file to the desired name.

[EDI -- MACRO NOT DEFINED]

An attempt has been made to execute a macro with the M command, but the specified macro has not been defined.

Use the MACRO command to define the desired macro and then execute it with the M command.

[EDI -- MACRO NUMERIC ARG UNDEFINED]

A macro has been executed with an M command that did not contain a numeric argument and the body of the referenced macro contains the numeric argument replacement character "%".

Retype the command specifying the appropriate numeric argument.

[EDI -- MCALL FILE DOES NOT EXIST]

An MCALL command has been executed to define a set of macros, but the file MCALL does not exist in the current directory.

The desired set of macro definitions may exist under another UIC. If this is the case, use PIP to copy or rename the MCALL file into the current directory.

[EDI -- NO INPUT FILE OPEN]

A PAGE, READ or RENEW command has been attempted and a new file is being created. These commands can be executed only when an input file is being edited.

[EDI -- NO MATCH]

A CHANGE command has specified a string to be changed that is not in the current line.

[EDI -- OVERLAYING PREVIOUSLY DEFINED MACRO]

A MACRO command has resulted in the redefinition of a previously defined macro. This message is merely to make the user aware that the previous definition is no longer in effect.

[EDI -- SAVE FILE DOES NOT EXIST]

A file was specified in an UNSAVE command that cannot be located in the respective directory.

Examine the file specifier to ensure its correctness. If the filespecifier is in error, correct the error, then retry the command.

[EDI -- SECONDARY FILE ALREADY OPEN]

An attempt has been made to open a secondary input file when another secondary input file is already open. Alternatively, a CLOSE or KILL command has been executed or an error has been encountered that causes EDI to restart and the secondary file is found to be open from the previous edit. The former case represents an error, whereas the latter informs the user that he still has a secondary file open.

Close the secondary input file using the CLOSES command, and then open the desired secondary file with the OPENS command.

[EDI -- SECONDARY FILE CURRENTLY SELECTED FOR INPUT]

A CLOSE or KILL command has been issued or an error has caused EDI to restart, when the secondary input file is open and selected for input.

Issue an SP command, CLOSES command and proceed.

[EDI -- SYNTAX ERROR]

A command has been entered that is syntactically incorrect.

[EDI -- TOO MANY CHARS]

A CHANGE, LC, PASTE, or SC command has resulted in a line that contains too many characters. EDI limits the length of a line to 132 characters.

Retype the line to ensure that the line is valid.

[EDI -- *BOB*]

The beginning-of-buffer has been reached. The current line pointer points to just before the first line in the buffer. Thus new text lines can be entered before the first line.

[EDI -- *EOB*]

The end-of-buffer has been reached. The current line pointer now points to the beginning of the buffer. Thus if new lines are inserted they appear before the first line in the buffer.

[EDI -- *EOF*]

The end-of-file has been reached on the input file.

If the editing session is complete, use the CLOSE or EXIT command to close the output file. Otherwise use the TOF command to return to the first block in the file and then continue editing the file.

E.2.2 File Access Warning Messages

Messages in this class represent attempts on the part of the user to access directories, files, or devices that are not present in the host system. Each message is prefixed with

EDI --

and after the message is displayed, EDI returns to command level and prints an asterisk to request input.

The messages in this class follow.

EDI -- DEVICE NOT IN SYSTEM - TRY ANOTHER COMMAND

A FILE, OPENS, SAVE, or UNSAVE command contains the specification of a device that does not exist in the host system.

Reissue the command, specifying only devices available on the system.

EDI -- FILE DOES NOT EXIST

An attempt has been made in a FILE or SAVE command to create a file under a ufd that has no directory on the specified volume.

WARNING

The remaining error messages in this class should not occur and represent failures in EDI. If such errors persist, consult your DEC software support representative.

EDI -- BAD DEVICE NAME

EDI -- BAD FILE NAME

EDI -- DEVICE NOT READY

EDI -- FILE ALREADY OPEN

EDI -- RENAME NAME ALREADY IN USE

EDI -- RENAME ON TWO DIFFERENT DEVICES

EDI -- WRITE ATTEMPT TO LOCKED UNIT

E.2.3 Error Messages That Result in Restarting the Editing Sessions

The error messages described in this section are caused by conditions encountered by EDI that make it impossible to continue the current editing session. EDI closes all open files (with the exception of the secondary input file), reinitializes, and then prompts for the next file to be edited.

As with file access warning messages, each message in this class is prefixed with:

EDI --

After the appropriate message has been displayed, EDI prompts with:

```
EDI>
```

The editing session may be terminated at this point by typing carriage return or can be continued by entering the next file specifier. If a secondary file was open when error condition was encountered, the secondary file must be closed using the EDI commands. The messages in this class follow.

EDI -- BAD RECORD TYPE-FILE NO LONGER USABLE

> The record type defined in the header block of the input file (primary input, secondary input, UNSAVE, or MCALL) is not supported by File Control Services (FCS) and thus the file cannot be used for input to EDI.
>
> The referenced file has been created without using FCS or the file structure on the volume is damaged. If the latter is the case, the disk structure verification program VFY should be run to determine the extent of the damage.

EDI -- FILE IS ACCESSED FOR WRITE

> The input file (primary input, secondary input, UNSAVE, or MCALL) is currently being written by another task.
>
> Wait for the file to be written and then retry the command.

EDI -- FILE IS LOCKED TO WRITE ACCESS

The output file (text output, FILE, or SAVE) is currently accessed for shared read by one or more tasks and is locked against all writers.

Wait for all readers of the file to finish, then retry the command.

EDI -- ILLEGAL RECORD ATTRIBUTES - FILE NOT USABLE

The record attributes defined in the header block of the input file (primary input, secondary input, UNSAVE, or MCALL) are not supported by FCS and thus the file cannot be used for input to EDI.

The referenced file has been created without using FCS or the file structure on the volume is damaged. If the latter is the case, VFY should be run to determine the extent of the damage.

EDI -- PRIMARY FILE NOT PROPERLY CLOSED - NOT USABLE - EDITOR
WILL RESTART

When the primary input file was last written, a close chech was specified and the writing task did not properly close the file (e.g., the task was aborted.) Thus the file attributes were not written and the file may contain inconsistent data.

Exit from EDI by typing carriage return. Run the peripheral interchange program (PIP) and use the /UN switch to unlock the file. Reinitiate EDI and try to recover the data in the file.

EDI -- PRIVILEGE VIOLATION

A privilege violation occurs during a file access for the following reasons:

1. The specified volume is not mounted.

2. The UIC under which EDI is running does not possess the necessary privileges to access the specified directory.

3. The UIC under which EDI is running does not possess the necessary privileges to access the specified file.

Exit from EDI by typing CTRL Z. Determine the cause of the privilege violation. If the volume is not mounted, then mount it using the MCR MOUNT command. Otherwise reinitiate EDI under a UIC that has appropriate access privileges to both the specified directory and the file.

EDI -- RECORD IS TOO LARGE FOR USER BUFFER

The input file (primary input, secondary input, UNSAVE, or MCALL) being accessed was not created by EDI (or SLP) and contains records that are too large. The maximum record length supported by EDI is 132 bytes.

EDI -- SECONDARY FILE NOT PROPERLY CLOSED - NOT USABLE

When the secondary input file was last written, a close check was specified and the writing task did not properly close the file (e.g., the task was aborted). Thus the file attributes were not written and the file may contain inconsistent data.

Exit from EDI by typing carriage return. Run PIP and use the /UN switch to unlock the file. Reinitiate EDI and try to recover the data in the file.

WARNING

The remaining error messages in this class should not occur and represent failures in EDI. if such errors persist consult your DEC software support representative.

EDI -- BAD DIRECTORY SYNTAX

EDI -- DUPLICATE ENTRY IN DIRECTORY

EDI -- END OF FILE

EDI -- ILLEGAL RECORD ACCESS BITS - FILE NOT USABLE

EDI -- ILLEGAL RECORD NUMBER - FILE NOT USABLE

E.2.6 Fatal Error Messages that Result in EDI Closing All Files and Terminating Its Execution - The fatal error messages represent system and/or hardware error conditions which make it impossible for EDI to continue execution. All files are closed and EDI terminates its execution. The output file may be truncated. Each error message is prefixed with:

EDI --

and followed by the exit message:

[EDI -- EXIT]

on the next line.

The advanced user can utilize the truncated version of an output file in the following manner to save the editing performed prior to the fatal error condition

1. Use PIP to rename the truncated version of the output file to avoid confusion.

2. Restart the editing session on the original input file.

3. Issue an OPENS command specifying the renamed file as the secondary file.

4. Issue an SS command to select the secondary file for input.

5. Issue an ERASE command to erase the first block of the input file unless the truncated output file did not contain the entire first block.

