pdp11

RSX-11M Operator's Procedures Manual

Order No. AA-2567D-TC

digital

RSX-11M
Operator's
Procedures Manual

Order No. AA-2567D-TC

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754

digital equipment corporation · maynard, massachusetts

RSX-11M Operator's
Procedures Manual
DEC-AA-2567D-TC

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 only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

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
ASSIST-11
DECsystem-10
DECtape
DIBOL
EDUSYSTEM
FLIP CHIP
FOCAL
INDAC
LAB-8
DECSYSTEM-20
RTS-8
MASSBUS
OMNIBUS
OS/8
PHA
RSTS
RSX
TYPESET-8
TYPESET-11
TMS-11
ITPS-10

6/78-14

CONTENTS

CONTENTS (Cont.)

CONTENTS (CONT.)

FIGURES

TABLES

PREFACE

0.1 MANUAL OBJECTIVES AND READER ASSUMPTIONS

The RSX-11M Operator's Procedures Manual provides all the required information to operate an RSX-11M system. Although the manual is primarily self-contained, the reader should be familiar with the Introduction to RSX-11M and the RSX-11M System Generation Manual to understand the function of this manual in the context of RSX-11M as a whole.

The manual does not attempt to train operators. The reader is assumed to be familiar with computer operating procedures, both real-time and batch, and to have had experience operating both the computer console and the terminal devices supported by RSX-11M for use as operator consoles.

Finally, the term operator is used broadly to define anyone who chooses to interface directly with RSX-11M to perform a task. The manual uses a broad definition because of the varied settings in which RSX-11M operates; some settings do not require an operator in the conventional sense of the term.

Because RSX-11S is a subset system compatible with RSX-11M, this manual can also be used as a reference document for RSX-11S operation.

0.2 STRUCTURE OF THE DOCUMENT

Chapter 1 introduces basic RSX-11M concepts that pertain to the operator's interface with the system.

Chapter 2 describes the procedures and conventions that an operator needs to know in order to control the system online from a terminal.

Chapter 3 discusses some basic concepts of the RSX-11M file system and highlights aspects of file handling that directly relate to operator functions.

Chapter 4 describes operating procedures for all the bootstraps supported by RSX-11M.

Chapter 5 provides detailed specifications of all the Monitor Console Routine (MCR) commands. The command specifications are in alphabetical order.

Chapter 6 defines indirect command files and describes the MCR indirect command file processor.

Chapter 7 lists in alphabetical order all the MCR and Task Termination Notification routine (TKTN) messages that can be returned by the system.

Appendix A consists of a summary, in alphabetical order, of all MCR command formats.

Appendix B lists the subset of MCR commands that comprises Basic MCR, the subset supported by RSX-11S.

0.3 ASSOCIATED DOCUMENTS

Other documents related to RSX-11M are described briefly in the RSX-11M/RSX-11S Documentation Directory. The directory defines the intended readership of each manual in the RSX-11M/RSX-11S set and provides a synopsis of each manual's contents.

CHAPTER 1

INTRODUCTION

This chapter introduces basic RSX-11M concepts pertaining to operator interface with the system.

RSX-11M is a real-time, multiprogramming operating system that controls the sharing of system resources among any number of user-prepared tasks. Users can store tasks on file-structured volumes, install them into the system, and subsequently run them by means of commands issued to the Monitor Console Routine (MCR). MCR provides the language interface between the operator and the system. (In the context of this manual, an operator is anyone who interfaces directly with RSX-11M.)

Once tasks have been installed and activated, they compete for system resources on the basis of priority and resource availability. The RSX-11M Executive manages this competition to create an efficient multiprogramming environment that also allows for quick response to real-time demands. The following sections describe some of the criteria that determine how multiprogramming operates in RSX-11M.

1.1 PARTITIONS

A task runs in a predetermined contiguous area of memory called a partition. A partition has the following characteristics:

1. A name,

2. A defined size,

3. A fixed starting address, and

4. A defined type.

The relationship between a task and the partition in which it runs depends on whether the system is mapped or unmapped. In unmapped systems, a task can run only in a partition that has the same starting address as the partition for which the task image was created by the Task Builder. In mapped systems, however, a task can run in any partition that is large enough to contain it. (See the RSX-11M Task Builder Reference Manual for more information about mapped and unmapped systems.)

1.1.1 Partition Types

RSX-11M supports two types of partition in which tasks can execute:

1. System-controlled and
2. User-controlled.

In a system-controlled partition, the Executive allocates available space to accommodate as many tasks as possible at any one time. This allocation may involve shuffling resident tasks to arrange available space into a contiguous block large enough to contain a requested task. Only mapped systems support system-controlled partitions.

A user-controlled partition, however, is exclusively allocated to one task at a time. This type of partition is supported by both mapped and unmapped systems.

In both types of partitions, the Executive can temporarily move a resident task out to a disk (checkpoint) in order to make space available for another task (see Section 1.2.1).

1.1.2 Subpartitions

A user-controlled partition can be subdivided into as many as seven non-overlapping subpartitions. Like its parent main partition, a subpartition can contain only one task at a time. Since the subpartitions occupy the same physical memory as the main partition, tasks cannot be simultaneously resident in both the main partition and one or more subpartitions. But since each subpartition can contain a task, up to seven tasks can potentially run in parallel within a pre-empted main partition.

The purpose of subpartitioning is to reclaim large storage areas in unmapped systems. For example, when a large task that requires a main partition is either no longer active or can be pre-empted (checkpointed), subpartitioning allows the partition space to be used for a number of smaller real-time tasks.

1.2 MULTIPROGRAMMING

Tasks compete for system resources (including memory) on the basis of priority and resource availability. The priority of a task is determined by a number that has been assigned to the task either when it was created by the Task Builder or when it was installed. The number is in the range 1 to 250 (decimal), where a higher number indicates a higher priority. The highest priority task, that has access to and can use all the resources it needs, has control of the CPU.

A task often becomes blocked when it requests system services; for example, it can block itself as it waits for an I/O transfer to complete. While the task is blocked in this way, the Executive looks for another task to use the CPU. The chosen task will be the one that has the highest priority and has access to all the resources it needs.

1.2.1 Checkpointing

Checkpointing provides a means by which tasks not currently resident in memory gain access to the CPU. In some instances, an activated task cannot compete for the processor because the partition in which it was installed is fully occupied. If the partition contains a task that has a lower priority and has been declared checkpointable, the Executive can move that task out of memory to mass storage in order to make room for the higher priority task. When the latter task is finished, the pre-empted task is reinstated and continues processing from the point at which it was interrupted. This roll-out, roll-in process is called checkpointing.

A system generation option allows the Executive to checkpoint a task waiting for input from a terminal so that another task can run until the terminal input completes. The Executive pre-empts the interim task at completion of the input and returns the previously waiting task to memory.

1.2.2 Example of a 16K Unmapped System

Figure 1-1 illustrates the memory layout of a sample 16K unmapped system. The Executive region, requiring 8K, consists of the Executive and the user-controlled main partition named SYSPAR. This partition contains the file system (F11ACP), MCR, and the Task Termination Notification routine (TKTN). The file system is checkpointable and has a lower priority than MCR or TKTN. Thus, if the file system is running and an operator requests MCR, the file system will be checkpointed, and MCR will be loaded and initiated.

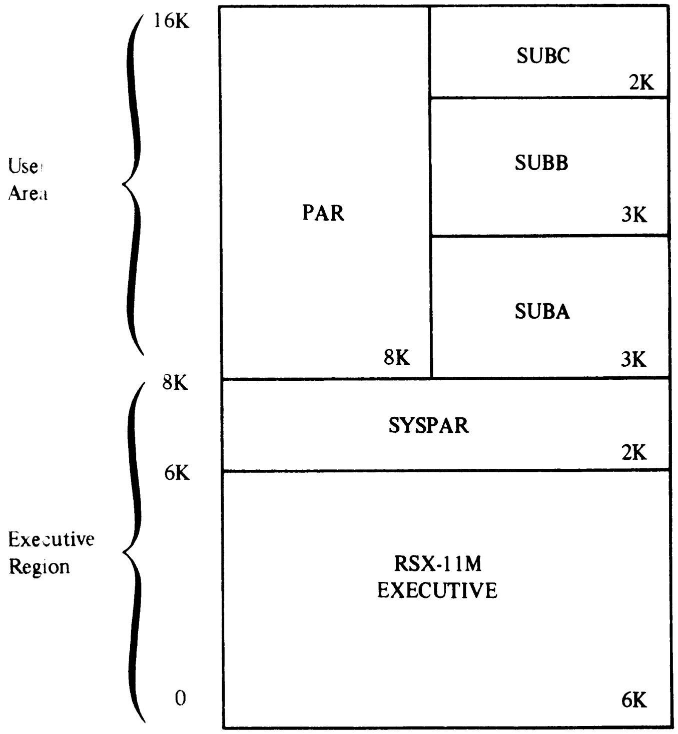

Figure 1-1 Sample Unmapped System Memory Layout

The user area is composed of a user-controlled main partition named PAR8K, 8K words in length, and three subpartitions, named SUBA, SUBB and SUBC. The partition PAR8K is used for program preparation tasks such as the language processors and the Task Builder. These programs will usually have a low priority and may be checkpointable.

The three subpartitions are for real-time tasks. Any time one of these tasks needs the processor and has a higher priority than the task occupying the main partition, the contents of the main partition will be checkpointed (if the occupant task is checkpointable).

If the partitions SUBA, SUBB, SUBC, and SYSPAR are occupied and the tasks in them are ready to run, to which task does the Executive give the processor? As stated earlier, the choice depends entirely on task priority, even for the SYSPAR partition. Hence, the highest priority task will run next, regardless of where it resides in memory.

1.3 TASK OPERATION

An RSX-11M user performs four basic steps to create a task and prepare it for execution:

1. Enters a program in a supported source language (MACRO-11, FORTRAN IV, FORTRAN IV-PLUS, COBOL, or BASIC) through an editor.

2. Submits the source code to the applicable translator, which produces an object file.

3. Submits the object file to the Task Builder, which produces a task-image file.

4. Finally, issues the MCR command Install to install the task image file in the system.

The output files that result from Steps 1, 2, and 3 are each stored on a Files-11 volume (see Section 3.1).

1.3.1 Installing a Task

When an operator installs a task (by issuing the MCR command Install or a form of the command Run), the system records a number of task parameters in a system-resident table called the System Task Directory (STD). The recorded parameters include the name and length of the task, and the address on the volume that contains the task image. An installed task is defined as a task that has an entry in the STD; it is neither resident in memory nor competing for system resources. The system considers it to be dormant until a request is made for it to be activated.

1.3.2 Running an Installed Task

An installed, dormant task can be activated either by:

1. The MCR Run command or

2. An Executive directive issued from within another task; either RQST$ (Request) or RUN$ (Run).

When the Executive receives either form of request, it performs a series of actions:

1. Allocates the necessary resources,
2. Brings the task into memory (if there is space available in its partition*), and
3. Places it in active competition for system resources with other resident tasks.

If the partition in which a task is installed is fully occupied and no resident task can be checkpointed, the task is placed in a queue of other activated tasks, each waiting for space to become available in its partition. Note that the number of installed, dormant tasks can, and usually will, far exceed the number of active tasks.

1.3.3 **Dormant and Active Tasks**

The concept of dormant tasks and active tasks is important in a real-time system. A dormant task (described in Section 1.3.1) uses very little memory; and yet when the task is needed to service a real-time event, the Executive can quickly and efficiently introduce it into active competition for system resources. An installed task's STD entry enables this quick response because it contains all the parameters the system needs to retrieve the requested task.

* The Executive may checkpoint a lower priority, checkpointable task, if there is one, to make room for the requested task.

CHAPTER 2

PROCEDURES AND CONVENTIONS

This chapter describes the procedures and conventions that permit an operator to control the system online from a terminal. The topics include:

- How to operate a terminal (Section 2.1)
- How to issue MCR commands (Section 2.2)
- How to manage peripheral devices (Section 2.3)
- How to operate in a multiuser protection system (Section 2.4)
- How the MCR Dispatcher services command input (Section 2.5)
- How to shutdown an RSX-11M system (Section 2.6)

2.1 TERMINALS

2.1.1 Special and Control Characters

The operator controls terminal operation by means of special character keys and control characters. The relevant special character keys are TAB, ESC (or ALT), CR (or RETURN) and RUBOUT (or DELETE). A control character is produced when an operator types a letter key while pressing CTRL (control). In this manual, a control character is written CTRL/X where X is the variable letter key.

When an operator types CTRL/U or CTRL/Z, the Executive performs the related function and displays ^U and ^Z at the issuing terminal. When CTRL/C is typed, the explicit MCR prompt (MCR>) is displayed. For other control characters, however, the Executive does not return a display.

Table 2-1 lists all the special character keys, control characters, and describes the function of each.

Table 2-1
Special Character Keys and Control Characters

Character	Description
CR or RETURN	The CR (Carriage Return) key terminates a line of input and advances the carriage or cursor to position 1 on the next line.
ESC or ALT	The ESCape or ALTmode key terminates a line of input without moving the carriage or cursor. When used to terminate an MCR command, it suppresses the default MCR prompt. See the descriptions of the commands OPEN and RUN (in Chapter 5), which have special uses for the ESC or ALT key. In systems that have generated the appropriate optional facility, the ESC key can be the first character in an escape sequence (see the <u>RSX-11M I/O Drivers Reference Manual</u> and the description of the SET /ESCSEQ command).
RUBOUT or DELETE	The RUBOUT or DELETE key deletes the last character typed at the terminal, and further contiguous characters if the key is pressed repeatedly. The first RUBOUT typed deletes a character and prints a backslash (\) followed by a display of the deleted character. Subsequent deletions cause the system to display the deleted character. Then, when the operator enters the first non-RUBOUT character, the system prints a terminating backslash followed by the character typed. For example: First RUBOUT Second RUBOUT Third RUBOUT First non-RUBOUT MISTKAE\EAK\AKE The result is MISTAKE. On a CRT terminal that supports backspace (a system generation option) and has been declared to be a CRT by the SET /CRT=ttn: command, the RUBOUT or DELETE key moves the printing position one space to the left and erases any character displayed in that position. For example, to correct the word MISTKAE, first press RUBOUT or DELETE three times to erase E, A and K respectively. The result is MIST. Then type AKE to get MISTAKE. (Note that when RUBOUT deletes a character in the last character position on the right of the screen, the last character appears to remain and the next to last character disappears. However, the system has in fact deleted the last character; the user can type CTRL/R for verification.)