6. Issue as many READ 1 and WRITE commands as necessary to reach the EOF on the secondary file.

7. Issue an SP command to select the primary file for input.

8. Issue a CLOSES command to close the secondary file.

9. Issue a WRITE command to ensure the last block was written into the output file.

10. Issue as many READ 1 and ERASE commands as necessary to bypass all input file blocks which are complete in the renamed file.

11. Continue the normal editing session.

The messages in this class follow.

EDI -- CALLER'S NODE EXHAUSTED

System dynamic storage has been depleted and insufficient space is available to allocate the control blocks necessary to open, close, read, or write a file.

This probably is a system failure, but also could represent a transient overload condition. Wait until system load has diminished and reinitiate EDI.

EDI -- DEVICE FULL

Insufficient space exists on the output volume to extend an output file (text output, FILE, or SAVE).

Determine which volume is being written on. If it is required that the specified file be written on this volume, then space must be made available. Use PIP to purge (/PU) or delete (/DE) unwanted files.

EDI -- FILE HEADER CHECKSUM ERROR

An input file (primary input, secondary input, UNSAVE, or MCALL) has a header block that does not contain a proper checksum.

The file structure on the specified volume is damaged. Run VFY to determine the extent of the damage.

EDI -- FILE HEADER FULL

Insufficient retrieval pointer space exists in the header block to extend an output file (text output, FILE, or SAVE).

An attempt is being made to create an output file that is larger than can be described in a file header block. Split the file into two or more files and process them separately.

EDI -- INDEX FILE FULL

File header block is not available to create an output file (text output, FILE, or SAVE). When a volume is initialized, the maximum number of files that may be created on the volume is established. An attempt has been made to exceed this maximum.

Determine which volume is being referenced. If it is required that the specified file be created on this volume, then space must be made available. Use PIP to purge (/PU) or delete (/DE) unwanted files.

WARNING

The following error messages signify hardware problems. If possible, all important files should be removed from the volume. If errors persist, consult the DEC field service representative.

EDI -- BAD BLOCK ON DEVICE

EDI -- FILE PROCESSOR DEVICE READ ERROR

EDI -- FILE PROCESSOR DEVICE WRITE ERROR

EDI -- HARDWARE ERROR ON DEVICE

EDI -- PARITY ERROR ON DEVICE

WARNING

The remaining error messages in this class should not occur and represent failures in EDI. If such errors persist, consult your DEC software support representative.

EDI -- BAD DIRECTORY FILE

EDI -- BAD PARAMETERS ON A QIO

EDI -- INVALID FUNCTION CODE ON A QIO

EDI -- NO BLOCKS LEFT

EDI -- REQUEST TERMINATED

EDI -- WRITE ATTRIBUTE DATA FORMAT ERROR

EDI -- UNEXPECTED ERROR - EDITOR WILL ABORT

TASK "...EDI" TERMINATED

APPENDIX F

SLP COMMAND SUMMARY AND ERROR MESSAGES

F.1 SLP COMMAND SUMMARY

The SLP utility has only one command.

```
outfile[,listfile/SP or -SP]=infile[/switch]
```

Performs batch-oriented editing to create and maintain source language files on disk or tape cassette.

where switch = AU and -AU — Enable and disable the editing audit trail, which indicates the changes made during the most recent editing session.

BF and -BF — Enable and disable blank fill when an audit trail is being produced.

DB and -DB — Enable and disable double spaced listing.

and where SP and -SP — Enable and disable the spooling of listing files to a file structure volume.

F.2 SLP ERROR MESSAGES

SLP error messages are issued in two different formats:

- SLP followed by two dashes and the error message. If applicable, the command line in error is printed on the next line.
- SLP followed by two dashes, the error message and the offending filename.

EXAMPLES

```
     SLP -- SYNTAX ERROR
     RICKSFILE;MAC,LP:=SHIRLEY.MAC;2

          or

     SLP -- OPEN FAILURE LINE LISTING FILE filename
```

SLP error messages, descriptions, and suggested user actions are as follows.

SLP -- COMMAND SYNTAX ERROR

command line

A syntax error has been detected in a command.

This error is fatal, and causes the editing session to be reinitialized. That is, the user must re-specify the file specifier correctly.

SLP -- ILLEGAL DEVICE NAME

command line

The device specified is not a legal device. This is a fatal error; it causes the editing session to be reinitialized.

Re-specify the file specifier correctly.

SLP -- ILLEGAL DIRECTORY

command line segment

The directory (ufd) specified is not legally specified. This is a fatal error; it causes the editing session to be reinitialized.

Re-specify the file specifier correctly.

SLP -- ILLEGAL ERROR/SEVERITY CODE p1 p2 p3

This error indicates an error in the SLP program.

Retry the command. If the error persists, contact your DEC field support representative.

SLP -- ILLEGAL FILE NAME

command line segment

A file specification is greater than 30 characters in length or contains a wild card (i.e., an asterisk in place of a file specification element). This is a fatal error; it causes the editing session to be reinitialized.

Re-specify the file specifier correctly.

SLP -- ILLEGAL GET COMMAND LINE ERROR

This error indicates an error in the SLP program.

Retry the command. If the error persists, contact your DEC field support representative.

SLP -- ILLEGAL SWITCH

command line segment

The switch is not a legal SLP switch. This is a fatal error; it causes the editing session to be reinitialized.

Re-enter command line with correct switch specified.

SLP -- INDIRECT COMMAND SYNTAX ERROR

command line

The filename specified for the indirect file does not adhere to the proper syntax. This is a fatal error; the currently opened files are closed, and SLP is reinitialized.

Respecify the indirect file specifier correctly.

SLP -- INDIRECT FILE DEPTH EXCEEDED

command line

More than two indirect files have been specified. This is a fatal error; the currently opened files are closed and SLP is reinitialized.

Respecify the indirect file specifier correctly.

SLP -- INDIRECT FILE OPEN FAILURE

command line

The indirect file can't be found on the specified volume. This is a fatal error; the currently opened files are closed and SLP reinitializes.

a. Respecify the file specifier correctly.

b. Ensure that the volume is on line and mounted.

SLP -- I/O ERROR COMMAND INPUT FILE

An attempt to read command input has failed.

a. Ensure that the volume is on line and mounted.

b. Retry the command.

SLP -- I/O ERROR COMMAND OUTPUT FILE

An attempt to write command output (i.e., an error message) has failed.

a. Ensure that the volume is online and mounted.

b. Retry the command.

SLP -- I/O ERROR CORRECTION INPUT FILE filename

An I/O error has been detected during a read of the correction input file.

a. Insure that the volume is online and mounted.

b. Retry the command.

SLP -- I/O ERROR LINE LISTING FILE filename

An error has been detected during the writing of the line listing file. Usually indicates a problem with a device (disk error, etc.)

a. Ensure that the volume is online and mounted.

b. Retry the command.

SLP -- I/O ERROR SOURCE OUTPUT FILE filename

An attempt to write into the source output file has failed.

a. Ensure that the volume is on line and mounted.

b. Retry the command.

SLP -- OPEN FAILURE CORRECTION INPUT FILE filename

The specified correction input file cannot be found; one of the following conditions exists.

1. The user directory is protected against an open.

2. There is a problem with the physical device (e.g., device not online).

3. The volume is not mounted.

4. The specified file directory does not exist.

5. The file does not exist as specified.

This is a fatal error; the currently opened files are closed and SLP reinitializes.

1. Determine which of the above conditions exists.

2. Rectify the condition.

3. Respecify the file specifier correctly.

SLP -- OPEN FAILURE LINE LISTING FILE filename

The file can not be opened for one of the following reasons:

1. The user directory is protected against an open.

2. There is a problem with the physical device (e.g., device not on line).

3. The volume is not mounted.

4. The specified file directory does not exist.

5. The file does not exist as specified.

This is a fatal error; it causes the editing session to be reinitialized.

1. Determine which of the above conditions exists.

2. Rectify the condition.

3. Respecify the filespecifier correctly.

SLP -- OPEN FAILURE SOURCE OUTPUT FILE filename

The specified source output file can not be opened; one of the following conditions exists:

1. The user directory is protected against an open.

2. There is a problem with the physical device (e.g., device not on line).

3. The volume is not mounted.

4. The specified file directory does not exist.

5. There is insufficient space on the disk.

This is a fatal error; it causes the editing session to be reinitialized.

1. Determine which of the above conditions exists.

2. Rectify the condition.

3. Respecify the file specifier correctly.

SLP -- PREMATURE EOF CORRECTION INPUT FILE filename

An out-of-range line number has been specified in a correction file or from the terminal e.g., -1000 has been specified for an 800 line file.

1. Terminate the current editing session.

2. Restart the editing session, entering the correct line number.

SLP -- PREMATURE EOF COMMAND INPUT FILE

This is caused by typing CTRL Z at the terminal, which sends an end-of-file to SLP before the / is read. SLP types out SLP>, indicating that a new file specification is expected.

Restart the editing session at the point where the CTRL/Z was inadvertently typed.

APPENDIX G

LBR COMMAND SUMMARY AND ERROR MESSAGES

G.1 LBR COMMAND SUMMARY

COMPRESS `outfile/CO:size:ept:mnt:=infile`	Physically deletes logically deleted records on a file and puts all free space at the end of the file.

CREATE
`outfile/CR:size:ept:mnt:type`
	Allocates a contiguous library file on a direct access device.

DELETE
`outfile/DE:module-1`
`[:module-2:...:module-n]`
	Deletes library modules and their associated entry points from a file.

DELETE GLOBAL
`outfile/DG:global-1`
`[:global-2:...:global-n]`
	Deletes specified library module entry points from a file.

INSERT
`outfile/IN=infile-1`
`[,infile-2,...,infile-n]`
	Inserts library modules into a library file.

LIST
`outfile[,listfile]/switch(es)`
`/switch(es)=LI`
	Lists all modules in the library file.

LE	Lists all modules in the library file and all their entry points.
FU	Lists all modules in the library file and their entry points, and provides a full module description including size, date of insertion, and version.

REPLACE

```
   outfile/RP=infile-1
     [,infile-2,...,infile-n]

   or

   outfile=infile-1[/RP]
     [,infile-2[/RP],...,infile-n[/RP]]
```

Replaces, and in certain cases, inserts library modules in a library file.

SPOOL

```
   outfile,listfile/SP
```

The listing file is spooled out for printing.

G.2 LBR ERROR MESSAGES

LBR -- ILLEGAL GET COMMAND LINE ERROR CODE

The system, for some reason, is unable to read a command line. This is an internal system failure.

Retry the command.

LBR -- INPUT ERROR ON filename

The file system, while attempting to process an input file, has detected an error.

There is a problem with the physical device (e.g., device cycled down).

Retry the command.

LBR -- COMMAND SYNTAX ERROR

command line

The user has entered a command in a format that does not conform to syntax rules.

Reenter the command with the correct syntax.

LBR -- OUTPUT ERROR ON filename

A write error has occurred on the output file. One of the following conditions exists.

1. The volume is full.

2. The device is write protected.

3. The hardware has failed.

If the volume is full, the user should delete any unneccessary files and rerun LBR.

If the device is write protected the user should write enable the device, and retry the command.

If the hardware has failed, the user can swap devices and retry, or wait until the device is repaired, and rerun LBR.

LBR -- ILLEGAL SWITCH

command line

The user specified a non-LBR switch or a legal switch in an invalid context.

Reenter the command line with the correct switch specifications.

LBR -- INSUFFICIENT DYNAMIC MEMORY TO CONTINUE

The partition in which LBR is running is too small.

Remove the task (LBR), install it in a larger partition, and retry the command.

LBR -- INVALID LIBRARY TYPE SPECIFIED

The user specified an illegal library type in a CREATE command. OBJ is currently the only valid type. See Section 7.5.2.

Reenter the command with OBJ specified or implied by default.

LBR -- COMMAND I/O ERROR

LBR encountered an error while retrieving a command line.

Rerun the job.

LBR -- INDIRECT FILE OPEN FAILURE

command line

The requested indirect command file does not exist as specified. One of the following conditions exists.

1. The user directory area is protected against an OPEN.

2. A problem exists with the physical device (e.g., device cycled down).

3. The volume is not mounted.

4. The specified file directory does not exists.

5. The file does not exist as specified.

1. Determine which of the above conditions exists.

2. Rectify the condition.

3. Reenter the command.

LBR -- INDIRECT COMMAND SYNTAX ERROR

command line

The user specified an indirect file in a format that does not conform to syntax rules.

Reenter the command with the correct syntax.

LBR -- BAD LIBRARY HEADER

Either the file is not a library file, or the file is corrupted.

1. If the file is not a library file, retry the command with a proper library file specified.

2. If the file is a proper library file, the user should run the VFY utility program against the volulme to determine if it is corrupted.

3. If the volume is corrupted, it must be reconstructed, before it can be used.

LBR -- INDIRECT FILE DEPTH EXCEEDED

command line

An attempt has been made to exceed one level of indirect command files.

Rerun the job with only one level of indirect command file.

LBR -- I/O ERROR IN INPUT FILE

A read error has occurred on an input file. One of the following conditions may exist.

1. A problem exists with the physical device (e.g., not cycled up).

2. The file is corrupted or the format is wrong. (Record length exceeds 132 bytes.)

3. The file is a non-object file. (Source file, etc.)

1. Determine which of the above conditions exists.

2. Rectify the condition.

3. Reenter the command.

LBR -- OPEN FAILURE ON FILE filename

The file system, while attempting to open a file, has detected an error. One of the following conditions may exist.

1. The user directory area is protected against an open.

2. A problem exists with the physical device (e.g., device cycled down).

3. The volume is not mounted.

4. The specified file directory does not exist.

5. The file does not exist as specified.

6. Insufficient contiguous space to allocate the file.

1. Determine which of the above conditions exists.
2. Rectify the condition.
3. Reenter the command.

LBR -- INVALID EPT AND/OR MNT SPECIFICATION

The user, when specifying a /CR or /CO command, entered an EPT or MNT value which was greater than 4096.

Retry the command with the correct value specified.

LBR -- POSITIONING ERROR ON filename

The device is write-locked.

If the device is write-locked, write-enable it and retry.

LBR -- EPT OR MNT EXCEEDED IN filename

The EPT or MNT table limit has been reached during the execution of an Insert or Replace command.

1. Copy the library, increasing the table space, using the COMPRESS command.
2. Retry the command.

LBR -- DUPLICATE MODULE NAME "name" IN filename

An attempt has been made to insert (without replacement) a module into a library that already contains a module with the specified name.

1. Determine if the specified input file is the correct file.
2. If the input file is correct, the user must decide whether to delete the duplicate module from the library file and insert the new one. or replace the duplicate module by rerunning LBR with the /RP switch appended to the input file specifier.

LBR -- GET TIME FAILED

This error occurs when LBR attempts to execute a GET Time Parameters directive and fails. The error is caused by some system malfunction.

Retry the command.

LBR -- NO MODULE NAMED "module"

The user has attempted to delete a module that is not in the specified library file.

1. Determine if the module name is misspelled, or if the wrong library file is specified.
2. Retry the command with module name correctly specified.

LBR -- INVALID NAME -- "name"

A module name or entry point that contains a non-Radix-50 character has been specified for deletion.

NOTE

Radix-50 characters consist of the letters of the alphabet A through Z, the numbers 0 through 9, and the special characters period (.) and dollar sign ($).

Retry the command with a valid name.

LBR -- LIBRARY FILE SPECIFICATION MISSING

The user has entered a command without specifying the library file.

Retry the command with the library file specified.

LBR -- ILLEGAL SWITCH COMBINATION

The user has specified switches that cannot be executed in combination. See Section 7.6.

Retry with command switches in the proper sequence.

LBR -- NO ENTRY POINT NAMED name

The user has attempted to delete an entry point that is not in the specified library file.

1. Determine if the entry point is misspelled, or if the wrong library file is specified.

2. Retry the command with the entry point correctly specified.

LBR -- DUPLICATE ENTRY POINT NAMED "name" IN filename

An attempt has been made to insert a module into a library file when both contain an identically named entry point.

1. Determine if the specified input file is the correct file. If not, retry with the correct input file specified.

2. If the input file is the correct file, the user may delete the duplicate entry point from the library and rerun.

LBR -- TOO MANY OUTPUT FILES SPECIFIED

The user has specified more than two output files. LBR makes the following assumptions:

1. The first output file specified is the output library file.

2. The second output file specified is the listing file.

3. The third through n files specified to the left of the equal sign are ignored.

No action is required. LBR continues as though the extra file(s) were not specified.

LBR -- I/O ERROR ON OUTPUT FILE

A write error has occurred on the output file. One of the following conditions exists.

1. The volume is full.

2. The hardware has failed.

3. The device is write protected.

4. The user does not have write access privileges to the file.

If the volume is full, the user should delete any unnecessary files, and retry the command.

If the hardware has failed, the user can swap devices and retry the command, or wait until the device is repaired, and retry the command.

If the device is write protected, the user should write enable it, and retry the command.

If the user does not have write access, consult the System Administrator who can take necessary action.

LBR -- EXACTLY ONE INPUT FILE MUST APPEAR WITH /CO

The user has specified no file or more than one input library file in the /CO command.

Retry the command with only one input file specified.

LBR -- FATAL COMPRESS ERROR

The user's input library file is corrupted or is not a library file.

No recovery is possible. The file in question must be reconstructed.

LBR -- EPT OR MNT SPACE EXCEEDED IN COMPRESS

The user has specified an EPT or MNT table size for the output library file that is not large enough to contain the EPT or MNT entries used in the input library file.