(continued on next page)

Table 2-1 (Cont.)
Special Character Keys and Control Characters

Character	Description
CTRL/C	CTRL/C typed either as the first character in the line or when the terminal is sending data causes MCR to prompt for command input with the explicit prompt MCR>. (Two exceptional effects of CTRL/C are described below.) If a task is prompting when the operator types CTRL/C followed by a carriage return, MCR displays its explicit prompt, then returns control to the interrupted task after receiving a single line of input. (An exception arises when an attached task specifies an Asynchronous System Trap [AST] for unrequested characters. In this case, CTRL/C causes an AST to occur thereby gaining the program's attention rather than MCR's.) See Section 2.1.3 for a more detailed discussion of the use of CTRL/C. Exceptional effects: ● If the last character entered at the terminal was CTRL/S, CTRL/C additionally performs the function of a CTRL/Q. ● If a terminal has been set to "hold screen" mode by the /HOLD option of the MCR SET command, CTRL/C has the effect of a SET /NOHOLD command; that is, CTRL/C disables hold screen mode. (This feature is a system generation option.)
CTRL/I or TAB	CTRL/I (or TAB) moves the current print position to the next horizontal tab stop on the line. The system establishes tab stops at every eighth character position in the line.
CTRL/K	CTRL/K causes a vertical tab by performing four line feeds.
CTRL/L	CTRL/L causes a form feed (but does not perform paging). The action appears on the terminal as eight line feeds.
CTRL/O	CTRL/O alternately suppresses and resumes the display of output at the terminal. The effect of CTRL/O depends on the state of the terminal when the operator typed the character. The system controls the function of CTRL/O internally by means of a single bit associated with the terminal called the disable output bit. When the bit is set, the system disables output to the terminal. When the bit is cleared, the system resumes the output display. Initially, the bit is clear; the first CTRL/O typed causes the bit to be set. The next CTRL/O typed then clears it again. Thus the net effect of successive CTRL/O characters is to alternately stop and start the display of output at the terminal. The system discards characters directed to a terminal that has disabled the display of its output.

(continued on next page)

Table 2-1 (Cont.)
Special Character Keys and Control Characters

Character	Description
	If more than one task is sending output to a terminal, it is essential that CTRL/O affect only the task delivering output at the time the operator types CTRL/O. The system handles this problem as follows: The disable bit is cleared for attached terminals at: ● The issuance of an Attach QIO directive, ● The issuance of a Detach QIO directive, ● The issuance of a Write QIO directive that has the "clear CTRL/O" subfunction bit set, ● A request for input, and ● The arrival of unsolicited input. For unattached terminals, the bit is also cleared each time a task initiates a request to output to the terminal. Given these conditions for clearing the disable bit, and the fact that entering CTRL/O always complements the bit, CTRL/O affects the entering terminal as follows: At an attached terminal, an operator can stop output by typing CTRL/O. The interrupted output stream will be discarded until the next: ● Detach, ● Solicited input, ● Unsolicited input, or ● CTRL/O. At unattached terminals, CTRL/O stops output only for the current buffer, since, at the next I/O initiation, the system always clears the disable output bit for unattached terminals.
CTRL/Q	CTRL/Q typed after a CTRL/S resumes output suspended by the previous CTRL/S. (This feature is a system generation option.)
CTRL/R	Typing CTRL/R before typing a line terminator causes the system to retype the current line on a new line, omitting any deleted characters. If the current line is empty, CTRL/R performs a carriage return and line feed. For example: MISTKAE\EAK\AKE <CTRL/R> MISTAKE (This feature is a system generation option.)

(continued on next page)

Table 2-1 (Cont.)
Special Character Keys and Control Characters

Character	Description
CTRL/S	Typing CTRL/S while the terminal is receiving output suspends additional output until the operator types CTRL/Q or CTRL/C. The suspended output is merely delayed, not lost (see the description of CTRL/O). The combined functions of CTRL/Q and CTRL/S are convenient when using a CRT terminal. (This feature is a system generation option.)
CTRL/U	Typing CTRL/U before typing a line terminator causes the previously typed characters to be deleted back to the beginning of the line. The system responds with a carriage return and line-feed so that the line can be retyped. CTRL/U is echoed as ^U.
CTRL/Z	Typing CTRL/Z indicates end-of-file. It is used as a signal to system tasks, such as MACRO-11, PIP, and TKB to indicate that the user is finished and the subject task may exit. CTRL/Z is echoed as ^Z.

2.1.2 Terminal Characteristics

In RSX-11M, a variable number of terminals can operate concurrently; each terminal operates independently of others in the system so that each can run one or more different tasks. Thus the PIP utility, the MACRO-11 Assembler, and MCR, for instance, can all run from different terminals at the same time, as long as the required memory is available.

If an RSX-11M system supports multiuser protection (a system generation option), each user must log onto a terminal before issuing further MCR or task commands (other than the HELP command). Section 2.4.1 describes how to log on and off a terminal in a multiuser protection system.

The following two sections describe certain other terminal characteristics that determine how the operator and/or the system can use each terminal at a given time.

2.1.2.1 **Terminal Privilege** - An operator can issue privileged commands only at a privileged terminal. (The description of each MCR command in Chapter 5 states whether it is privileged or nonprivileged.) In RSX-11M systems that support multiuser protection, individual users are either privileged or nonprivileged. When a user logs on (see Section 2.4.1.1), the terminal assumes the privilege status of the user logging on. In other RSX-11M systems, a terminal's privilege status is determined initially at system generation. Subsequently, an operator can use the MCR command SET (see Section 5.5) at a privileged terminal to modify the privilege status of any other terminal connected to the system.

CAUTION

Operators must take great care when more than one terminal is privileged. Privileged commands can destructively interfere with system operation and with each other.

2.1.2.2 **Attached and Unattached Terminals** - A terminal is attached when all its input and its output are directed to or from a task other than MCR. For example, when an operator invokes a task (such as the text editor EDI), the task usually attaches to the terminal so that no other task can use the terminal for I/O. When the programmer wishes to exit (for example, by typing CTRL/Z), the task is detached from the terminal, which normally directs subsequent input to MCR.

Since an attached terminal directs all input to the dedicated task, the operator must either exit from the task or gain MCR's attention before attempting to communicate with MCR. (See Section 2.1.3 for a description of CTRL/C as a means of temporarily gaining MCR's attention.)

2.1.2.3 **Slave Terminals** - When an installation needs to dedicate a terminal exclusively to one or more tasks, an operator issues an MCR Set command option that sets the terminal to slave status. (A special I/O function issued by a task can also set a terminal to slave status.) The difference between a slave terminal and an attached terminal is that a slave terminal rejects all unsolicited input, including CTRL/C, but excepting CTRL/O, CTRL/Q, and CTRL/S. Attached terminals accept certain forms of unsolicited input.

Until the operator issues another MCR command (or a task issues another special I/O function) to delete the slave status, the terminal can only be used to communicate with tasks soliciting input from the terminal. Slave terminals are often dedicated to real-time applications.

2.1.3 Input Prompts

There are three types of display that indicate that a terminal is waiting for input:

1. The default prompt (>),
2. The task prompt (tsk>) and
3. The MCR prompt (MCR>).

2.1.3.1 **The Default Prompt** - A default prompt (the character > in position one of a new line) indicates that the terminal is ready to accept unsolicited input. The term unsolicited implies that no specific task has requested input from the terminal. Whatever the operator types in response to the default prompt is normally directed to MCR. (An abnormal situation can occur, however, if a task either solicits input or attaches to the terminal without displaying a task prompt. A soliciting or attached task might receive input that the operator intended for MCR. See Section 2.1.3.2 below.) When MCR receives unsolicited input, it identifies, analyzes, and responds appropriately to the input. Section 2.5 discusses how MCR (using the MCR Dispatcher) responds to commands.

2.1.3.2 **The Task Prompt** - When MCR activates a task to service a command, the task normally prompts:

```
tsk>
```

where tsk is a 3-character task name. (All DIGITAL system tasks identify themselves by a 3-character prompt, and user tasks should do the same.)

As long as a prompting task has attached to a terminal, the task receives all solicited input entered at the terminal until the task detaches. (The terminal discards all unsolicited input.) An unattached task, however, encounters the possibility that another task could solicit input from the terminal before the operator types a response to the prompt. The intervening task would then receive the input intended for the task that displayed the prompt. To avoid this situation, every task intending to solicit input from a terminal should perform the following:

- Attach to the terminal,
- Prompt with a properly formatted identifier (that is, tsk>), and then
- Solicit input.

2.1.3.3 **The MCR Prompt** - A terminal always sends to MCR any input typed in response to the explicit prompt:

```
MCR>
```

The operator can invoke the explicit MCR prompt by typing:

```
CTRL/C
```

or (as shown in the second case described below):

```
CTRL/C <CR>
```

(CTRL/C does not invoke the MCR prompt in an exceptional case described in the second note below. In addition, the description of CTRL/C in Table 2-1 describes two further situations in which CTRL/C does not obtain the MCR prompt.)

The following notes explain various uses of the explicit MCR prompt:

- When a terminal is displaying the default prompt (see Section 2.1.3.1), there is no guarantee that input typed in response will be directed to MCR. To ensure that MCR does receive the input, the operator can press CTRL/C to invoke the MCR prompt. After MCR processes the single line of input,* the terminal resumes displaying the default prompt.

- When a terminal is displaying a task prompt (tsk>), the user can invoke MCR by typing

  ```
  CTRL/C <CR>
  ```

  After MCR processes the single line of input* received, the terminal again displays the original task prompt.

  NOTE

  If an attached task specifies an AST on receiving unsolicited input, CTRL/C causes an AST to occur, thereby gaining the task's attention rather than MCR's. Consequently, the MCR prompt does not appear.

- When any terminal is receiving output from a task, the operator can interrupt the output by typing CTRL/C. At the completion of the current I/O operation, the terminal displays the explicit MCR prompt. After MCR processes the single line of input* directed to it, the system resumes the interrupted output.

2.2 MCR COMMAND STRINGS

To issue an MCR command, an operator types a command string in response to a default (>) or explicit (MCR>) MCR prompt. A command string consists of the following three components:

- A command name,
- Any required parameters, and
- A line terminator.

All the examples given in this manual use the default prompt.

2.2.1 Command Names

Chapter 5 describes in detail all the MCR commands needed to control system operation. Each description includes the full command title and the 3-letter command name recognized by MCR. (An exceptional command is Help, which must be entered in full to distinguish it from the Hello command.)

* The single line of input received by MCR may in fact invoke a task that proceeds to issue its own prompt. The prompt of the invoked task can, however, also be answered by a CTRL/C to invoke the MCR prompt. The operator can then request another task that issues its own prompt, and so on.

For example, an operator can type the command entitled Cancel as follows:

```
>CAN taskname  <CR>
```

where taskname is the command parameter and <CR> stands for carriage return, the line terminator (see Section 2.2.3 below). As in this example, the 3-letter command name often consists of the first three letters of the command title. Some commands, however, have 3-letter acronyms as command names. For instance, the names for the Assign and User File Directory commands are ASN and UFD, respectively.

Although MCR recognizes only the first three letters of a typed command name, the operator can concatenate any number (up to the maximum buffer length of the terminal) of alphanumeric characters to those three initial letters. To make the intent of the command clear to someone reading a printed copy of command input, the operator could type:

```
>CANCEL taskname <CR>
```

A space or a tab signals the end of the command name; therefore, no spaces or tabs are allowed within the name itself. For example,

```
>OPENREGISTER address
```

is permissible, whereas

```
>OPEN REGISTER address
```

is not.

2.2.2 Command Parameters

The operator separates the command name from any required parameters by one or more spaces or tabs. Typically, a parameter is a task name, the name of a file, or a device specification. A parameter may be modified by either a switch or a keyword. (Switches and keywords are defined in Sections 3.2 and 5.2.2, respectively.)

The command descriptions in Chapter 5 specify the parameters required (if any) for each command and indicate whether the parameters themselves need to be modified.

2.2.3 Line Terminators

Pressing carriage return (<CR>) or the ALTmode or ESCape key (<ESC>) terminates a command string. When <CR> is pressed, MCR returns the default prompt after processing the command. Except when <ESC> has special significance as a terminator (with the Run and Open Register commands, for example), MCR suppresses the default prompt after processing a command string terminated by <ESC>. (MCR accepts subsequent unsolicited input even though the default prompt does not appear.)

2.2.4 Correcting Errors

Before terminating a line, the operator can correct typing errors or change the line completely by using the RUBOUT or DELETE key, or CTRL/U. See Section 2.1.

2.2.5 Command Input Errors

When MCR detects an error in command input, it returns an appropriate error message (prefixed by the command name) at the issuing terminal. Messages that are unique to a given command are listed in the command's description in Chapter 5. Chapter 7 lists all command error messages in alphabetical order.

2.3 DEVICES

A primary operations function is to manage all connected peripheral devices to maintain efficient system performance. This section provides the basic information needed to accomplish this function. (Devices within systems that support multiuser protection are subject to a special set of MCR functions, described in Section 2.4.2.)

At system generation, the operator explicitly describes all peripheral devices attached to the system (see the RSX-11M System Generation Manual). Each device has a unique identifier consisting of a 2-character ASCII device name and an optional 1- or 2-digit octal unit number, followed by a colon (:) (for example, DK1:, MT2:). If the unit number is omitted, the system defaults to unit 0; thus LP: indicates line printer 0.

Table 2-2 lists the ASCII device names supported by RSX-11M. The table also lists the RSX-11M pseudo devices, which are explained in Section 2.3.3. The hardware controller designations appear in parentheses.