Reenter the command with a larger EPT or MNT table size specified.

LBR -- ERROR IN LIBRARY TABLES, FILE filename

The library file is corrupted, or is a non-library file.

If the file is corrupted no restart is possible; the file must be reconstructed.

If the file is a non-library file, the user should retry the command with the correct library file specified.

LBR -- INVALID FORMAT, INPUT FILE filename

The specified input file is not a standard format object file or the input file is corrupted.

Reenter the command with the correct input file specified.

LBR -- OPEN FAILURE ON LBR WORK FILE

The file system, while attempting to open the LBR work file, has detected an error.

NOTE

The LBR work file is created on the volume from which LBR was installed.

One of the following conditions exists:

1. The volume is full.
2. The device is write protected.
3. A problem exists with the physical device.

1. Determine which of the above conditions exists.
2. Rectify the condition.
3. Retry the command.

LBR -- MARK FOR DELETE FAILURE ON LBR WORK FILE

When LBR begins processing commands, it automatically creates a work file, and automatically marks it for delete. For some reason, this marking for delete failed.

The work file constitutes a lost file, because it does not appear in any file directory.

The file may be deleted by running the utility (See Chapter 9).

LBR -- ILLEGAL FILENAME

command line

The user has entered one of the following:

1. A file specifier which contains a wild card.
2. A file specifier which does not contain a filename or type.

Reenter the command correctly.

LBR -- ILLEGAL DEVICE/VOLUME

command line

The device specifier does not conform to the proper syntax.

NOTE

> A device specifier consists of 2 ASCII characters followed by one or two octal digits.

Retry the command with the correct device syntax specified.

LBR -- ILLEGAL DIRECTORY

command line

The ufd field in a file specifier does not conform to the proper syntax.

NOTE

> ufd syntax consists of a left square bracket, followed by one to three octal digits, followed by a comma, followed by one to three octal digits, followed by a right square bracket.

Retry the command with the correct ufd syntax.

LBR -- WORK FILE I/O ERROR

A write error has occurred on the LBR work file. One of the following conditions exists.

1. The volume is full.
2. The device is write protected.
3. The hardware has failed.

If the volume is full, the user should delete any unnecessary files and rerun.

If the device is write protected, the user should write enable the device, and retry the command.

If the hardware has failed, the user can swap devices and retry, or wait until the device is repaired and rerun LBR.

LBR -- VIRTUAL STORAGE REQUIREMENTS EXCEED 65536 WORDS

This error may occur with maximum size libraries and a single command line which logically deletes a large number of modules and entry points, and continues to replace them with an equally large number of modules and entry points having highly dissimilar names.

Normally, this message indicates some sort of internal system error.

Rerun the job, but divide the complicated command line into several smaller command lines which do the same operations.

APPENDIX H

SQZ COMMAND SUMMARY AND ERROR MESSAGES

H.1 SQZ COMMAND SUMMARY

The Squeeze utility has only one command.

SQZ outfile=infile-1 [,infile-2,...,infile-n]	Reduces the size of programs.

H.2 SQZ ERROR MESSAGES

SQZ -- INDIRECT FILE OPEN FAILURE

The requested indirect command file does not exist as specified. One of the following conditions exists.

1. The user directory file is protected against an open.
2. A problem exists with the physical device (e.g., device cycled down).
3. The volume is not mounted.
4. The specified file directory does not exist.
5. The file does not exist as specified.

1. Determine which of the above conditions exists.
2. Rectify the condition.
3. Reenter the command.

SQZ -- INDIRECT COMMAND SYNTAX ERROR

The user has specified an indirect file in a format that does not conform to syntax rules.

Reenter the command with the correct syntax.

SQZ -- INDIRECT FILE DEPTH EXCEEDED

The maximum depth of indirect command files is one level. An attempt has been made to reference an indirect command file when the current depth of indirect command is already one.

Rerun the job with one level of indirect files.

SQZ -- COMMAND I/O ERROR

SQZ has encountered an error while retrieving a command line.

Reenter the command.

SQZ -- COMMAND SYNTAX ERROR

The user has entered a command in a format that does not conform with syntax rules.

Reenter the command with the correct syntax.

SQZ -- ILLEGAL SWITCH

The user has specified a switch where none is allowed. SQZ has no switches, therefore, any switch specification is illegal.

Reenter the command without switch specifications.

SQZ -- ILLEGAL FILE SPECIFICATION

The user has entered one of the following:

1. A file specifier which contains a wild card.
2. A file specifier which does not contain a filename or type.

1. Examine the command in error.
2. Reenter the command correctly.

SQZ -- OUTPUT FILE OPEN ERROR

The file system, while attempting to open the output file (SQZ.TMP), has detected an error. One of the following conditions may exist.

1. The user directory file is protected against an open. (The user's UIC is not recognized by the RSX-11D Executive as being valid for that user directory area.)
2. A problem exists with the physical device (e.g., device cycled down).
3. The volume is not mounted.
4. The specified file directory does not exist.
5. The device is full or write protected.

1. Determine which of the above conditions exists.
2. Rectify the condition.
3. Reenter the command.

SQZ -- OUTPUT FILE I/O ERROR

A write error has occurred on the output file. One of the following conditions exists.

1. The device is full or is write protected.

2. The hardware has failed.

NOTE

The file SQZ.TMP remains intact.

If the volume is full, delete any unnecessary files and rerun SQZ.

If a hardware error has occurred, swap devices and rerun SQZ, or wait until the device is repaired, and rerun SQZ.

SQZ -- OUTPUT FILE RENAME ERROR

SQZ has failed in its attempt to rename the SQZ.TMP file to the name specified in the output file specifier. SQZ restarts and the SQZ.TMP file remains intact.

Rename the SQZ.TMP file to the desired name by using the PIP utility program.

SQZ -- INPUT FILE OPEN FAILURE

The file system, while attempting to open an input file, has detected an error. One of the following conditions may exist.

1. The user directory file is protected against an open. (The user's UIC is not recognized by the RSX-11D Executive as being valid for that user directory area).

2. A problem exists with the physical device (e.g., device cycled down).

3. The volume is not mounted.

4. The specified file directory does not exist.

5. The file does not exist as specified.

1. Determine which of the above conditions exists.

2. Rectify the condition.

3. Reenter the command.

SQZ -- INPUT FILE I/O ERROR

A read error has occurred on an input file. One of the following conditions may exist.

1. A problem exists with the physical device (e.g., device cycled down).

2. The file is corrupted or the format is wrong (line length exceeds 132 bytes).

3. The file is a nonsource file. (Object file, etc.)

1. Determine which of the above conditions exists.

2. Rectify the condition.

3. Reenter the command.

APPENDIX I

VFY COMMAND SUMMARY AND ERROR MESSAGES

I.1 VFY COMMAND SUMMARY

DELETE

```
   listfile,scratchdev=indev/DE
     or indev/DE
```

Resets the marked-for-delete indicators in the file header area of those files marked for deletion, but were never actually deleted.

FREE

```
   listfile=indev/FR
     or indev/FR
```

Prints out the available space on a volume.

LIST

```
   listfile, scratchdev=indev/LI
     or indev/LI
```

Lists the entire index file by file identification.

LOST

```
   listfile, scratchdev=indev/LO
     or indev/LO
```

Scans the entire file structure looking for files that are not in any directory.

REBUILD

```
   listfile, scratchdev=indev/RE
     or indev/RE
```

Recovers blocks that appear to be allocated, but are not contained in any file.

UPDATE

```
   listfile, scratchdev=indev/UP
     or indev/UP
```

Allocates blocks that appear to be available, but are actually allocated to a file.

READ CHECK

```
   listfile=dev:/RC[:n]
     or dev:/RC[:n]
```

Checks that every block of every file on specified volume can be read.

I.2 VFY ERROR MESSAGES

VFY -- COMMAND SYNTAX ERROR

The command does not conform to command syntax rules.

Re-enter the command with the correct syntax specified.

VFY -- FAILED TO ALLOCATE SPACE FOR TEMP FILE

The volume used for the temp file is full.

Use PIP to delete any unnecessary files, and rerun VFY.

VFY -- FAILED TO ATTACH DEVICE

or

VFY -- FAILED TO DETACH DEVICE

or

VFY -- ILLEGAL DEVICE

The file specifier entered contains an illegal device.

Reenter the command with the correct device specified.

VFY -- ILLEGAL SWITCH

The switch specified is not a valid VFY switch.

Reenter the command with the correct switch specified.

VFY -- I/O ERROR ON INPUT FILE

or

VFY -- I/O ERROR ON OUTPUT FILE

One of the following conditions exists:

1. The device is not online.
2. The device is not mounted.
3. The hardware has failed.

1. Determine which of the above conditions exists.
2. Rectify the condition.
3. Rerun VFY.

VFY -- NO DYNAMIC MEMORY AVAILABLE - PARTITION TOO SMALL

VFY does not have enough buffer space to run.

The system manager must rebuild VFY to increase buffer space.

VFY -- OPEN FAILURE ON BIT MAP

or

VFY -- OPEN FAILURE ON INDEX FILE

or

VFY -- OPEN FAILURE ON LISTING FILE

or

VFY -- OPEN FAILURE ON TEMPORARY FILE

One of the following conditions exists:

1. VFY is not running under a system UIC, but it should be.
2. The named file does not exist as specified.
3. The device has not been mounted.

1. Determine which of the above conditions exists:
2. Rectify the condition.
3. Rerun VFY.

I.2.1 VFY Error Codes

If VFY cannot access the message files because the system device (SY:) is not mounted or MO is not loaded, errors are reported in the following format

```
VFY -- ERROR CODE nn.
```

where

nn. is one of the error codes contained in Table I-1.

Refer to Section I.2 for error description and suggested user action.

Table I-1
VFY Error Codes

ERROR CODE	VFY ERROR MESSAGE IS:
1.	ILLEGAL DEVICE
2.	OPEN FAILURE ON BIT MAP
3.	OPEN FAILURE ON TEMPORARY FILE
4.	FAILED TO ALLOCATE SPACE FOR TEMP FILE
5.	FAILED TO DETACH DEVICE
6.	FAILED TO ATTACH DEVICE
7.	COMMAND SYNTAX ERROR
8.	I/O ERROR ON INPUT FILE
9.	I/O ERROR ON OUTPUT FILE
10.	ILLEGAL SWITCH
11.	OPEN FAILURE ON LISTING FILE
12.	OPEN FAILURE ON INDEX FILE
13.	NO DYNAMIC MEMORY AVAILABLE - PARTITION TOO SMALL

APPENDIX J

ZAP COMMAND SUMMARY AND ERROR MESSAGES

J.1 ZAP COMMAND SUMMARY

```
ZAP>dev:infile[/switch]

        or

MCR>ZAP@dev:infile[/switch]
```

where switch = AB	ZAP commands are treated as absolute byte addresses within the file.
LI	Line segment data is listed on user terminal.
RO	File is opened for read-only.

J.2 ZAP ERROR MESSAGES

ZAP -- NO SUCH RELOCATION REGISTER

Relocation registers are numbered 0 through 7.

Retry command using the correct relocation register number.

ZAP -- ADDRESS NOT WITHIN SEGMENT

The specified address was not within the overlay segment specified.

Retry command using the correct address or overlay segment number.

ZAP -- NO SUCH SEGMENT

The starting disk block was not the start of any segment in the task image.

Retry command using the correct starting disk block address.

ZAP -- TOO MANY ARGUMENTS

More arguments were given in a command than were expected.

Retry the command using the correct command syntax.

ZAP -- NO SUCH INTERNAL REGISTER

The character following a dollar sign was not the name of an internal register.

Retry command using the correct internal register name.

ZAP -- UNRECOGNIZED COMMAND

Possibly, the command contained an illegal character.

Retry command using correct command syntax.

ZAP -- NO OPEN LOCATION

A modification command was given with no location open to receive the data.

Open the desired location and perform modification.

ZAP -- CANNOT BE USED IN BYTE MODE

The commands, @, ^ and > cannot be used when a location is open as a byte.

Open location in word mode if it is an even address.

ZAP -- ERROR IN FILE SPECIFICATION

The file specification was not properly formed.

Reenter the file specifier.

ZAP -- OPEN FAILURE FOR TASK IMAGE FILE

The file to be modified could not be opened. Possibly it does not exist, or is locked, or the device was not mounted, or the user does not have write access to the file.

Check the file specifier for errors; check that the volume is properly mounted, check file permissions using PIP.

ZAP -- ERROR ON COMMAND INPUT

An I/O error occurred while a command was being read.

Possible hardware error; check hardware and retry command if hardware is operational.

ZAP -- I/O ERROR ON TASK IMAGE FILE

An I/O error occurred while reading or writing the file to be modified.

Possible hardware error; check hardware and retry command if hardware is operational.

ZAP -- NOT A TASK IMAGE FILE

An error occurred while the segment description tables were being constructed. Possibly, the file is not a task image, /AB was not specified, or the task image is defective.

Reinitiate ZAP and specify the /AB switch in the file specifier.

ZAP -- VERIFY FAILURE

The V command determined that the actual contents of a location did not match the expected contents.

Error could be in an indirect file or unexpected content in the task file. Locate cause of error, correct error, and retry command.

APPENDIX K

MACRO-11 ASSEMBLER COMMAND SUMMARY AND ERROR MESSAGES

K.1 COMMAND SUMMARY

The MAC command string format is as follows:

```
objectfile/switch,listfile/switch=source1,source2,...,
sourcen/switch
```

where:

objectfile - object file specifier for the object file output of compliation in the format:

```
dev:[UIC]filename.typ;ver
```

listfile - listing file in the format:

```
dev:[UIC]filename.typ;ver
```

source - input source file specifier in the format

```
dev:[UIC]filename.typ;ver
```

/switch - assembler control switch. Section 7.3 contains a summary of the assembler control switches and their descriptions.

K.1.1 Summary Of Macro Assembler Control Switches

K.1.2 Listing Control Switches

The following switches may only appear in the Listfile specifier.

Switch	Function
/LI:arg	Printed listing requested
/NL:arg	No printed listing requested

NOTE

The optional values for ":arg" are described in Table K-1.

Table K-1
Argument Values for Listing Control Switches

Argument	Default	Function
SEQ	list	Controls the listing of source line sequence numbers. Error flags are normally printed on the line preceding the questionable source statement.
LOC	list	Controls the listing of the location counter (this field would not normally be suppressed).
BIN	list	Controls the listing of the generated binary code.
BEX	list	Controls listing of binary extensions; that is, those locations and binary contents beyond the source statement. This is a subset of the BIN argument.
SRC	list	Controls the listing of the source code.
COM	list	Controls the listing of comments. This is a subset of the SRC argument and can be used to reduce listing time and/or space where comments are unnecessary.
MD	list	Controls listing of macro definitions and repeat range expansions.
MC	list	Controls the listing of Macro calls and repeat range expansions.
ME	no list	Controls listing of macro expansions.
MEB	no list	Controls listing of macro expansion binary code. A LIST MEB causes only those macro expansion statements producing binary code to be listed. This is a subset of the ME argument.
CND	list	Controls the listing of unsatisfied conditions and all .IF and .ENDC statements. This argument permits conditional assemblies to be listed without including unsatisfied code.
LD	no list	Controls listing of all listing directives having no arguments (those used to alter the listing level count).
TOC	list	Controls listing of table of contents on pass 1 of the assembly. The full assembly listing is printed during pass 1 of the assembly.

Table K-1 (cont)
Argument Values for Listing Control Switches

Argument	Default	Function
TTM	Console mode	Controls listing output format. The TTM argument (the default case) causes output lines to be truncated to 72 characters. Binary code is printed with the binary extensions below the first binary word. The alternative (.NLIST TTM) to Teletype mode is line printer mode.
SYM	list	Controls the listing of the symbol table for the assembly.

K.1.3 Function Control Switches

The following switches may only be appended to the objectfile specifier.

Switch	Purpose
/EN:arg	Enable the function specified as "arg".
/DS:arg	Disable the function specified as "arg".

NOTE

The optional values for "arg" are described in Table K-2.

Table K-2
Function Control Arguments

Argument	Default	Function
ABS	disable	Enabling of this function produces absolute binary output; i.e., input to the Paper Tape Software System Absolute Loader.
AMA	disable	Enabling of this function directs the assembly of all relative addresses (address mode 67) as absolute addresses (address mode 37). This switch is useful during the debugging phase of program development.
CDR	disable	The statement .ENABL CDR causes source columns 73 and greater to be treated as comment. This accommodates sequence numbers in card columns 73-80.
FPT	disable	Enabling of this function causes floating-point truncation, rather than rounding, as is otherwise performed. .DSABL FPT returns to floating point rounding mode.
LC	disable	Enabling of this function causes the Assembler to accept lower case ASCII input instead of converting it to upper case.
LSB	disable	Enable or disable a local symbol block. While a local symbol block is normally entered by encountering a new symbolic label or .PSECT directive, .ENABL LSB forces a local symbol block which is not terminated until a label or a .PSECT directive following the .DSABL LSB statement is encountered.
PNC	enable	The statement .DSABL PNC inhibits binary output until an .ENABL PNC is encountered.

REG	enable	The statement .DSABL REG inhibits the default register definitions. That is, until .DSABL REG is seen, the following code is implied as being present:
		R0=%0
		R1=%1
		R2=%2
		R3=%3
		R4=%4
		R5=%5
		SP=%6
		PC=%7
		The .ENABL REG statement may be used to re-enable these definitions. Such use is not recommended.