Table 2-2
RSX-11M Peripheral, Pseudo, and Null Devices

Peripheral Devices	Device-unit
Analog-to-Digital Converter (AD01-D)	ADnn:
(AFC11)	AFnn:
Laboratory Peripheral System (AR11)	ARnn:
Card Reader (CR11)	CRnn:
Cassette (TA11)	CTnn:
DECtape (TC11)	DTnn:
Disk (RH11/RH70/RP04/RP05/RP06)	DBnn:
(RF11)	DFnn:
(RK11)	DKnn:
(RK611)	DMnn:
(RL01)	DLnn:
(RM03)	DRnn:
(RP11)	DPnn:
(RH11/RH70/RS03/RS04)	DSnn:

(continued on next page)

Table 2-2 (Cont.)

RSX-11M Peripheral, Pseudo, and Null Devices

Peripheral Devices	Device-unit
(RX11)	DXnn:
Graphics Display Processor and Scope (VT11/VS60)	GRnn:
Industrial Control System Local and Remote (ICS/ICR-11)	ICnn:
Industrial Control Subsystem (DSS11/DRS11)	ISnn:
Laboratory Peripheral Accelerator (LPA11-K)	LAnn:
Laboratory Peripheral System (LPS11)	LSnn:
Line Printer (LA11/LS11/LP11/LV11)	LPnn:
Magtape (RH11/RH70/TM02/TM03)	MMnn:
(TM11/TMA11/TMB11)	MTnn:
Paper Tape Punch (PC11)	PPnn:
Paper Tape Reader (PC11/PR11)	PRnn:
Terminal (DL11/DH11/DJ11/DZ11)	TTnn:
Parallel Line Interface (DA11-B)	XBnn:
Asynchronous Line Interface (DL11-E)	XLnn:
Interprocessor Link (DMC)	XMnn:
Synchronous Line Interface (DP11)	XPnn:
(DQ11)	XQnn:
(DUP11)	XWnn:
(DU11)	XUnn:
Universal Digital Controller (UDC11)	UDnn:
Pseudo Devices	
Console Listing	CL:
Console Output	CO:
System Default Device	LB:
Network	NT:
Pseudo Input Terminal	TI:
User Default Device	SY:
Null Device	
Null Device	NL:

2.3.1 Logical Unit Numbers (LUN)

A program performs I/O on logical units identified by numbers, called Logical Unit Numbers (LUNs), rather than on specific device units. Either the programmer or an operator assigns each LUN to a specific device before the program actively uses the LUN. The LUN assignment occurs at one of the three points:

- At task build,
- From within the task at run time via an Executive directive, or
- Via an MCR command (Reassign) after the task has been installed and while it is dormant.

Refer to the RSX-11M I/O Drivers Reference Manual or the appropriate language User's Guide for more information about logical units.

2.3.2 Reassigning and Redirecting Devices

The MCR command Reassign (see Section 5.5) allows the operator to change a LUN assignment if for any reason an assigned device is not available to the task (the device might be temporarily inoperable, for instance). In practice, an operator usually circumvents this type of problem by another means: device redirection.

Rather than change one task's LUN assignment, the operator can use the Redirect command (see Section 5.5) to divert I/O from one device unit to another. LUN reassignment affects only one task, whereas redirection affects any task performing I/O on a redirected device.

Example

A programmer has assigned LUN 1 within a task to TT:. While the task is running, TT: becomes inoperable, making it impossible for the task to continue. The operator then issues a Redirect command to transfer all I/O performed on TT: to TT1: as follows:

```
>RED TT1:=TT:
```

The redirection remains in effect until the operator issues another Redirect command. If TT1: were also to fail, the redirected I/O could be diverted to another appropriate device, and so on. (Circular redirection, however, is not allowed.)

Note that redirection does not alter a task's LUN assignments.

2.3.3 Pseudo Devices

The previous section described the use of device redirection in exceptional conditions, when one or more devices are not operating properly. Redirection has a more routine function in relation to pseudo devices.

A pseudo device is a device unit name that does not correspond to a real device until it has been redirected. Every task in the system, including a system task, needs to communicate with one or more of the following pseudo devices.

CO:	Console output device
CL:	Console listing device
LB:	System library device
NT:	Network communcations device
TI:	Terminal input device
SY:	User's system device

In a given task, LUN 1 may be assigned to CL:, the console listing device. When the task uses LUN 1, the data is sent to whatever device the operator has redirected CL:. The operator can redirect CL: to any appropriate device (a line printer or a terminal, for example) without affecting the task's execution.

A pseudo device assumes the privilege status of the physical device to which it is redirected. If TT: is a privileged terminal, CL: becomes privileged when it is redirected to that terminal. But if CL: is subsequently redirected to a nonprivileged terminal, it will also become nonprivileged.

The normal procedure is to redirect CO: to the main operator's terminal, and CL: to the line printer. The system automatically redirects TI:, the most commonly used pseudo device, depending on how a task has been activated:

- When an operator issues the MCR command Run to immediately activate a task, the TI: for that task is redirected to the issuing terminal.

- If the Executive directive RUN$ activates another task, that task's TI: defaults to the TI: of the task that issued the RUN$ directive.

- If the Executive enters a task into a clock queue for activation after a specified interval, the Task's TI: defaults to CO:. (Both the MCR command Run and the Executive directive RUN$ allow the user to activate a task at a specified interval after the command or directive has been issued.)

2.3.4 The Null Device

Program testing often requires a data source and a data sink. A data source is a device used as a source of unlimited data; and a data sink is a device capable of receiving unlimited amounts of data. These devices are testing mechanisms only, and are not real devices. Testing programs use the null device to serve as both data source and sink. For example, while testing a program that normally generates large amounts of printout, the user can redirect or assign the appropriate LUN to NL:. The system then discards the output directed to NL:. When a program reads from NL:, the null device returns the code for end of file (IE.EOF); when a program writes to NL:, the null device returns the code indicating success (IS.SUC).

2.3.5 Logical Devices

Logical device names provide another means by which tasks can maintain device independence. (This feature is a system generation option). A logical device name has the same syntax as a real device unit; it consists of a 2-character ASCII name (alphabetic) and an optional 1- or 2-digit octal unit number, followed by a colon (:). The 2-character name can either be equivalent to a standard RSX-11M device name (e.g. DK:), or it can consist of two letters picked at random (e.g. XY:). Before a task refers to a logical device rather than a real device unit, an operator must issue the MCR command Assign to associate the task's logical device name with a real device unit (see Section 5.5).

There are three types of logical device assignment: global, local, and login.

- Global apply to all tasks running in the system.
- Local apply only to tasks initiated from the terminal used to make the assignments. Local assignments override conflicting global and login assignments; and the same logical device name may be locally assigned to different devices by different terminals.
- Login apply to systems that support multiuser protection. When a user issues the MCR Hello command to log into the system, the system automatically establishes one or more login logical device assignments. Login assignments override global assignments, but local assignments made from the terminal override both login and global assignments.

  If a local assignment overrides an assignment established at login, the system reinstates the login assignment when the user deassigns the local logical device name.

  The login logical device assignments remain in effect until the user logs off or until a privileged terminal deassigns them.

The Executive sets up a logical device table that records all logical device assignments and notes whether each assignment is local, login or global. The Executive then searches this table every time a task or command refers to a device. A logical name found in the table receives precedence over a physical device-unit having the same identifier.

2.4 MULTIUSER PROTECTION FUNCTIONS

Multiuser protection, a system generation option, allows an RSX-11M installation to monitor and control individual users of the system.

2.4.1 Logging On and Off A Terminal

Each user must log onto a terminal before the system allows the user to issue further MCR or task commands, other than the MCR HELP commands. Depending on the user's UIC, the terminal logged onto becomes privileged or nonprivileged. Each user then logs off the terminal after completing a session. The process of logging on and off a terminal enables the system to keep accounting information about each user.

The following information is printed on the console terminal following each log on/off in the format

```
time  LOGIN USER  lastname   [g,m]   TTnn:
time  LOGOUT USER [g,m]      TTnn:
```

The MCR commands that log a user on and off a terminal are Hello and Bye.

2.4.1.1 The Hello Command - The parameters to the Hello command are the user's UIC (or last name) and a password. The standard format for a UIC specifier is [g,m], where g and m are octal numbers from 1 to 377 that represent the user's group and member numbers, respectively. The Hello command allows several UIC formats; the UIC format specified affects the system's response to the command. A privileged user has a group number less than or equal to 10 octal. Each UIC has an associated 1- to 6-character alphanumeric string or password that guards against unauthorized access to the system. No one can successfully log onto a terminal without supplying the correct password for the specified UIC (or last name).

For example:

```
HELLO <CR>
ACCOUNT OR NAME:  [301,365] <CR>
PASSWORD:
```

Note that the Hello command prompts for its parameters when the user presses <CR> after each element of the command string. Alternatively, a user can type the entire command string on one line, as follows:

```
HELLO [301,365] /BOB
```

where BOB (preceded by a mandatory slash) is the password for the specified UIC. (The slash is required only when the password is on the same line as the UIC or last name.) When the user types a password in response to the prompt PASSWORD:, the system does not display the typed characters.

The user can also type the command in a mixed format, supplying one element of the string on one line and prompting for the other. The following example ensures that the system suppresses the user's password:

```
HELLO AMBER <CR>
PASSWORD:
```

Note that this example of the Hello command specifies the user's last name, from which the system determines the user's UIC.

After a user has successfully logged onto a terminal, the system automatically assigns the logical name SY: to the user's system disk (a login logical device assignment) and then searches the corresponding UFD for a file LOGIN.CMD. If found, the file is sent to the indirect processor (see Chapter 6). This is especially useful when the operator issues the same commands after each log on. For example, if the user created LOGIN.CMD file contained

```
SET /LOWER=TI:
SET /BUF=TI:80.
SET /CRT=TI:
```

the following would be displayed each time the user logged on:

```
>HEL AMBER
PASSWORD:

          RSX-11M BL22  MULTI-USER SYSTEM

GOOD AFTERNOON
19-SEP-77 14:06 LOGGED ON TERMINAL TT20:


>@LOGIN.CMD
>SET /LOWER=TI:
>SET /BUF=TI:80.
>SET /CRT=TI:
>@ <EOF>
```

If the user's group number is less than 11, the terminal becomes privileged; if the group number is 11 or greater, the terminal assumes nonprivileged status (see Section 2.1.2.1).

2.4.1.2 **The Bye Command** - A user who has completed a session at the terminal issues the Bye command to log off. This command has no parameters; it simply instructs the system to terminate the session. The system deletes all local and login logical device assignments, dismounts all mounted private devices (see Section 2.4.2 below), aborts all active nonprivileged tasks requested from the terminal, displays a termination message, and prevents further use of the terminal until another user logs on.

Example:

```
>BYE
>
HAVE A GOOD AFTERNOON
15-AUG-77 2:15 TT5:  LOGGED OFF
```

2.4.2 Public and Private Devices

Systems that support multiuser protection impose special characteristics on devices. Devices are either public, private, or unowned. These characteristics are defined as follows:

- A public device is a device that anyone logged into the system can use. A privileged user can make a device public by issuing an MCR Set command option.
- A private device is a device that a user has allocated (by means of the MCR command Allocate). Nonprivileged users must allocate nonpublic devices before accessing them for use within the Files-11 file system (see Section 3.1).
- An unowned device is a device that is neither public nor private. Only privileged users can issue the following MCR commands for unowned devices without previously allocating them: Mount, Dismount, Initialize Volume, and User File Directory.

A privileged user can designate any device in the system as a public device. The devices most commonly made public are the line printer and the system disk. See Section 5.5 for a description of the Set command /PUB option that makes a device public.

The MCR command Allocate makes a specified unowned device the private property of the user issuing the command. Device allocation prevents any other nonprivileged user from accessing the device. (Privileged users can override a nonprivileged allocation.) When a user accesses a device without allocating it, other users can then access and/or allocate that device.

A nonprivileged user can choose to make his or her own private device public by issuing the SET command /PUB option. The owner can also change the device's status back to private. However, when the owner logs off, the system automatically eliminates the public status of any associated private devices if the owner has not done so already.

The Deallocate command (see Section 5.5) changes a private device back to an unowned device, thereby allowing other users to gain access to it. Privileged users can deallocate any private device no matter who owns it. Nonprivileged users can deallocate only their own private devices. When a user logs off, the system automatically dismounts and deallocates any private devices belonging to that user. To encourage efficient use of system resources, however, users should specifically deallocate devices as soon as the devices are no longer needed.

2.4.3 The Account File Maintenance Program (ACNT)

RSX-11M provides an Account File Maintenance Program (ACNT) for the creation and maintenance of a multiuser account file. Whenever a user tries to log onto a terminal, the system checks the Hello command parameters against the account file to determine whether the user should be allowed access to the system (see Section 2.4.1). The account file describes all the UIC's that have been authorized for use within the multiuser protection system. One UIC can have several corresponding users, each user having his or her own distinct password.

The program ACNT is an interactive program that allows a privileged user to:

- Create the account file,
- Add new accounts to the file,
- Examine individual account entries,
- Modify individual account entries,
- List all the account entries in the file,
- Delete an account from the file, and
- Sort the account file.

When activated, ACNT lists the options and requests the user to select one. The user then enters the first letter of the verb that identifies the desired option (C for Create, A for Add, and so on). The program responds by requesting further input or by displaying information according to the option selected.

Each account entry includes the following information:

- UIC, which serves as an account number,
- Password,
- User's system device,*
- First name,
- Last name,
- Date and time of the user's most recent login, and
- The number of times that the user has logged into the system.

A nonprivileged user can run the program ACNT to change the account entry description of his or her password, first name, and/or last name (see Section 2.4.3.7).

The legal characters for user names are A-Z, 0-9, "$", "'", ",", and "!". Lower-case characters are converted to upper-case equivalents. Any ASCII character greater than 40 octal is valid for a password.

2.4.3.1 **Creating the Account File** - A multiuser system cannot simultaneously have more than one account file. The initial file should be created immediately after system generation to allow normal use of the system. The Account File Maintenance Program should be run under a privileged UIC (uic [1,2] is used by convention) to prevent nonprivileged users from accessing the account file. The program's create function allocates a file called [0,0]RSX11.SYS and sets the file's protection to [RWED,RWED,,]. This protection allows file access to privileged users and privileged tasks only.

A privileged user may copy the account file (RSX11.SYS) using PIP. The new copy will receive the RSX-11M default file protection, which includes read access for the world. The system manager should, therefore, restore proper file protection to all copies of the account file to prevent unauthorized access to ACNT.

The following example shows how to create the account file:

```
>RUN $ACNT
 * ACCOUNT FILE MAINTENANCE PROGRAM *
OPTIONS ARE: A - ADD, C - CREATE FILE, D - DELETE, E - EXAMINE,
L - LIST, M - MODIFY, S - SORT,  CTRL/Z - EXIT
ENTER OPTION: C <CR>
ENTER MAXIMUM NUMBER OF ACCOUNTS: n. <CR>
```

The user enters a decimal number (n.) to specify the maximum number of account entries to be listed in the file. The system then creates the account file large enough to contain that number of entries. If the account file proves to be insufficient, additional account file space can be obtained by copying the file using PIP and specifying the /BL switch. Note that the protection precautions listed above apply in this case.

* When a user logs in successfully, the system assigns the login logical name SY: to this device.

2.4.3.2 **Adding Account Entries** - After the user has entered the maximum number of accounts, the program prompts

```
ENTER ACCOUNT OR <ESC> FOR OPTIONS ( N,N ):
```

In this way, the ACNT program allows the user to start adding accounts to the file immediately after its creation. To respond, the user enters either the octal group and member numbers of a UIC, separated by a comma (g,m), or the ESCape or ALTmode key to request a list of the available options. Each new account is entered so that the file is maintained in sorted order by UIC.

The following example starts up ACNT and then shows how to add account entries to the file.

```
>RUN $ACNT
 * ACCOUNT FILE MAINTENANCE PROGRAM *
OPTIONS ARE: A - ADD, C - CREATE FILE, D - DELETE, E - EXAMINE,
L - LIST, M - MODIFY, S - SORT,  CTRL/Z - EXIT
ENTER OPTION: A <CR>
ENTER ACCOUNT OR <ESC> FOR OPTIONS ( N,N ): 200,202 <CR>
PASSWORD ( <=6 CHARS. ): BARLEY <CR>
DEFAULT SYSTEM DEVICE ( DDU ): SY <CR>
FIRST NAME ( <=12 CHARS. ): CATHY <CR>
LAST NAME ( <=14 CHARS. ): GRAHAM <CR>
UFD SY00:[200,202]
>
ENTER ACCOUNT OR <ESC> FOR OPTIONS ( N,N ): 200,203 <CR>
      .
      .
      .
```

ACNT prompts for each required item of information after the user has entered the UIC group and member numbers. The prompts include the number of characters permitted in the password, the first name, and the last name. The default system device prompt requests a device name and unit number (optional if 0).

After receiving the last name, ACNT issues an MCR UFD command to create a directory for the account on its disk. ACNT then repeats the prompt:

```
ENTER ACCOUNT OR <ESC> FOR OPTIONS ( N,N ):
```

The user can either continue to add account entries, choose another option, or terminate the program.

2.4.3.3 **Examining Account Entries** - The Examine option to the ACNT program displays details about individual account entries. After ACNT displays the options and the prompt

```
ENTER OPTION:
```

the user enters E, and ACNT responds by displaying

```
ENTER ACCOUNT OR <ESC> FOR OPTIONS ( N,N ):
```

The user then types the UIC numbers of the account to be examined. For example:

```
OPTIONS ARE: A - ADD, C - CREATE FILE, D - DELETE, E - EXAMINE,
L - LIST, M - MODIFY, S - SORT,  CTRL/Z - EXIT
ENTER OPTION: E <CR>
ENTER ACCOUNT OR <ESC> FOR OPTIONS ( N,N ): 200,202 <CR>
[200,202]     BARLEY    CATHY          GRAHAM           LB00:
     8/1/77  19:44:46   00001
ENTER ACCOUNT OR <ESC> FOR OPTIONS ( N,N ):
```

ACNT displays two lines of information about each entry corresponding to the specified account. The first line shows, from left to right, the account UIC, password, first name, last name, and default system device. The second line consists of the date and time of the account's last login, followed by the number of times a user has logged into the system with this account since the account file was created. After displaying all the entries for the specified UIC, the program prompts for another account or <ESC> for a list of options.

If ACNT receives another account number, it displays entry details about the specified UIC; ACNT remains in Examine mode until the user types <ESC> to request a list of options.

2.4.3.4 **Modifying Account Entries** - The Modify option allows a privileged user to change the password, default system device, first name and/or last name of an account. (A nonprivileged user can run the ACNT program to change the password, first name, and/or last name of his or her own account; see Section 2.4.3.7.) For example:

```
OPTIONS ARE: A - ADD, C - CREATE FILE, D - DELETE, E - EXAMINE,
L - LIST, M - MODIFY, S - SORT,  CTRL/Z - EXIT
ENTER OPTION: M <CR>
ENTER ACCOUNT OR <ESC> FOR OPTIONS ( N,N ): 200,202 <CR>
PASSWORD = BARLEY ? [Y/N]: Y <CR>
[200,202]     BARLEY    CATHY          GRAHAM                LB00:
     8/1/77  19:44:46   00001
TYPE <ESC> TO LEAVE ENTRY UNCHANGED
PASSWORD ( <=6 CHARS. ): PEEK <CR>
DEFAULT SYSTEM DEVICE ( DDU ): DB1 <CR>
UFD DB01:[200,202]
>
FIRST NAME ( <=12 CHARS. ): <ESC>
LAST NAME ( <=14 CHARS. ): <ESC>
ENTER ACCOUNT OR <ESC> FOR OPTIONS ( N,N ):
```

After the user has specified the account to be modified, the program asks the user for verification by displaying the account's password. The user then types either Y (Yes) or N (No) to indicate that the correct account was or was not specified. If ACNT receives a negative reply, it searches the file for another entry with the same UIC. For example:

```
ENTER OPTION: M <CR>
ENTER ACCOUNT OR <ESC> FOR OPTIONS ( N,N ): 200,202 <CR>
PASSWORD = PACKY  ? [Y/N]: N <CR>
PASSWORD = BARLEY ? [Y/N]: Y <CR>
```

When ACNT receives a positive reply, it lists the entry details, requests <ESC> in response to an item not to be changed, and then prompts for each changeable account item. In the first Modify example above, the user changes the account's password to PEEK and the default system device to DB1. The program issues an MCR UFD command to create a directory for the account on the new system disk, and then prompts for the remaining items (first and last names).

An examination of the [200,202] account might now read:

```
[200,202]     PEEK      CATHY        GRAHAM        DB01:
     8/1/77  19:44:46    00001
```

The user can either continue in Modify mode by entering another account, or type <ESC> to obtain a list of options.

2.4.3.5 Listing All Account Entries

2.4.3.5 Listing All Account Entries - The List option causes the ACNT program to display on either the line printer or the user's terminal all the accounts described in the account file. For example:

```
OPTIONS ARE: A - ADD, C - CREATE FILE, D - DELETE, E - EXAMINE,
L - LIST, M - MODIFY, S - SORT,  CTRL/Z - EXIT
ENTER OPTION: L <CR>
PRINT PASSWORDS? [Y/N]: N <CR>
ALL ACCOUNTS? [Y/N]: N <CR>
ENTER GROUP NUMBER (N): 7 <CR>
ENTER T - TERMINAL OR L - LINEPRINTER: T <CR>
[007,001]                     BOB        AMBER            SY00:
   10/12/77     9:52:21       01995
[007,002]                     TOM        MCBRIDE          LB00:
   10/11/77    11:54: 6       01391
[007,003]3                    CHUCK      DUFF             LB00:
    9/27/77    13: 8:51       00195
[007,004]                     CLARK      STOT             SY00:
   10/12/77     9:42: 3       00899
[007,005]                     BENN       STILL            SY00:
   10/12/77     9:46:44       01469
```

In this example, ACNT lists the entries of a specific group in the file (without displaying the passwords) at the user's terminal. The program then displays all its options and requests a choice from the user.

2.4.3.6 Sorting the Account File

2.4.3.6 Sorting the Account File - The Sort option sorts the Account File in ascending order by UIC. For example:

```
OPTIONS ARE:  A - ADD, C - CREATE FILE, D - DELETE, E -  EXAMINE,
L - LIST, M - MODIFY, S - SORT, CTRL/Z - EXIT
ENTER OPTION:  S <CR>
```

When the sort is complete,

```
OPTIONS ARE:  A - ADD, C - CREATE FILE, D - DELETE, E -  EXAMINE,
L - LIST, M - MODIFY, S - SORT, CTRL/Z - EXIT
ENTER OPTION:
```

is displayed and ACNT prompts for another option.

When updating an RSX-11M account file version prior to 3.1, the file is automatically sorted by UIC upon the first privileged access. The file protection will also be updated at this time. The message

```
SORTING BY UIC
```

will be displayed on the terminal to remind the user not to disturb the sort. The sort may take several minutes, depending on the size of the account file.

2.4.3.7 **Deleting Account Entries** - The Delete option enables the user to eliminate individual accounts from the file. For example:

```
OPTIONS ARE: A - ADD, C - CREATE FILE, D - DELETE, E - EXAMINE,
L - LIST, M - MODIFY, S - SORT,  CTRL/Z - EXIT
ENTER OPTION: D <CR>
ENTER ACCOUNT OR <ESC> FOR OPTIONS ( N,N ): 200,202 <CR>
PASSWORD = PACKY ? [Y/N]: N <CR>
PASSWORD = PEEK  ? [Y/N]: Y <CR>
DELETE UFD AND FILES? [Y/N]: Y <CR>
PIP DB01:[200,202]*.*;*/DE
>PIP DB01:[0,0]200202.DIR;*/DE
>
ENTER ACCOUNT OR <ESC> FOR OPTIONS ( N,N ):
```

After the user has specified the account to be deleted, ACNT displays the account's password and requests a Y (yes) or N (no) reply for verification. A negative reply causes ACNT to search for another entry with the same UIC. If the reply is Y for Yes, the program asks if the user wants to delete the account's UFD and other files. Another positive reply produces Peripheral Interchange Program (PIP) commands to delete these files. If the user enters N in response to the deletion query, the account's files and UFD are left intact. The user can then remain in Delete mode by entering another account or type <ESC> for the option list.

2.4.3.8 **Nonprivileged Account Modifications** - A nonprivileged user can run the ACNT program to change his or her password, first name, and/or last name; all other ACNT options are privileged. In the following example, the user changes only the account's password:

```
>RUN ACNT
   ACCOUNT FILE MAINTENANCE PROGRAM
>
PASSWORD ( <=6 CHARS. ): EMILY
[325,352]     EMILY        KATE              CHARLES              SY00:
TYPE <ESC> TO LEAVE ENTRY UNCHANGED
PASSWORD ( <=6 CHARS. ): JAMES <CR>
FIRST NAME ( <=12 CHARS. ): <ESC>
LAST NAME ( <=14 CHARS. ): <ESC>
OPERATION COMPLETE
```

When a nonprivileged user runs the ACNT program, it identifies itself and immediately prompts for the password of the user currently logged onto the terminal. It then searches the account file for the entry that matches the UIC and password. When ACNT locates the correct entry, it displays the account's UIC, password, first and last name, and default system device. The next line says to type <ESC> to leave an entry unchanged. ACNT then prompts on three consecutive lines for the entry items that a nonprivileged user can change. An item is changed whenever the user types characters other than <ESC> in response to the prompt. The prompts themselves show the maximum character length allowed for each item.

ACNT displays OPERATION COMPLETE and automatically exits after the user terminates the LAST NAME prompt line. The program also exits if the user enters the wrong password in response to the initial query.

To enable this nonprivileged option, the ACNT task must already be installed in the system. A nonprivileged user cannot request ACNT by the install-run-remove option of the Run command because ACNT is a privileged task.

2.4.3.9 **ACNT Error Messages** - The ACNT program returns the following error messages:

ACNT -- ACCOUNT ALREADY EXISTS

The user attempted to add an account with a UIC and password that are already contained in the account file.

ACNT -- ACCOUNT DOES NOT EXIST

The user attempted to delete, examine, or modify an account that does not exist.

ACNT -- ACCOUNT FILE ERROR - x.

The program detected an I/O error while processing the account file. See the IAS/RSX-11 I/O Operations Reference Manual, Appendix I, for a definition of the error code.

ACNT -- ACCOUNT FILE FULL

The account file is full; the user cannot add further accounts. See the description of the Create command for details on how to enlarge the account file.

ACNT -- INVALID DEFAULT DEVICE NAME

The default system device specified is either not in the system or not a Files-11 device.

ACNT -- INVALID OPTION

In response to the prompt ENTER OPTION:, the user did not specify A, C, D, E, L, M, S, or CTRL/Z, which represent valid ACNT options.

ACNT -- SYNTAX OR COMMAND INPUT ERROR

The user typed an invalid character. Re-enter the correct data.

ACNT -- WORKFILE - DYN. MEM. EXHAUSTED

There is insufficient dynamic memory. Either install the task in a larger partition, or install it with a larger increment.

ACNT -- WORKFILE - VIRTUAL STORAGE EXCEEDED

The workfile exceeds virtual memory.

2.5 THE MCR DISPATCHER

A feature of the Monitor Console Routine (MCR) is the MCR Dispatcher, which manages all unsolicited input submitted from unattached terminals. MCR commands are serviced either by:

- An MCR overlay or
- An independent task.

By following the procedures outlined below, the Dispatcher activates the appropriate overlay or task to service each MCR command.

To the Dispatcher, the first three letters of the command name are the significant part of a command line (see section 2.2.1). The Dispatcher uses the name to determine the required overlay or task. First of all, the Dispatcher searches a table containing names of commands serviced by MCR overlays. If the table contains a matching command, the Dispatcher loads the corresponding overlay, which proceeds to process the command line.