GBL	enable	The statement .DSABL GBL inhibits attempts to resolve references which remain undefined at the end of pass 1, as being global references.

K.1.4 Assembler Control Switches

Switch	Function
/PA:1	Assemble file during Pass 1 only
/PA:2	Assemble file during pass 2 only

NOTE

The above switches apply only to the sourcefile specifier.

Switch	Function
/SP	Spool output (default)
/-SP	Do not spool output

NOTE

The above switches apply only to the listingfile specifier.

K.2 MAC ERROR MESSAGES

MAC -- COMMAND I/O ERROR

An error has been returned by the file system during MACRO-11's attempt to read a command line. This is an unconditionally fatal error, causing MACRO-11 to exit to the Executive. No MACRO-11 restart is attempted when this message appears.

MAC -- COMMAND SYNTAX ERROR

An error has been detected in the syntax of the MACRO-11 command line.

MAC -- ILLEGAL FILENAME

Neither the device name nor the filename is present in the input file specifier (i.e., the input file specifier is null), or a wildcard convention (asterisk) is employed in an input or output file specification. Wildcard specifiers (*) are not permitted in MACRO-11 file specifiers.

MAC -- ILLEGAL SWITCH

An illegal switch is specified for a file, an illegal value is specified with a switch, or an invalid use of a switch has been detected by MACRO-11.

MAC -- INDIRECT COMMAND SYNTAX ERROR

The name of the indirect command file (@filename) specified in the MACRO-11 command line is syntactically incorrect.

MAC -- INDIRECT FILE DEPTH EXCEEDED

More than one indirect command file (@filename) is specified in the MACRO-11 command line. (Only one level of indirect command files is permitted in MACRO-11.)

MAC -- INDIRECT FILE OPEN FAILURE

The indirect command file specified as "@filename" in the MACRO-11 command line does not exist.

MAC -- INVALID FORMAT IN MACRO LIBRARY

The library file has been corrupted or it was not produced by the Librarian Utility Program (LBR).

MAC -- I/O ERROR ON INPUT FILE

In reading a record from a source input file or macro library file, an error was detected by the file system, e.g., a line containing more than 132(10) characters is encountered. This message may also indicate that a device problem exists or that either a source file or a macro library file has been corrupted with incorrect data.

MAC -- I/O ERROR ON MACRO LIBRARY FILE

Same meaning as I/O ERROR ON INPUT FILE, except that the file is a macro library file and not a source input file.

MAC -- I/O ERROR ON OUTPUT FILE

In writing a record to the object output file or the listing output file, an error was detected by the file system. This message may also indicate that a device problem exists or that the storage space on a device has been exhausted (i.e., the device is full).

MAC -- OPEN FAILURE ON INPUT FILE

The file specified as "filename" in the input file specifier does not exist.

MAC -- OPEN FAILURE ON OUTPUT FILE

MACRO-11 is unable to create the specified output file.

MAC -- SYMBOL TABLE OVERFLOW

Available symbol table space in dynamic memory has been exceeded.

APPENDIX L

TKB COMMAND SUMMARY AND ERROR MESSAGES

L.1 TKB COMMAND SUMMARY

The TKB command string format is as follows:

```
image[/switch(es)],map,[/switch(es)]symbol[/switch(es)]=input/switch(es)
```

where:

image — is the relocatable executable task file (.TSK file) in the format:

```
dev:[UIC]filename.typ;ver
```

map — is the listing device in the format:

```
dev:
```

symbol — is the STB file which contains all the global symbols defined in this task. This file has the format:

```
dev:[UIC]filename.typ;ver
```

input — is the .OBJ file to be task built in the format:

```
dev:[UIC]filename.type;ver
```

/switch(es) — is one or more of the TKB control switches.

L.1.1 TKB File Switches

Table L-1
Task Image File Switches

Switch	Function
/AB	The task is abortable. /AB is assumed.
/CP	The task is checkpointable. /-CP (not checkpointable) is assumed.
/DA	A debugging aid is to be loaded with the task. This switch causes ODT.OBJ, which resides in the system area (SY:[1,1],) to be automatically included in the task image. /-DA is assumed.
/DS	The task can be disabled. /DS is assumed.
/FP	The task uses floating point. /FP is assumed.
/FX	The task can be fixed in memory. /FX is assumed.
/HD	The task image includes a header. /HD is assumed.
/MU	The task is multiuser. /-MU is assumed.
/PI	The task contains only position-independent code. /-PI is assumed.
/PR	The task is privileged. /-PR is assumed.
/SQ	The segments in the task image file are constructed of P-sections in the order in which they were encountered during input file processing. The default, /-SQ, causes the segments to be constructed alphabetically by P-section name.
/TA	The task is accountable. /-TA is assumed.
/TR	The processor T bit is to be set in the initial PS word of the task. /-TR is assumed.
/XT:n	Exit after any n diagnostics are produced. /-XT is assumed.
If no switches are specified, the resultant task image file has the following default attributes: /AB,/CP,/-DA,/DS,/-FP,/FX,/HD,/-MU/-PI,/-PR,/-SQ,/-TA,/-TR,/-XT	
Memory Allocation Map File Switches	
/CR	Produce a cross reference listing. /-CR is assumed.
/SH	Produce a short map. /-SH is assumed.
/SP	Spool the memory allocation file. /SP is assumed.

Table L-1 (cont)
Task Image File Switches

Switch	Function
Symbol Definition File Switches	
/PI	The task image contains only position-independent code. Therefore, the symbol definition file is written as a relocatable file (i.e., the symbol definitions may be relocated). /-PI is assumed.
Input Object File Switches	
/CC	The specified file contains concatenated object files. Note that when /CC is used, it results in an additional pass of the object file and, therefore, should be used judiciously.
/DA	The specified file contains a debugging aid (e.g., ODT) transfer address. This switch must be included to specify a debugging aid with a transfer address as an input file. It is included so that experimental or private copies of debugging aids other than ODT may be included in the task image.
/LB	The specified file contains a user relocatable library (as built by the RSX-11D Librarian; refer to Part II of this manual). This library will be searched to resolve undefined global symbols.
/MP	The specified file contains an overlay description of the Task to be built. If this switch is present, it must appear on the first input file specification. See below.
/SS	Perform selective symbol search. /-SS is assumed.

```
TKB>/MP

ENTER OPTIONS:

TKB>

     .

     .     OPTIONS

     .

TKB>/
```

OPTIONS:

ABBRT = ANY DIGIT -- ABORT THE BUILD

ABSPAT = SGNAM:PADDR:VALUE:VALUE:...:VALUE

ACTFIL = NUM

ASG = devnam:unit:unit:...:unit

ASG default assignments:

Symbolic	LUN
SY0	1
SY0	2
SY0	3
SY0	4
TI	5
CL	6

BASE = BOUND

COMMON = CNAME:ACCESS:APR

EXTSCT = CNAME:LENGTH

EXTTSK = TASK EXTENSION

FMTBUF = BUFFER EXTENSION

GBLDEF = SNAME:VALUE

GBLPAT = SGNAM:SNAME+OFFSET:VALUE:VALUE:...:VALUE

LIBR = LNAME:ACCESS:APR

MAXBUF = MAXIMUM RECORD BUFFER SIZE

ODTV = SNAME:LENGTH

PAR = PNAME

POOL = PNUM

PRI = PNUM

STACK = SNUM

TASK = TASKNAME

TSKV = ADDRESS OF TASK SST VECTOR

UIC = [PROJ,PROG]

```
UNITS = NUNIT
```

L.2 TKB ERROR MESSAGES

The Task Builder produces diagnostic and fatal error messages. Error messages are printed in the following forms:

```
TKB -- *DIAG*-error-message
```

or

```
TKB -- *FATAL*-error-message
```

Some errors are dependent upon correction from the terminal. If the user is entering text at the terminal, a diagnostic error message can be printed, the error corrected, and the task building sequence continued. If the same error is detected by the Task Builder in an indirect file, the Task Builder cannot request correction and thus the error is termed fatal and the task build is aborted.

Some diagnostic error messages are simply informative and advise the user of an unusual condition. If the user considers the condition normal to his task, he can install and run the task image.

This appendix tabulates the error messages produced by the Task Builder. Most of the error messages are self-explanatory. The Task Builder prints the text shown in this manual in upper case letters. In some cases, the Task Builder prints the line in which the error occurred, so that the user can examine the line which caused the problem and correct it.

0. ILLEGAL GET COMMAND LINE ERROR

   System error. (No recovery.)

1. COMMAND SYNTAX ERROR
   invalid-line

   The invalid-line printed has incorrect syntax.

2. REQUIRED INPUT FILE MISSING

   At least one file is required for a task build.

3. ILLEGAL SWITCH
   invalid-line

   The invalid line printed contains an illegal switch or switch value.

4. NO DYNAMIC STORAGE AVAILABLE

   The Task Builder needs additional symbol table storage and cannot obtain it. The input has exceeded the Task Builder's capability.