If the table does not contain a match, the Dispatcher prefixes three dots (...) to the command name and then searches the System Task Directory (STD) for a task by that name (that is, ...tsk). MCR returns an error if the STD does not contain the specified task. If ...tsk is already active at the terminal, MCR displays a TASK ACTIVE message. If the STD does contain ...tsk, and it is not running at the issuing terminal, the Dispatcher responds conditionally as follows:

- If ...tsk is not active, the Dispatcher requests the Executive to run the task at the issuing terminal.
- If ...tsk is active at another terminal, the Dispatcher creates a temporary STD entry that points to the already active task's disk image. The Dispatcher names the temporary task tskTnn, where nn is the unit number of the terminal that issued the command, and requests the Executive to run the task. When tskTnn exits, the Executive automatically deletes its STD entry.

2.6 THE SHUTUP PROGRAM

SHUTUP is a system program that shuts down an RSX-11M system; it issues warning messages to all terminals connected or logged into the system, aborts all nonprivileged tasks, and dismounts all devices. It then halts the system. SHUTUP can be run from any privileged terminal or from TT0:.

To successfully shut down a system using SHUTUP, the following tasks must be installed:

1. AT., the MCR indirect command file processor
2. ACS, only if dynamic checkpoint space has been established
3. BYE, if multi-user protection support is included
4. DMO, if there are any mounted volumes
5. ERF, if error logging is active.

When activated, SHUTUP identifies itself and then prompts for the number of minutes before the system is to be shut down, the interval in minutes between message broadcasts, and the number of minutes to wait before logins are disabled. For example:

```
>RUN $SHUTUP

RSX11M SHUT DOWN PROGRAM

ENTER MINUTES TO WAIT BEFORE SHUTDOWN:  15
ENTER MINUTES BETWEEN MESSAGES:  3
ENTER MINUTES TO WAIT BEFORE DISABLING LOGINS:  5
```

After receiving this input (the numbers 15, 3, and 5), SHUTUP broadcasts the following message to all connected or logged-in (multiuser protection systems only) terminals:

```
PLEASE FINISH UP, 15 MINUTES BEFORE SHUTDOWN
```

Then 3 minutes later, the terminals receive another message:

```
PLEASE FINISH UP, 12 MINUTES BEFORE SHUTDOWN
```

SHUTUP continues to decrement the first number entered and to issue the warning messages at the specified interval until the waiting time has elapsed.

If the system supports multiuser protection, after the specified delay before disabling logins has expired (measured from the time SHUTUP is started), SHUTUP displays the following message on the invoking terminal:

```
ALL FURTHER LOGINS ARE DISABLED
```

When the delay before system shutdown has expired, SHUTUP logs off all logged-in terminals (multiuser system only), stops the error logger task if active, and invokes the indirect command file [1,2]SHUTUP.CMD (which may contain system-specific MCR commands to be executed at system shutdown). Finally, all checkpoint space is deallocated, mounted volumes are dismounted, and the system halts.

At this point, the operator may either boot another system, or resume the current system by depressing the Continue switch on the console. If it is intended to continue the current system operation, it is essential to Mount the volumes and Allocate checkpoint space.

Example:

```
RUN $SHUTUP

RSX11M SHUT DOWN PROGRAM

ENTER MINUTES TO WAIT BEFORE SHUTDOWN: 15
ENTER MINUTES BETWEEN MESSAGES: 3
ENTER MINUTES TO WAIT BEFORE DISABLING LOGINS: 5

PLEASE FINISH UP, 15 MINUTES BEFORE SHUTDOWN

PLEASE FINISH UP, 12 MINUTES BEFORE SHUTDOWN

ALL FURTHER LOGINS ARE DISABLED

PLEASE FINISH UP, 9 MINUTES BEFORE SHUTDOWN
```

CHAPTER 3

FILES AND VOLUMES

This chapter discusses basic concepts of the RSX-11M file system and highlights aspects of file handling that directly relate to MCR functions. In RSX-11M, a file is an owner-named area on a volume, where volumes are magnetic media such as disks, DECtapes, and magnetic tapes.

3.1 FILES-11

RSX-11M includes a software system called Files-11 to oversee the storage and handling of files. Three MCR functions prepare volumes for use within Files-11: Initialize Volume (INITVOL), Mount (MOU), and User File Directory (UFD).

Files-11 supports three types of specially formatted volumes: disks, DECtapes, and magnetic tapes. (Files-11 magnetic tapes conform to the American National Standard Magnetic Tape Labels and File Structure for Information Interchange X3.27-1969.) Before any of these volumes can be used on RSX-11M, an operator must properly format it by issuing the MCR command INITVOL (see Section 5.5). A volume that has not been formatted by the INITVOL command is a "foreign" volume. Files-11, RSX-11M's file system, cannot process foreign volumes.

A system utility called the File Exchange Program (FLX) allows RSX-11M users to access nonsupported, foreign volumes in DIGITAL's DOS or RT-11 format (cassettes, for example). FLX translates files on such volumes into Files-11 format. In some cases, DECtape may have a foreign format and therefore require the use of the FLX utility. See the RSX-11 Utilities Procedures Manual for a full description of FLX.

Before Files-11 can access a file, the volume that contains it must be known to the system. An operator makes a volume known to the system by issuing the MCR command Mount for the device on which the volume is loaded. (Note that user tasks can access unmounted foreign volumes to perform I/O independently of Files-11. This capability is often necessary for real-time applications. See the IAS/RSX-11 I/O Operations Reference Manual.)

3.1.1 File Ownership and Directories

When an RSX-11M user creates a file, the system places the file name in a User File Directory (UFD), and stores the user's current User Identification Code (UIC) in the file header to indicate the owner of the file. In most cases, the UFD corresponds to the owner UIC; but a file can be listed in a UFD that is not related to the owner code. It is also possible for a file to be listed simultaneously in more than one UFD.

A UFD is itself a file, which an operator must explicitly create by means of the MCR command UFD (see Section 5.5). A user specifies a UFD in the format of a UIC: [g,m] where g and m are octal numbers in the range of 1-377 that represent the user's group and member number respectively. The actual name of the UFD is a concatenation of the group and member numbers, terminated by .DIR. For example, the name of the directory that corresponds to the UIC [203,65] is 203065.DIR. (Note that leading zeroes are added to the UFD to make g and m 3-places each.) All UFDs are listed in each volume's Master File Directory (MFD), which corresponds to UIC [0,0] and is therefore named 000000.DIR. The directories (both UFDs and the MFD) list the names of files and contain pointers to each file's header. The file header contains information about the file's owner and the physical location of the file segments; the header occupies one block of the volume's index file ([0,0]INDEXF.SYS).

To delete a UFD, first delete all of the files contained in it (thereby freeing the space for future use) by typing:

```
PIP> [g,m] *.*;*/DE
```

After all files have been deleted, the UFD can be deleted by typing:

```
PIP> [0,0]gggmmm.DIR;*/DE
```

where gggmmm is the concatenation of the group and member number to be deleted.

Refer to the RSX-11M Utilities Procedure Manual for a detailed description of the Peripheral Interchange Program (PIP).

3.1.2 File Protection

Anyone who wants to access a file must know the UFD in which it is listed. This knowledge, however, is not sufficient to guarantee access. A user must also satisfy conditions specified in a protection mask associated with the file to be accessed.

Every file has an associated protection mask that describes the types of access allowed to four defined user groups. RSX-11M defines four types of access:

- Read
- Write
- Extend
- Delete

The four user groups are defined according to UIC:

- System - All system tasks, which are those running under a UIC group number less than or equal to 10(8).
- Owner - Tasks that run under the same UIC as the file's owner.
- Group - Any task that is running under the same UIC group number (the first octal number within the brackets) as the file's owner.
- World - Any task or user that does not fit in one of the above categories.

A file owner assigns access rights by one of the following means:

- The MCR command Initialize Volume (INITVOL). This command allows the user to specify default file protection for all files created on a volume when the volume is initialized (formatted for use within Files-11).
- The MCR command User File Directory (UFD). This command establishes access rights to a UFD file only. A user often allows read access to a UFD, but denies access to the files listed within it.
- The Peripheral Interchange Program (PIP). PIP has an option that enables the owner to alter access right to files. See the RSX-11 Utilities Procedures Manual.
- The MCR command Mount. This command allows the user, while mounting a volume, to specify the default file protection to be given to files created on the volume. This specification overrides that established by means of the Initialize Volume command.

To gain access to a file, a task must satisfy the protection mask of both the file to be accessed and the UFD in which the file is listed. For example, to write to an existing file, a user needs at least Read access to the UFD and Write access to the file. To create a new file, a user needs both Write and Extend access to the UFD.

3.2 FILE SPECIFIERS

RSX-11M has a standard format for referring to files in a command line. The format is a string consisting of one or more input and/or output file specifiers, as follows:

```
outfile1,...outfilen=infile1,...infilen
```

where outfile is an output file specifier and infile is an input file specifier. The dots (...) indicate that any number of specifiers may be included in the string, the number being determined by the receiving task. The command string (the string of specifiers plus the task call or command) cannot exceed 80 characters, the maximum line length. The MCR commands INI and MOU (see Section 5.5) are exceptions to this rule.

The outfile portion may not be required. When it is omitted, the equal sign (=) may also be omitted in some cases. The individual file specifiers (whether input or output) have the following format:

```
dev:[g,m]filename.type;version/switch1.../switchn
```

where

dev: is the physical or logical device unit on which the volume containing the file is mounted. The device unit is expressed as a 2-character alphabetic ASCII device name (see Section 2.3) and an optional 1- or 2-digit octal unit number, followed by a colon (DK0:, DT1: or LP:, for example).

[g,m]
: is the User Identification Code (UIC) that specifies the User File Directory (UFD) in which the file is listed. g and m are octal numbers from 0 to 377 that represent the owner's group and member number, respectively. The brackets are a mandatory part of the UIC.

filename
: is the name of the file, an alphanumeric string from 1 to 9 characters in length. A dot always separates the filename from the file type.

type
: is a 3-letter mnemonic that identifies the nature of the file's contents. The standard RSX-11M file types are listed in Section 3.2.2. For example, FTN indicates that the file contains a FORTRAN source program, and OBJ indicates that the contents are object (compiled) code. A semicolon (;) always separates the file type from the version number.

version
: is an octal number from 0 to 77777 that differentiates among various versions of a file. For example, when a file is created, the system assigns it a version number of 1. When a user subsequently opens the file for editing, the system retains the original file for backup and creates a new file. The new file has the same filename and type, but the version number is 2.

The numbers -1 and 0 have special significance: -1 implies the lowest existing version of a file, and 0 implies the highest existing version.

/switch
: consists of an ASCII switch name (usually 2-characters long) that identifies a switch option, optionally modified as described below. Individual tasks specify relevant switch options, which generally modify the task's function in some way or provide information required by the task to respond correctly to the command line.

The switch itself has one of three forms:

```
/SW      sets the switch action and

/-SW     negates the switch action
or
/NOSW
```

The switch name may be modified by any number of values; these values may be ASCII strings, octal numbers, or decimal numbers. The following rules apply:

- Numbers terminated by a decimal point (.) are explicitly decimal.

- Numbers preceded by a pound sign (#) are explicitly octal.

- Whether the default for a numeric value is octal or decimal depends on the task to which the switch is directed.

- Any numeric value may be preceded by a plus (+) or minus (-) sign, where plus is the default.
- A plus or minus sign, if present, must precede the pound sign (#) of an explicit octal number (for example, -#323).

The number of permissible values and the interpretation of a switch depend on the task to which the switch is directed.

The following are valid switch specifications:

```
/BL:2:13
/NU:#37
/-CO
/NOSP
```

3.2.1 User Identification Codes (UICs)

In a system that does not support multiuser protection, each terminal has one associated User Identification Code (UIC), determined by the last SET /UIC command issued for the terminal, or [200,200] if the terminal's UIC has not been altered since system generation. The system uses the current UIC of the issuing terminal whenever the user omits the UIC field of a file specifier. The system also checks this UIC against the protection mask of any file that the user wants to access; the check determines the user's access rights to the file (see Section 3.1.2).

In a multiuser protection system, a terminal has two associated UIC's: a protection UIC and a default UIC. The protection UIC is checked against a file's protection mask to determine the user's access rights to the file. The default UIC determines the UFD to be accessed whenever the user omits the UIC field in a file specifier. How the system sets these two UIC's depends on the privilege status of the user logged onto the terminal.

When a nonprivileged user logs onto a terminal, the terminal assumes the user's logon UIC as its protection UIC. The logon UIC also becomes the initial default UIC; but the nonprivileged user can subsequently issue a SET /UIC command to alter the default UIC. The protection UIC remains unchanged. When a privileged user logs onto a terminal, the terminal also assumes the logon UIC as both the protection and the default UIC. However, when the privileged user subsequently issues a SET /UIC command, the system alters both the default and the protection UIC.

After logging into a multiuser protection system, a nonprivileged user cannot alter the terminal's protection UIC; whereas a privileged user always alters both UIC's when it issues a SET /UIC command. Therefore, a nonprivileged user can have a default UIC that differs from the protection UIC; but the protection and default UIC's of a privileged user are always the same.

3.2.2 File Specifier Defaults

The user may omit any field in a command line file specification except the filename.* When one or more fields in a file specifier is omitted, the system assumes the following defaults:

Field Default

dev: The pseudo device unit SY:, usually the device on which the system volume is mounted.**

[g,m] The default UIC (see Section 3.2.1) specified for each terminal either:

- at system generation (see the RSX-11M System Generation Guide)
- by means of the MCR Set command (see Section 5.5), or
- by means of the MCR Hello command (see Section 2.4.1.1).

filename No default.* A specifier must include an explicit filename or an asterisk (wildcard) within the filename field (see Section 3.2.4).

type One of the standard 3-letter mnemonics listed in Section 3.2.3. The type defaulted to depends on the task to which the specifier is directed, and whether the specifier refers to an input or an output file. For example, the FORTRAN compiler defaults the type of an input file to FTN and the type of an output file to OBJ.

version For input files, the highest existing version number.