5. ILLEGAL ERROR/SEVERITY CODE

   System error. (No recovery.)

6. COMMAND I/O ERROR

   I/O error on command input device. (Device may not be online or possible hardware error.)

7. INDIRECT FILE OPEN FAILURE
   invalid-line

   The invalid-line contains a reference to a command input file which could not be located.

8. INDIRECT COMMAND SYNTAX ERROR
   invalid-line

   The invalid-line printed contains a syntactically incorrect indirect file specification.

9. MAXIMUM INDIRECT FILE DEPTH EXCEEDED
   invalid-line

   The invalid-line printed gives the file reference that exceeded the permissible indirect file depth (2).

10. I/O ERROR ON INPUT FILE file-name

11. OPEN FAILURE ON FILE file-name

12. SEARCH STACK OVERFLOW ON SEGMENT segment-name

    The segment segment-name is more than 16 branch segments from the root segment.

13. PASS CONTROL OVERFLOW AT SEGMENT segment-name

    The segment segment-name is more than 16 branch segments from the root segment.

14. FILE file-name HAS ILLEGAL FORMAT

    The file file-name contains an object module whose format is not valid.

15. MODULE module-name AMBIGUOUSLY DEFINES P-SECTION p-sect-name

    The p-section p-sect-name has been defined in two modules not on a common path and referenced ambiguously.

16. MODULE module-name MULTIPLY DEFINES P-SECTION p-sect-name

    1. The p-section p-sect-name has been defined in the same segment with different attributes.

    2. A global p-section has been defined in more than one segment along a common path with different attributes.

17. MODULE module-name MULTIPLY DEFINES XFR ADDR IN SEG segment-name

    This error occurs when more than one module comprising the root has a start address.

18. MODULE module-name ILLEGALLY DEFINES XFR ADDRESS p-sect-name addr

    The module module-name is in an overlay segment and has a start address. The start address must be in the root segment of the main tree.

19. P-SECTION p-sect-name HAS OVERFLOWED

    A section greater than 32K has been created.

20. MODULE module-name AMBIGUOUSLY DEFINES SYMBOL sym-name

    Module module-name references or defines a symbol sym-name whose definition cannot be uniquely resolved.

21. MODULE module-name MULTIPLY DEFINES SYMBOL sym-name

    Two definitions for the relocatable symbol sym-name have occurred on a common path. Or two definitions for an absolute symbol with the same name but different values have occurred.

22. SEGMENT seg-name HAS R-O SECTION

    An attempt has been made to allocate a read-only p-section in an overlay segment. The build continues with R-W attribute forced.

23. SEGMENT seg-name HAS ADDR OVERFLOW: ALLOCATION DELETED

    Within a segment, the program has attempted to allocate more than 32K. A map file is produced, but no task image file is produced.

24. ALLOCATION FAILURE ON FILE file-name

    The Task Builder could not acquire sufficient contiguous disk space to store the task image file. (If possible, delete unnecessary files on disk to make more room available.)

25. I/O ERROR ON OUTPUT FILE file-name

    This error may occur on any of the three output files.

26. LOAD ADDR OUT OF RANGE IN MODULE module-name

    An attempt has been made to store data in the task image outside the address limits of the segment.

27. TRUNCATION ERROR IN MODULE module-name

    An attempt has been made to load a global value greater than +127 or less that -128 into a byte. The low-order eight bits are loaded.

28. number UNDEFINED SYMBOLS SEGMENT seg-name

    The Memory Allocation File lists each undefined symbol by segment.

29. INVALID KEYWORD IDENTIFIER
    invalid-line

    The invalid-line printed contains an unrecognizable keyword.

30. OPTION SYNTAX ERROR
    invalid-line

    The invalid-line printed contains unrecognizable syntax.

31. TOO MANY PARAMETERS
    invalid-line

    The invalid-line printed contains a keyword with more parameters than required.

32. ILLEGAL MULTIPLE PARAMETER SETS
    invalid-line

    The invalid-line printed contains multiple parameters for a keyword which only allows a single parameter.

33. INSUFFICIENT PARAMETERS
invalid-line

The invalid-line contains a keyword with an insufficient number of parameters to complete the keyword meaning.

34. TASK HAS ILLEGAL MEMORY LIMITS

An attempt has been made to build a task whose size exceeds the partition boundary.

35. OVERLAY DIRECTIVE HAS NO OPERANDS
invalid-line

All overlay directives except .END require operands.

36. ILLEGAL OVERLAY DIRECTIVE
invalid-line

The invalid-line printed contains an unrecognizable overlay directive.

37. OVERLAY DIRECTIVE SYNTAX ERROR
invalid-line

The invalid-line printed contains a syntax error.

38. ROOT SEGMENT MULTIPLY DEFINED
invalid-line

The invalid-line printed contains the second .ROOT directive encountered. Only one .ROOT directive is allowed.

39. LABEL OR NAME IS MULTIPLY DEFINED
invalid-line

The invalid-line printed contains a name that has already appeared on a .FCTR, .NAME, or .PSECT directive.

40. NO ROOT SEGMENT SPECIFIED

The overlay description did not contain a .ROOT directive.

41. BLANK P-SECTION NAME IS ILLEGAL
invalid-line

The invalid-line printed contains a .PSECT directive that does not have a p-section name.

42. ILLEGAL P-SECTION ATTRIBUTE
invalid-line

The invalid-line printed contains a p-section attribute that is not recognized.

43. ILLEGAL OVERLAY DESCRIPTION OPERATOR
invalid-line

The invalid-line printed contains an unrecognizable operator in an overlay description.

44. TOO MANY NESTED .ROOT/.FCTR DIRECTIVES
invalid-line

The invalid-line printed contains a .FCTR directive that exceeds the maximum nesting level (32).

45. TOO MANY PARENTHESES LEVELS
invalid-line

The invalid-line printed contains a parenthesis that exceeds the maximum nesting level (32).

46. UNBALANCED PARENTHESES
invalid-line

The invalid-line printed contains unbalanced parentheses.

47. ILLEGAL BASE OR TOP ADDRESS OFFSET

The task is too large to fit into the space allowed by BASE= or TOP= keywords.

48. ILLEGAL LOGICAL UNIT NUMBER
invalid-line

The invalid-line printed contains a device assignment to a unit number larger than the number of logical units specified by the UNITS keyword or assumed by default if the UNITS keyword is not used.

49. ILLEGAL NUMBER OF LOGICAL UNITS
invalid-line

The invalid-line printed contains a number of logical unit greater that 250.

50. ILLEGAL NUMBER OF ACTIVE FILES SPECIFIED

51. ILLEGAL BASE OR TOP BOUNDARY VALUE
invalid line

52. ILLEGAL POOL USAGE NUMBER SPECIFIED
invalid line

The pool request is greater than 255 or it is zero.

53. ILLEGAL DEFAULT PRIORITY SPECIFIED
invalid-line

The invalid-line printed contains a priority greater than 250.

54. ILLEGAL ODT OR TASK VECTOR SIZE

SST vector size specified greater than 32 words.

55. ILLEGAL FILENAME
invalid-line

The invalid-line printed contains a wild card (*) in a file specification. The use of wild cards is prohibited.

56. ILLEGAL DEVICE/VOLUME
invalid line

The device/volume string is too long.

57. LOOKUP FAILURE ON FILE filename
invalid-line

The invalid-line printed contains a filename which cannot be located in the directory.

58. ILLEGAL DIRECTORY
invalid-line

The invalid-line printed contains an illegal UFD.

59. INCOMPATIBLE REFERENCE TO A LIBRARY P-SECTION p-sect-name

A task has attempted to reference more storage in a shareable global area than exists in the shareable global area definition.

60. ILLEGAL REFERENCE TO LIBRARY P-SECTION p-sect-name

A task has attempted to reference a p-sect-name existing in a resident library (shareable global area) but has not named the library in a COMMON or LIBR keyword.

61. RESIDENT LIBRARY MEMORY ALLOCATION CONFLICT
keyword-string

One of the following problems has occurred:

1. More than three shareable global areas have been specified.

2. The same shareable global area has been specified more than once.

3. Shareable global areas whose memory allocations overlap have been specified.

4. BASE or TOP specifications conflict.

62. LOOKUP FAILURE RESIDENT LIBRARY FILE
    invalid-line

    No symbol table or task image file found for the shareable global area on SY0 under UFD [1,1].

63. INVALID ACCESS TYPE
    invalid-line

    Requested access to shareable global area was not RW or RO.

64. ILLEGAL PARTITION/COMMON BLOCK SPECIFIED
    invalid-line

    User defined base or length not on 32 word bound or user defined length = 0.

65. NO MEMORY AVAILABLE FOR LIBRARY library-name

    Insufficient virtual memory available to cover total memory needed by referenced shareable global areas.