For output files, the highest existing version number plus 1.

Note that the version number must be included when PIP is used to delete a file. This requirement prevents the user from inadvertently deleting the highest version of a file. See the RSX-11 Utilities Procedure Manual.

/switch Switch defaults are set by individual tasks.

* The MCR Boot command is the only exception to this rule. See Section 5.5.

** The MCR Boot command and specification of $ in the MCR commands Install and Run default to device LB:. See Section 5.5.

3.2.3 Standard File Types

RSX-11M has a standard set of file types used by all DIGITAL-supplied software to reflect actual file contents. Although the user can assign arbitrary 3-letter file types, it is advisable to use the standard types described below.

Type	File Contents
BAS	A BASIC language source program
CBL	A COBOL language source program
CMD	MCR or task commands (an indirect command file)
COR	A SLP correction file
DAT	Data (as opposed to a program)
DIR	A directory (e.g., a User File Directory)
FTN	A FORTRAN language source program
LST	A listing
MAC	A MACRO-11 source program
MAP	A Task Builder memory allocation map
MLB	A macro library
OBJ	An object program (output from the MACRO-11 Assembler or a compiler)
ODL	A Task Builder overlay description
OLB	An object module library
SML	The system macro library
SYS	A bootable system image
TMP	A temporary file
TSK	A task image
TXT	A text file

The defaults used by DIGITAL-supplied software are described in the manuals that pertain to the individual tasks.

3.2.4 The Asterisk Convention (Wildcards)

The asterisk convention allows a user to specify more than one file in a single specifier by placing an asterisk (*) or "wildcard" in one or more fields of the specifier. The wildcard causes the system to ignore the contents of the wild field and to select all the files that satisfy the remaining explicitly described fields.

An asterisk can be placed in any part of the file specifier except the device-unit field, which must be explicitly supplied or defaulted to SY: (see Section 3.2.1).

The following example, using the Peripheral Interchange Program (PIP), illustrates the use of wildcards. The command line

```
     PIP>PROG.MAC;1,PROG.OBJ;1,PROG.TSK;1/DE
```

deletes the three individual files specified. Since the three files have the same filename and version number, but different types, the following command line deletes the same files:

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

(The switch /DE instructs PIP to delete the files.) The command also deletes any other files (on the default device) named PROG with a version number of 1. A user should check existing files before using wildcards (for example, by issuing the command line PIP> PROG.*;1/LI).

3.2.5 Examples Of Filename Command Strings

In the following examples, the three letters followed by > are standard prompts for the respective system programs.

- Assemble the MACRO-11 source file CRGPT.MAC and create the object output file CRGPT.OBJ. Both files are on DK0: under UIC [200,200].

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

- Delete the file SBG.OBJ;5. The output filename string is not applicable and therefore is omitted. The input device defaults to SY:.

  ```
  PIP>[200,200]SBG.OBJ;5/DE
  ```

- Task build the object file CRGPT.OBJ. The output will be named CRGPT.TSK because TSK is the default file type for the Task Builder's output file; the device and UIC are also defaulted.

  ```
  TKB>CRGPT=CRGPT.OBJ
  TKB>//
  ```

- Delete all files with the name CATHDAT, regardless of type or version. The system uses default values for the device and UIC.

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

CHAPTER 4

START-UP PROCEDURES

This chapter describes how to start up RSX-11M according to the bootstrap to be used.

4.1 BM792-YB BOOTSTRAP ROM

Follow this procedure to start up the BM792-YB bootstrap ROM:

1. Press HALT.
2. Set the console switches to 173100.
3. Press LOAD ADDRESS.
4. Enter the address of the load device in the console switches:

Device	Code
RK11 Disk	177406
RF11 Disk	177462
RP11 Disk	176716
TC11 DECtape	177344

5. Lift the HALT key.
6. Press START.

The system then displays on the console terminal the start-up sequence described in Section 4.8.

4.2 MR11-DB BOOTSTRAP ROM

Follow this procedure to start up the MR11-DB bootstrap ROM:

1. Press HALT.
2. Enter the address of the load device in the console switches:

Device	Code
RF11 Disk	173100
RK11 Disk	173110

TC11 DECtape	173120
TM11 Magtape	173136
RP11 Disk	173154

3. Press LOAD ADDRESS.
4. Lift the HALT key.
5. Press START.

The system then displays on the console terminal the start-up sequence described in Section 4.8.

4.3 BM873-YA BOOTSTRAP ROM

Follow this procedure to start up the BM873-YA bootstrap ROM:

1. Press HALT.
2. Enter the address of the load device in the console switches:

Device	Code
RF11 Disk	173000
RK11 Disk	173010
TC11 DECtape	173030
TM11 Magtape	173050
RP11 Disk	173100
RC11 Disk	173144
KL11/DL11	173210
TA11 Cassette	173230
PC11 Papertape	173312

3. Press LOAD ADDRESS.
4. Lift the HALT key.
5. Press START.

The system then displays on the console terminal the start-up sequence described in Section 4.8.

4.4 BM873-YB BOOTSTRAP ROM

To start up the BM873-YB bootstrap ROM, follow this procedure:

1. Press HALT.
2. Enter the address of the load device in the console switches:

Device	Code
RH11/RH70/RS03/RS04 Disk (unit 0)	173000
RH11/RH70/RS03/RS04 Disk (unit in switch register)	173002
RK11 Disk (unit 0)	173030
RK11 Disk (unit in switch register)	173032
TC11 DECtape (unit 0)	173070
TM11 Magtape (unit 0)	173110
RF11 Disk	173136
RH11/RH70/TM02/TU16 Magtape (unit 0)	173150
RH11/RH70 Device Combination (unit zero)	173230
RH11/RH70 Device Combination (unit in switch register)	173232
RH11/RH70/RP04/RP05/RP06 Disk (unit 0)	173320
RH11/RH70/RP04/RP05/RP06 Disk (unit in switch register.)	173322
RP11/RP02/RP03 Disk (unit 0)	173350
RP11/RP02/RP03 Disk (unit in switch register)	173352
KL11/DL11 Console Terminal Reader	173510
TA11 Cassette (unit zero)	173524
TA11 Cassette (unit in switch register)	173526
PC11 Papertape Reader	173620

3. Press LOAD ADDRESS.

   For units other than zero, enter unit number in switch register.

4. Lift the HALT key.

5. Press START.

The system then displays on the console terminal the start-up sequence described in Section 4.8.

4.5 M9301-YA OR M9301-YB BOOTSTRAP ROM

Follow this procedure to start up either the M9301-YA or the M9301-YB bootstrap ROM:

1. Press the BOOT switch.

2. In response to the $ prompt on the console terminal, enter one of the following 2-letter codes:

Code	Device	
DB	RP04 05/06 Disk	M9301-YB only
DK	RK11 Disk	
DP	RP11 (RP02/03) Disk	
DS	RS03/04 Disk	M9301-YB only
DT	TC11 DECtape	
DX	RX11 Floppy Disk	
CT	TA11 Cassette Tape	
MC	Mixed Massbus Device	M9301-YB only
MM	TM02/TU16/TU45 Magtape	M9301-YB only
MT	TM11/TMA11/TMB11/TU10 Magtape	
PR	PC11/PR11 Paper Tape Reader	
TT	KL11/DL11 Console Terminal Reader	

The two alphabetic characters may be followed by an optional single octal digit for bootstrapping from units other than zero, and terminated by a carriage return.

The system then displays on the console terminal the start-up sequence described in Section 4.8.

4.6 M9301-YC BOOTSTRAP (PDP-11/70 ONLY)

The M9301-YC bootstrap operates only on the PDP-11/70. Perform the following steps to use this bootstrap:

1. Press the HALT switch and set back to the ENABLE position.
2. Set the start address of 17765000 in the console switches.
3. Press LOAD ADDR.
4. Set the device unit number in switches 0 through 2.
5. Set the device code in switches 3 through 6. The device codes are:

Code	Device
1	TM11/TU10/TS03 Magnetic Tape
2	TC11/TU56 DECtape
3	RK11/RK05 DECpack Disk Cartridge
4	RP11/RP02/RP03 Disk Pack
6	RH70/TU16/TU45 Magnetic Tape

7	RH70/RP04/RP05/RP06 Disk Pack
10	RH70/RS04/RS03 Fixed Head Disk

6. Ensure that switches 7 through 21 are off (down).

7. Press START.

The system then displays on the console terminal the start-up sequence described in Section 4.8.

Before the M9301-YC bootstrap actually boots the system, it performs CPU tests, instruction and addressing tests, and memory and cache tests. If a hardware failure is detected, the diagnostic program halts. The lights contain the ROM address of the halt. If this occurs, call the DIGITAL Field Service Engineer.

It may, however, be possible to continue with the bootstrap operation if the lights contain the address 17773764, which indicates a cache failure. To continue in this case, press CONT. This is the ONLY case in which it is possible to continue bootstrapping after the diagnostic detects an error.

The M9301-YC bootstrap is capable of relocating its input to various banks of memory. However, it is not possible to do this with RSX-11M.

4.7 M9301-YF BOOTSTRAP ROM

The following instructions bootstrap the system by means of the M9301-YF console emulator. Refer to the M9301 Bootstrap/Terminator Maintenance Manual, for other ways to boot the system with the M9301-YF.

Follow this procedure at the operator's terminal:

1. Press the console BOOT switch.

   If diagnostics are enabled*, the M9301-YF performs primary CPU diagnostics. The terminal then displays the contents of R0, R4, SP, and R5, respectively; the contents appear as four 16-bit octal numbers. Note that pressing the BOOT switch at this time causes PDP-11 systems without console registers to copy the PC into R5 before the power-up sequence begins.

   The terminal then displays the prompt $ at the beginning of a new line.

2. In response to the prompt $, type the appropriate device code at the terminal, followed by carriage return:

* The M9301-YF micro switches must be set to 000 to perform diagnostics when using the console emulator to boot the system. If the switches are set to 002, the M9301-YF console emulator boots the system without performing diagnostics. Consult a DIGITAL Field Service engineer or see the M9301 Bootstrap/Terminator Maintenance Manual.

Code	Device
DBn	RP04, RP05, RP06 Disk
DKn	RK05 Disk
DMn	RK06 Disk
DSn	RS03 or RS04 Disk
DTn	TU56 DECtape
DXn	RX01 Floppy Disk
MMn	TU16 Magnetic Tape
MTn	TU10 Magnetic Tape
PR	PC11 Paper Tape Reader
TT	DL11 Console Terminal Reader

In the list of codes above, n represents the device unit number; the only permissible values are 0 through 7. The unit number can be omitted if it is 0.

If diagnostics are enabled, the M9301-YF performs secondary CPU and memory diagnostics before booting the system.

The system then displays on the console terminal the start-up sequence described in Section 4.8.

4.8 START-UP DISPLAY SEQUENCE

If the system being bootstrapped was built with a different device configuration than the current host, a series of display lines appears on the console in the form:

```
DEVICE ddnn:  NOT IN CONFIGURATION*
```

where:

dd = device name.
nn = unit number.

Then the system displays:

```
>RED ddnn:=SY0:RSX-11M <version-number> <base level> <memory size> [MAPPED]
>RED ddnn:=LB0:
>MOU ddnn:volnam
>@[1,2]STARTUP
```

where:

<version number> - the release version number (for example, Release 3.1 of RSX-11M is V3.1).

* Devices not in the configuration are declared offline and the display OFFLINE is appended to these devices as part of the output of the Devices command.

<base level> - The software development group at DIGITAL periodically brings together new versions of the software under development. Internal groups may use this software to ensure correct operation and to build new software. These development milestones are called base levels. For example, Release 3.1 of RSX-11M is base level BL22.

<memory size> - the memory size (in the form of nnnK) of the machine on which the system was bootstrapped.

[MAPPED] - MAPPED is displayed for systems built to use the memory management unit.

ddnn - Device name and unit number of the system-resident device.

volname - The name of the volume that contains the booted system.

[1,2]STARTUP - The file specifier for the indirect command file, which is invoked automatically at bootstrap time and which contains user commands for initializing the bootstrapped system.

The system is now ready for operation. The RED and MOU displays are internally generated Redirect and Mount commands. The Redirect command establishes the load device as the system devices SY: and LB:, and the Mount command mounts the load device as a Files-11 volume.

CHAPTER 5

MCR COMMANDS

5.1 COMMAND SUMMARY

The MCR commands specified in this chapter are listed below by category. This command summary provides a compact overview of the facilities available at an operator's console. Privileged and nonprivileged commands are distinguished by P and NP (not part of the syntax). Detailed descriptions of the MCR commands are presented in alphabetical order, according to command name, in Section 5.5.

Initialization Commands

ASN (P & NP) — Define or delete a logical device assignment. List current assignments on user's terminal.

ACS (P) — Allocate or discontinue the use of a checkpoint file on disk.

BOOT (P) — Bootstrap a new system into memory and transfer control to it.

DMOUNT (P & NP) — Dismount a volume by running down file system activity, deleting the Volume Control Block (VCB), and declaring the volume offline.

INITVOLUME (P & NP) — Initialize an RSX-11M Files-11-structured volume.

INSTALL (P) — Install a task in the system.

MOUNT (P & NP) — Create the volume control block (VCB) and declare volume online for access by the file system.

SET (P & NP) — Alter or display system-wide or terminal characteristics.

TIME (P & NP) — Enter time and/or date into the system; display the time and date on entering terminal.

UFD (P & NP) — Create a user file directory (UFD) file and enter its filename into the volume's master file directory (MFD) file.

MCR COMMANDS

Informational Commands

DEVICES (P & NP) Display on the entering terminal the list of peripheral devices recognized by the system.

LUNS (P &NP) Display on the entering terminal the list of LUN assignments for an indicated task.

PARTITIONS (P & NP) Display on the entering terminal the list of partition definitions.

TASKLIST (P & NP) Display on the entering terminal the System Task List.

Task Control Commands

ABORT (P & NP) Terminate execution of a running task.

ALTER (P) Alter the priority of a task.

CANCEL (P & NP) Cancel time-based initiation requests for a task (no effect on current execution).

CLQUEUE (P) Display information about tasks in the clock queue.

FIX (P) Fix a task in memory (task becomes memory-resident).

REASSIGN (P) Change LUN assignment.

REDIRECT (P) Redirect all I/O requests from one physical device to another.

REMOVE (P) Remove a task from the system.

RESUME (P & NP) Resume execution of a suspended task.

RUN (P & NP) Schedule a task's activation. The task may run immediately, after a delay, or in synchronization with the system clock. Periodic rescheduling is optional.

UNFIX (P) Unfix a task from memory.

System Maintenance Commands

ACT (P & NP) Display on the entering terminal names of active tasks.

ATL (P) Display on the entering terminal name and status information for active tasks.

BRK (P) Pass control to the Executive Debugging Tool (XDT). Valid only for systems that have generated the XDT facility.

BRO (P & NP) Broadcast a message to one or a set of terminals.

OPEN (P)	Display on the entering terminal the contents of a memory location for examination or modification.
SAVE (P)	Save the image of memory in the file from which the system was booted.
TAL (P & NP)	Display on the entering terminal names and status of all tasks in the system.

Multiuser Protection Commands

ALLOCATE (P & NP)	Allocate a device to a user (establishes the device as the user's private device).
BYE	Log off the system.
DEALLOCATE (P & NP)	Deallocate a user's private device.
HELLO	Log into the system.
HELP	Display the contents of the Help file.

5.2 MONITOR CONSOLE INTERFACE

The operator communicates with the RSX-11M system from a terminal.

MCR (Monitor Console Routine) is the interface between the terminal and the RSX-11M system. An operator communicates with MCR by typing input in response to the default prompt (>) at an unattached terminal, or by typing input in response to the explicit MCR prompt (MCR>) at any terminal. See Section 2.1.3 for a complete description of input prompts.

All examples in this manual show the default prompt.

5.2.1 Command Syntax

It is not necessary to type the entire command name when submitting a command. MCR requires only the first three letters of a command name, followed by the command parameters, if any exist. If parameters exist, they must be preceded by at least one blank or tab.

The following example shows how the Time command can be specified. Note that the square brackets in the example indicate that the "E" is optional; the square brackets are not part of the command name. The angle brackets denote that the enclosed values are not literally part of the command syntax. For example, <CR> indicates that a carriage return terminates the input line.

```
>TIM[E] <CR>
14:00:04 22-AUG-77
```

Thus, either

```
     >TIM <CR>

        or

     >TIME <CR>
```

is acceptable.

5.2.2 Keywords

Some commands use keywords that generally apply to a command argument. A keyword is similar in function to the switches described in Section 3.2. A keyword consists of a slash (/), followed by an ASCII identification, and optionally followed by an equal sign (=) and the value of the keyword, as follows:

```
     /Keyword=value
```

Keywords can be entered in any order. As an example of keyword usage, the Install command requires a filename argument specifying the task to be installed. Keywords can be appended to the filename. One such keyword, /TASK, can specify the name under which the task is to be installed. Thus,

```
     >INS JIM.TSK/TASK=SUPER
```

causes the task contained in the file named JIM.TSK to be installed with a name of SUPER. Keywords are command-specific and are defined with each command.

An example of a keyword without a value is the UNLOCK switch of the Mount command:

```
     >MOU DK0:/UNL
```

This command specifies that disk unit 0 is to be mounted unlocked.

5.2.3 Comments

MCR treats a line of text as a comment if the first character in the line is a semicolon (;). In addition, the exclamation point (!) may be used to delimit comments in a command. The first exclamation point starts the comment and the next exclamation point or end of line terminates the comment. All text between the two exclamation points is ignored. For example:

```
     >;  THIS LINE IS A COMMENT

     >TAS !THIS IS A COMMENT STRING!
```

Comments are especially useful to clarify commands in MCR indirect command files, which are described in Section 6.1.

5.2.4 Command References to Active Tasks

5.2.4.1 **Task-Naming Convention** - In a system that supports multiuser protection, more than one copy of a task can be active simultaneously. When a user requests a task that is already active, the MCR dispatcher automatically creates a name for a copy of the requested task: tskTnn where nn is the unit number of the requesting terminal (see Section 2.5). However, to refer to that task from the terminal that requested it, the user only needs to specify the 3-character task name (tsk).

For example, if a user requests PIP from TT2:, when PIP is already active at another terminal, the MCR Dispatcher runs a task called PIPT2. To abort PIPT2 from TT2:, the user enters the command

```
>ABO PIP
```

and MCR aborts the correct version of PIP. However, to abort TT2:'s version of PIP from another terminal, a privileged user must type

```
>ABO PIPT2
```

(The user at TT2: could also explicitly specify PIPT2, but the full name is not necessary in most cases.)

Note that the naming convention does not apply to tasks requested by the MCR Run command.

5.2.4.2 **Install-Run-Remove Tasks** - Whenever a user activates a task by means of the install-run-remove option of the Run command, the MCR Dispatcher names the task TTnn, where nn is the unit number of the terminal that requested the task. To refer to task TTnn in a command issued from terminal TTnn:, the user can omit the task name altogether.

For example, if a user issues the command >RUN $CALC from TT1:, MCR runs the task contained in a file called CALC.TSK listed in the system directory (for example, [1,54] in a mapped system), and names it TT1. To abort this task from TT1:, the user can issue the Abort command without specifying a parameter; that is,

```
>ABO <CR>
```

The command

```
>ABO TT1 <CR>
```

issued either from TT1: or any privileged terminal also causes MCR to abort the task originally referred to as $CALC in a Run command issued from terminal TT1:. (Only a privileged user, or a user at a privileged terminal, can abort a task requested from another terminal.)

5.3 COMMAND NAMES

Command names are at least three characters in length. MCR accepts the first three characters of a command and then searches for a blank, tab, carriage return, or ESCape character. (An exception is the Help command; the user must type all four characters of the command name to distinguish it from the Hello command.) Therefore, embedded blanks are not allowed in a command name. If a command is entered incorrectly, the system displays an error message at the entering terminal.

Syntactical descriptions of commands and messages described in this chapter observe the following notational conventions:

1. Lower case indicates a variable whose actual value is determined when the command is entered or the message is issued. For example, the value of taskname depends on the name of the task associated with the command or message.

2. Brackets [] enclose optional items. A syntactical element enclosed in brackets (for example, [dt]) is optional; it may or may not be included in the command. One common exception to this rule is the syntax for the specification of a UIC. The format for a UIC is [g,m], where the square brackets are required syntactical elements.

3. Unless explicitly qualified, all numeric values required in a command may be entered as decimal or octal. Decimal values are indicated by a trailing period; octal values are indicated by the absence of a trailing period. Thus,

   255.

   and

   377

   have the same value. (The value 255. is base 10; the value 377 is base 8.)

5.4 COMMAND DESCRIPTION FORMAT

The following subheadings are used in the detailed command descriptions given in Section 5.5. Subheadings that do not apply to a particular command are simply omitted from the command description.

COMMAND NAME

This subheading identifies the command name in capital letters. The minimal MCR command acronym is shown on the upper right or left margin. If the command can be executed only from a privileged terminal, the letter P appears in parentheses following the command name. The text following this subheading describes the command's function.

Format:

The command format is given, and all parameters are described.

Examples:

Example uses are shown.

Notes:

A list of special considerations that may assist the programmer in using the command properly appears under this subheading.

Command Related Error Messages:

Error messages specific to the command appear under this subheading. The command may also produce the common error messages listed in Chapter 7. These common error messages are not listed with the individual commands presented in this chapter.

5.5 MCR COMMAND DESCRIPTIONS

ABO

ABORT (P & NP)

The Abort command allows a console operator to terminate the execution of an indicated task. If the issuing terminal is not privileged, the operator can abort only a task requested from the issuing terminal. Any task, however, can be aborted from a privileged terminal.

Aborting a task causes the system to force an orderly termination of the specified task. To effect the termination, the system:

- Performs I/O rundown. I/O for all non-file-structured devices is cancelled. I/O for file-structured devices is allowed to complete and then the files are deaccessed. All attached devices are detached. I/O rundown may take a considerable amount of time for tasks connected to a network node.

- Requests the Task Termination Notification routine (TKTN) to display a message on the aborted task's TI: terminal. The display describes the reason for the termination and the task's registers at termination. However, if the task is checkpointed at the time TKTN is called to process the abort, TKTN does not display the task's registers. TKTN also does not display the registers when the aborted task's name begins with three dots (...) or ends in Tnn, where nn is a terminal unit number.

- Releases the task's partition if the task is not fixed.

- If the system supports the ALTER PRIORITY directive (a system generation option), alters the task's running priority to 247. If the task to be aborted is currently rolled out to disk and waiting for a return to memory, the change in running priority usually causes a quick return. A task must be memory-resident to be aborted.

If the command includes the optional keyword /PMD, it forces the system to produce a post-mortem dump of the aborted task. The system therefore produces the dump whether or not the task was built with the /PM Task Builder switch or installed with the /PMD=YES keyword.

Format:

```
ABO[RT] taskname[/PMD]
```

where:

taskname - The name of the task to be aborted.

/PMD - A keyword that requests a post-mortem dump of the aborted task.

Examples:

```
>ABORT SCAN
TASK "SCAN  " TERMINATED
ABORTED VIA DIRECTIVE OR MCR
```

```
     PC=076400
     PS=000030
     R0=000000
     R1=004230
     R2=074350
     R3=000020
     R4=000000
     R5=076000
     SP=076114

>ABO MAC
TASK "MACT3" TERMINATED
ABORTED VIA DIRECTIVE OR MCR
```

MCR aborted a task called MACT3, even though the command, issued from TT3:, specified the 3-letter task name only (see Section 5.2.4.1). The command >ABO MACT3, issued from any privileged terminal, would have the same effect.

```
>ABO <CR>
TASK "TT24" TERMINATED
ABORTED VIA DIRECTIVE OR MCR
```

MCR aborted the task, TT24, from TT24:; the taskname need not be specified to abort task TTn.

Command Error Messages:

On receiving an Abort command, MCR requests the Executive to abort the task. The Executive subsequently activates the Task Termination Notification routine (TKTN). TKTN then produces the termination display shown in the above example. Messages following an Abort command can therefore come either from the command itself or from TKTN.

Messages From Abort command:

ABO -- TASK MARKED FOR ABORT

A previous request has already been made to abort the specified task, but the abort has not yet occurred. The task marked for abort is probably not currently resident in memory.

ABO -- TASK NOT ACTIVE

The specified task is not currently active.

Messages from Task Termination Notification Routine (TKTN):

TKTN displays information about task aborts, whether caused by the MCR Abort command or some other force. The display has the format:

```
TASK "<taskname>" TERMINATED
<abort cause>
```

Following the displayed cause for the abort is a list of the task's registers at the time of the abort. The possible causes of the abort are described below.

Abort Cause Messages:

ABORTED BY DIRECTIVE OR MCR

Either MCR or an Executive directive issued by another task caused the task to be aborted.

ABORTED VIA MCR

MCR aborted the task and requested a post-mortem dump.

11/40 F.P. EXCEPTION

The task encountered a floating point exception while executing on a PDP11/40, and no SST routine was specified to process the trap.

AST ABORT. BAD STACK

An AST could not be effected because the AST parameters could not be pushed onto the task's stack.

CHECKPOINT FAILURE. READ ERROR.

The task could not be read back into memory after being checkpointed.

IOT EXECUTION

The task executed an IOT instruction, and no SST routine was specified to process the trap.

LOAD FAILURE. READ ERROR

The task could not be loaded because of a hardware error.

MEMORY PROTECT VIOLATION

The task encountered a memory-protect violation, and no SST routine was specified to process the trap.

NON RSX EMT EXECUTION

The task executed an EMT instruction with an argument other than 377(8), and no SST routine was specified to process the trap.

ODD ADDRESS OR OTHER TRAP FOUR

The task executed a word instruction with an odd address, or it referenced a nonexistent memory location, and no SST routine was specified to process the trap.

PARITY ERROR

A parity error occurred while the task was executing. The task was fixed in memory so that the memory could not be reused by another task.

RESERVED INST EXECUTION

The task executed an illegal instruction, and no SST routine was specified to process the trap.

SST ABORT. BAD STACK

An SST could not be effected because the SST parameters could not be pushed onto the task's stack, or a stack overflow was detected in an unmapped system, as indicated by a nonzero value in the header guard word.

TASK EXIT WITH OUTSTANDING IO

The task exited with one or more outstanding I/O requests. Tasks should terminate all I/O operations before exiting. The system does, however, clean up all outstanding I/O.

T BIT TRAP OR BPT EXECUTION

The task either set the T bit in the Processor Status Word or executed a Breakpoint Trap instruction, and no SST routine was specified to process the trap.

TRAP EXECUTION

The task executed a Trap instruction, and no SST routine was specified to process the trap.

ACS

ALLOCATE CHECKPOINT SPACE (P)

The Allocate Checkpoint Space (ACS) command allocates or discontinues a checkpoint file on disk for systems that support the dynamic allocation of checkpoint space. The disk containing the checkpoint file must be in Files-11 format. Only one checkpoint file per disk is allowed; but numerous disks can each contain a checkpoint file. When looking for file space to checkpoint a task, the system initially searches the first checkpoint file to be created. If space is not available in the primary file, it searches the second file to be created, and so on. The order in which checkpoint files were created determines the order in which the system scans the files when searching for available checkpoint space. In general, sufficient checkpoint space is twice the total amount of memory that all of the running tasks are using.

An ACS command that discontinues a checkpoint file may not be effected immediately. If the specified file contains one or more checkpointed tasks, the tasks must be returned to memory before the system can discontinue the file. However, once the request to discontinue the file has been made, the system does not use the file for any more checkpointed tasks.

Format:

```
ACS ddn:/BLKS=n
```

where:

ddn: - A device-unit that contains a mounted Files-11 disk. If the unit number is not specified, it defaults to zero.

/BLKS - A keyword that specifies the number (n) of blocks on the disk to be allocated for the checkpoint file. (A block is 512(10) bytes long.) For example, the command

```
>ACS DB:/BLKS=250.
```

allocates 250(10) blocks for a checkpoint file on DB:.