66. PIC LIBRARIES MAY NOT REFERENCE OTHER LIBRARIES
    invalid-line

67. ILLEGAL APR RESERVATION

    APR specified on COMMON or LIBR keyword that is outside the range 0-7.

68. I/O ERROR LIBRARY IMAGE FILE

    An I/O error has occurred during an attempt to open or read the Task Image File of a shareable global area.

69. LIBRARY REFERENCES UNDEFINED LIBRARY

    A shareable global area specified by LIBR or COMMON references another area which is undefined.

70. not used.

71. INVALID APR RESERVATION

    APR specified on a LIBR or COMMON keyword for an absolute shareable global area.

72. COMPLEX RELOCATION ERROR - DIVIDE BY ZERO: MODULE
    module-name

    A divisor having the value zero was detected in a complex expression. The result of the divide was set to zero. (Probable cause - division by an undefined global symbol.)

73. WORK FILE I/O ERROR

I/O error during an attempt to reference data stored by the Task Builder in a work file. Possibly an attempt to extend the file when no more space is available on the volume.

74. LOOKUP FAILURE ON SYSTEM LIBRARY FILE

The Task Builder cannot find the system Library (SY0:[1,1]SYSLIB.OLB) file to resolve undefined symbols.

75. UNABLE TO OPEN WORK FILE

Work file device is not mounted or has not been initialized as Files-11, or there is no space on the volume.

76. NO VIRTUAL MEMORY STORAGE AVAILABLE

Maximum permissible size of the work file exceeded (no recovery).

77. MODULE module-name NOT IN LIBRARY

The Task Builder could not find the module named on the LB switch in the library.

78. INCORRECT LIBRARY MODULE SPECIFICATION
invalid-line

The invalid-line contains a module name with a non-Radix-50 character.

79. LIBRARY FILE filename HAS INCORRECT FORMAT

A module has been requested from a library file that has an empty module name table.

80. RESIDENT LIBRARY IMAGE HAS INCORRECT FORMAT
invalid-line

The invalid-line specifies a shareable global area that has one of the following problems:

1. The library file image has a header.

2. The shareable global area references another shareable global area with invalid address bounds (i.e., not on 4K boundary).

3. The shareable global area has invalid address bounds.

81. PARTITION partition-name HAS ILLEGAL MEMORY LIMITS

    The user has attempted to build a privileged task whose length exceeds 16K.

82. not used.

83. ABORTED VIA REQUEST
    input-line

    The input-line contains a request from the user to abort the task build.

84. not used

85. END OF FILE REACHED BEFORE .END DIRECTIVE IN file-name

    The overlay description file named in this message does not contain a .END directive as required.

APPENDIX M

BATCH COMMAND SUMMARY AND ERROR MESSAGES

M.1 COMMAND SUMMARIES

$COBOL

```
$COBOL[/sws] [src/SOURCE] [obj/OBJECT] [list/LIST] [!comments]
```

$COPY

```
$COPY [dev:]filename.type/OUTPUT [dev:]filename1.type[+dev:filename2.type[/INPUT]][!comments]
```

$CREATE

```
$CREATE filename.type [!comments]
```

$DATA

```
$DATA [!comments]
```

$DELETE

```
$DELETE filename1 [filename2 ... filenamen] [!comments]
```

$DIRECTORY

```
$DIR [filename.type/DIRECTORY] [filename1.type1[/INPUT]] [!comments]
```

$DISMOUNT

```
$DISMOUNT lun:  [!comments]
```

$EOD

```
$EOD
```

$EOJ

```
$EOJ
```

$FORTRAN

```
     $FORTRAN[/sws] [src/sw] [obj/OBJECT] [list/LIST] [!comments]
```

$JOB

```
     $JOB[/switches] [!comments]
```

$LINK

```
     $LINK[/switches] [obj/OBJECT] [lib/LIBRARY] [map/MAP] [task/EXECUTE] [!comments]
```

$MACRO

```
     $MACRO[/sws] [src/SOURCE] [obj/OBJECT] [list/LIST] [!comments]
```

$MESSAGE

```
     $MESSAGE[/WAIT] message
```

$MOUNT

```
     $MOUNT dev[/PHYSICAL] [/switch] [/VOLUME=x] [/VID=x] lun:/LOGICAL [!comments]
```

$PRINT

```
     $PRINT[filespec1 [filespec2 ... filespecn]
```

$RUN

```
     $RUN filespec
```

$SORT

```
     $SORT[/sws] [infile/INPUT] [outfile/OUTPUT] [specfile/SPECIFICATION] [!comments]
```

M.2 BATCH ERROR MESSAGES

BAT -- ERROR MISSING JOB CARD

1. A $JOB card did not follow in sequence after a $EOJ.

2. A $JOB card was omitted at the beginning of the input file.

3. The submitted $JOB command has an error (i.e., MCR mode was specified from a non-privileged terminal).

In each of the above cases, the error message is printed and all of the images on the input file are flushed until the next valid $JOB command is encountered.

Place a $JOB card or command at the appropriate place or correct the erroneous $JOB command and resubmit the job.

BAT -- ERROR READING INPUT FILE

A physical error has been encountered while the input job file is read, or an input file contains extraneous records following a $EOJ command.

Verify, by some copy or editing operation, that the file is error free, and resubmit the job. If extraneous records follow an $EOJ but all valid jobs have been processed correctly, delete the extraneous records before the next submission of the batch job.

BAT -- ERROR WRITING OUTPUT FILE

An attempt to issue a write to a disk file has failed due to a lack of file space on the disk

Delete superfluous files and resubmit the job.

APPENDIX N

SYSTEM STANDARD ERROR CODES

CODE	DESCRIPTION
-01	Unavailable Pool node or Bad parameters
-02	Invalid function code
-03	Device not ready
-04	Parity error on device
-05	Un-assigned LUN
-06	Handler task not resident or Illegal user buffer
-07	Device not attached
-08	Device already attached
-09	Device not attachable
-10	End of file detected
-11	End of volume detected
-12	Write attempted to locked unit
-13	Data overrun
-14	Send/Receive failure
-15	Request terminated
-16	Privilege violation
-17	Not used
-18	Illegal overlay request
-19	Odd byte count (or virtual address)
-20	Logical block number too large

SYSTEM STANDARD ERROR CODES

CODE	DESCRIPTION
-21	Invalid UDC module #
-22	UDC connect error
-23	Caller's nodes exhausted
-24	Device full
-25	Index file full
-26	No such file
-27	Locked from file access
-28	File header full
-29	Accessed for write
-30	File header checksum failure
-31	Write attribute data format error
-32	File processor device read error
-33	File processor device write error
-34	File already accessed on LUN
-35	File ID, file number check
-36	File ID, sequence number check
-37	No file accessed on LUN
-38	File was not properly closed
-39	Open - No buffer space available for file
-40	Illegal record size
-41	File exceeds space allocated; no space
-42	Illegal operation on file descriptor block
-43	Bad record type
-44	Illegal record access bit set
-45	Illegal record attributes bit set
-46	Illegal record number - Too large
-47	Multiple block Read/Write - not implemented yet
-48	RENAME - 2 different devices

SYSTEM STANDARD ERROR CODES

CODE	DESCRIPTION
-49	RENAME - new file already in use
-50	Bad directory file
-51	Can't rename - old file system
-52	Bad directory syntax
-53	File already open
-54	Bad file name
-55	Bad device name
-56	Bad block on device
-57	ENTER - duplicate entry in directory
-58	Not enough stack space (FCS or FCP)
-59	Fatal hardware error on device
-60	File ID was not specified
-61	Illegal sequential operation
-62	End of tape detected
-63	Bad version number
-64 to -94	Unused
-95	Invalid priority (>250.)
-96	Invalid LUN
-97	Invalid event (>64.)
-98	Part of DPB out of user's space
-99	DIC or DPB size invalid

INDEX

RSX-11D User's Guide
DEC-11-OXDUA-B-D

READER'S COMMENTS

NOTE: This form is for document comments only. Problems with software should be reported on a Software Problem Repcrt (SPR) form.

Did you find errors in this manual? If so, specify by page.

____
____
____
____
____
____

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

____
____
____
____
____
____

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

____
____
____
____
____
____

Please indicate the type of user/reader that you most nearly represent.

- □ Assembly language programmer
- □ Higher-level language programmer
- □ Occasional programmer (experienced)
- □ User with little programming experience
- □ Student programmer
- □ Non-programmer interested in computer concepts and capabilities

Name ______________________________ Date ____________________

Organization ____

Street ____

City ____________________ State __________ Zip Code ______________
or
Country

If you require a written reply, please check here. □

Please cut along this line.

-- **Fold Here** --

-- **Do Not Tear - Fold Here and Staple** --

Postage will be paid by:

digital

Software Communications
P. O. Box F
Maynard, Massachusetts 01754

digital

digital equipment corporation

Printed in U.S.A.