When n is 0, the system discontinues the use of the checkpoint file; the file is neither deleted nor truncated. For example, the command

```
>ACS DB:/BLKS=0
```

discontinues the checkpoint file on disk DB:.

Examples:

-
  ```
  >ACS DS:/BLKS=250.
  >ACS DB:/BLKS=1024.
  ```

  Allocate a primary and a secondary checkpoint file. The primary file on DS: (a fixed-head disk) is 250(10) blocks long. The secondary file on DB: is 1024(10) blocks long.

- `>ACS DS:/BLKS=0`

  Discontinue use of the checkpoint file on DS:.

Command Error Messages:

ACS -- CHECKPOINT FILE ALREADY IN USE ON DEVICE

A previous ACS command established a checkpoint file on the volume mounted on the specified device-unit. A volume can contain only one checkpoint file.

ACS -- CHECKPOINT FILE NOW INACTIVE

This message appears after an ACS command has been issued and the file does not contain any checkpointed tasks. The message indicates that use of the file was discontinued immediately after the command was issued. If the file were still in use, TKTN would issue a message when the file was finaly discontinued.

ACS -- DYNAMIC ALLOCATION OF CHECKPOINT SPACE NOT SUPPORTED

The ACS command was issued on a system that did not include support of dynamic allocation of checkpoint space at system generation.

ACS -- FILE I/O ERROR nnn.

The ACS command detected an error when it tried to allocate or open the checkpoint file. The code nnn. is an FCS error code that defines the cause of the error. See Appendix I of the IAS/RSX-11 I/O Operations Reference Manual. (If the code is -24, the message may indicate that the disk, usually a fixed-head disk, does not have enough contiguous space to allocate the file. The normal implication of the code -24 is device full.)

ACS -- FILE NOT IN USE

A checkpoint file was not in use on the device specified in the ACS command.

ACS -- NOT FILES-11 DEVICE

The command specified a device that did not contain a Files-11 disk. The specified device-unit must contain a mounted Files-11 disk.

*** ddnn: -- CHECKPOINT FILE NOW INACTIVE

The message, issued by TKTN, indicates that a request to discontinue use of a checkpoint file on device ddnn: has finally been satisfied. The need to return checkpointed tasks from the discontinued file back into memory caused the delay between the request and this message.

ACT

ACTIVE (P & NP)

The Active command causes the system to display at the issuing terminal the names of only terminal-initiated tasks.

When issued in conjunction with /ALL (ACT /ALL), the command causes the system to display at the issuing terminal the names of all the active tasks.

The Active command may also be issued in conjunction with /TERM (ACT /TERM=TTn:) to display on TI: the names of all the active tasks for a specific terminal.

Formats:

```
ACT[IVE] [/keyword]

Keywords:

     /ALL
     /TERM=TTn:
```

Examples:

```
>ACT
...MCR
...SYS
>ACT /ALL
. LDR.
RMDEMO
...MCR
...SYS
F11ACP
...PIP
>ACT /TERM=TT7:
...PIP
>
```

ALL

ALLOCATE (P & NP)

The Allocate command establishes a specified device as the user's private device. (This command applies only to systems that support multiuser protection.) Device allocation prevents other nonprivileged users from accessing a device. Only the device's owner or a privileged user can deallocate a private device.

A nonprivileged user must allocate a device before mounting it. A privileged user can mount a device without allocating it beforehand, but mounting alone does not prevent another user from accessing and/or allocating the device.

Public devices or other users' private devices cannot be allocated.

The system automatically deallocates a user's private devices when the user logs off (issues the Bye command).

Format:

```
ALL[OCATE] ddn:
```

where:

ddn: - The device-unit name and number of the device to be allocated.

Example:

```
>ALL  DK2:
```

Allocate disk DK2:.

```
>ALL DK:
```

Allocate disk DK0:.

Command Error Messages:

ALL -- DEVICE ALLOCATED TO OTHER USER

The specified device has already been allocated by another user.

ALL -- DEVICE ATTACHED

The specified device cannot be allocated because it is attached to a running task.

ALL -- PSEUDO DEVICE ERROR

The specified device is a pseudo device. Pseudo devices cannot be allocated.

ALL -- PUBLIC DEVICE

The command attempted to allocate a public device. Public devices cannot be allocated.

ALL -- USER LOGGED ON TERMINAL

The command attempted to allocate a terminal that has been logged-in by another user. Logged-in terminals cannot be allocated.

ALT

ALTER (P)

The Alter command allows the operator to change the static or running priority of an installed task.

Format:

```
     ALT[ER] taskname /keyword

     Keywords:

          /PRI=running and static priority

          /RPRI=running priority only

     where:

          taskname - The name of the task whose priority is to be
                     altered.

          priority - The new priority to be assigned to the specified
                     task, from 1 to 250 (decimal).

          /PRI     - Changes the task's static priority, and, if the
                     system supports the ALTER PRIORITY directive, the
                     task's running priority.

          /RPRI    - Changes the task's running priority only (only
                     valid if the system supports the ALTER PRIORITY
                     directive).
```

Examples:

```
     >ALT TEST /PRI=248.

          Alter the static and running priority of task TEST to
          248(10).

     >ALT CMFRUN /RPRI=70.

          Alter the running priority of task CMFRUN to 70 (10).
```

ASN

ASSIGN (P & NP)

The Assign command defines, deletes, or displays logical-device assignments. Logical-device assignments are a means of associating logical names with physical devices, pseudo devices, or other logical devices. When a user assigns a logical name to a pseudo or logical device, the system resolves the assignment to the associated physical device. There are three types of logical-device assignment: global, local, and login.

- Global assignments apply to all tasks running in the system. Only privileged terminals can issue ASN commands to define or delete global assignments.
- Local assignments apply only to tasks initiated from the terminal used to make the assignments. Any terminal can define its own set of local assignments.
- Login assignments occur only in systems that support multiuser protection. When a user issues a Hello command to log into the system, the system automatically assigns the logical name SY: to the user's system device, which is the device that contains the user's files. The login assignments apply only to tasks initiated from the terminal while the current user is logged in. Only from a privileged terminal can an operator define or delete login assignments.

Local assignments have precedence over both login and global assignments; and login assignments have precedence over global assignments. When a user deassigns a local logical-device name, the system defaults to any conflicting login assignments.

A logical-device name has the same syntax as a real device unit; it consists of a 2-character ASCII name (alphabetic) and an optional 1- or 2-digit octal unit number, followed by a colon(:). The 2-character name can either be equivalent to a standard RSX-11M device name (for example, DK:) or it can be two letters picked at random (for example, XY:).

When a user installs a task, or the system processes an ASSIGN LUN directive, the system scans the logical device table before the physical device table when searching for a specified device. As a result, a user can define logical device names that are identical to physical device names even though the logical and physical device names do not refer to the same physical device. The system resolves a specified device name to the first matching table entry found. If a logical device name is identical to a physical device name, the logical name has precedence.

Nonprivileged formats:

```
ASN ppnn:=llnn:
```

Assign the logical name llnn: to device ppnn:. The command establishes a local assignment for the issuing terminal.

```
ASN
```

Display at the issuing terminal all local and login logical-device assignments.

ASN =[llnn:]

Delete the local assignment of the specified logical-device name; or delete all local logical-device assignments for the issuing terminal if the command omits llnn:.

Privileged formats:

ASN ppnn:=llnn:/GBL

Define a global assignment that associates the logical name llnn: with device ppnn:.

ASN /GBL

Display all global, local, and login assignments for all terminals in the system.

ASN =[llnn:]/GBL

Delete the specified global logical-device assignment or all global assignments if the command omits llnn:.

ASN ppnn:=llnn:/LOGIN[/TERM=ttnn:]

Establish a login logical-device assignment for a specified terminal, or for the issuing terminal if the command omits /TERM=ttnn:.

ASN /TERM=ttnn:

Display at the entering terminal all local and login logical-device assignments for the specified terminal.

ASN =llnn:/LOGIN[/TERM=ttnn:]

Delete the login logical-device assignment indicated by the name llnn: for a specified terminal, or for the issuing terminal if the command omits /TERM=ttnn:.

ASN =/LOGIN[/TERM=ttnn:]

Delete all login logical-device assignments for the specified terminal, or for the issuing terminal if the command omits /TERM=ttnn:.

ASN =[llnn:]/TERM=ttnn:

Delete the local logical-device assignment indicated by the name llnn: for the specified terminal; or delete all local logical-device assignments for the specified terminal if the command omits llnn:.

where:

pp - Physical, logical, or pseudo device name

nn - Unit number

ll - Logical device name

tt - Terminal device name

Example:

```
>ASN    DP:=US1:       !Define US1:
>ASN    DT:=US2:       !Define US2:
>ASN    DK:=GB0:/GBL   !Define global logical device name GB0:
>ASN                   !List local and login assignments
US2:    DT0:    LOCAL TI - TT0:
US1:    DP0:    LOCAL TI - TT0:
SY:     DP0:    LOGIN TI - TT0:
>ASN    /GBL           !List all assignments
US2:    DT0:    LOCAL TI - TT0:
US1:    DP0:    LOCAL TI - TT0:
SY:     DP0:    LOGIN TI - TT0:
GB0:    DK0:    GLOBAL
>ASN    =US2:          !Eliminate local logical device name US2:
>ASN    =/GBL          !Eliminate all global assignments
>ASN    =US1:          !Eliminate local logical device name US1:
>ASN    DK:=XY:        !Assign XY: to DK:
>ASN    XY:=ZZ:        !Assign ZZ: to XY:
>ASN                   !Display assignments
ZZ0:    DK0:    LOCAL TI-TT0:
XY0:    DK0:    LOCAL TI-TT0:
```

Command Error Messages:

ASN -- DEVICE NOT TERMINAL

The /TERM keyword value in an ASN command did not specify a terminal.

ASN -- LOGICAL DEVICE NOT IN SYSTEM

The specified logical device name was not defined and therefore could not be found in the logical device assignment table for the terminal.

ATL

ACTIVE TASK LIST (P & NP)

This command displays on the entering terminal the name and status of all active tasks in the system or the status of a particular task (taskname). The display provides information useful for determining the exact status of each active task. The display contains the following information for each task:

- Task name,
- Task control block physical address (octal),
- Partition name,
- Partition control block physical address (octal),
- Partition base and limit physical addresses (octal),
- Task's running priority and default priority,
- Task status flags,
- TI terminal physical device-unit,
- I/O count (decimal),
- Task local event flags, and
- Task registers and Processor Status Word (memory-resident tasks only).

The displayed task status flags are from the Task Control Block (TCB). This command displays the name of the bit if the bit is set to one at the time of the display. Zero bits are not displayed. Flag names are three characters long and correspond to the last three characters of the mnemonic defining the status bits. (See the RSX-11M System Generation Manual for a description of the TCB.) For example, FXD represents the bit T2.FXD and indicates that the task is fixed in memory. Names prefixed by a minus (-) sign indicate that the bit represents the complement of the condition. For example, -CHK indicates that the task is not checkpointable.

If the task is not in memory (the OUT flag is displayed), the contents of the PC, the PS, and the registers are not displayed.

STATUS	TCB FLAG	DESCRIPTION
ABO	T2.ABO	Task is being aborted.
ACP	T3.ACP	Task is an Ancillary Control Processor.
AST	T2.AST	Task is processing an AST.
BFX	T2.BFX	Task is being fixed in memory.
CAF	T2.CAF	Checkpoint space allocation failure occurred.
CAL	T3.CAL	Checkpoint space is allocated in task image.
-CHK	TS.CHK	Task is not checkpointable.
CKD	T2.CKD	Task checkpointing is disabled.
CKP	TS.CKP	Task is checkpointed.
CKR	TS.CKR	Task checkpoint request pending.
DST	T2.DST	Task AST's are disabled.
-EXE	TS.EXE	Task is not in execution.
FXD	T2.FXD	Task is fixed in memory.
HLT	T2.HLT	Task is being terminated.
MCR	T3.MCR	Task was activated by MCR.
MSG	TS.MSG	Task aborted, waiting for TKTN message.
NRP	TS.NRP	Task is mapped to nonresident partition.
NSD	T3.NSD	Task cannot receive data (no send data allowed).
-PMD	T3.PMD	Suppress task Post-Mortem Dump on SST abort
OUT	TS.OUT	Task is out of memory.
PRV	T3.PRV	Task is privileged.
RDN	TS.RDN	Task I/O is being run down.
REM	T3.REM	Task is to be removed on exit.

STATUS	TCB FLAG	DESCRIPTION
ROV	T3.ROV	Task has resident overlays.
SLV	T3.SLV	Task is slave.
SPN	T2.SPN	Task is being suspended.
SPNA		Task was suspended prior to AST.
STP	TS.STP	Task stopped for terminal input.
STPA		Task stopped prior to AST.
TIO	T2.TIO	Task is waiting for terminal input.
WFR	T2.WFR	Task is in a "wait-for" state.
WFRA		Task was in a "wait-for" state before AST.

Format:

```
ATL [taskname]
```

Example:

```
>ATL
. LDR. 053626 LDR     053572 00000000-00000000 PRI - 248.  DPRI - 248.
   STATUS:  -CHK FXD STP PRV
   TI - CO0:  IOC - 0.  EFLG - 000001 000000  PS - 170000  PC - 042516
   REGS 0-6  000162 005600 177777 105312 064240 105260 053540
RMDEMO 054222  GEN     056600 00204000-00217600  PRI - 225.  DPRI - 225.
   STATUS:  WFR PRV CAL
   TI - TT0:  IOC - 0.  EFLG - 000001 000000  PS - 170010  PC - 125072
   REGS 0-6  123354 000061 000000 125032 001601 131530 120352
...MCR 105530  SYSPAR 107734 00110000-00120000  PRI - 160.  DPRI - 160.
   STATUS:  STP -PMD PRV NSD CAL
   TI - TT20:  IOC - 0.  EFLG - 000001 040000  PS - 170000  PC - 122276
   REGS 0-6  000000 120470 121766 120424 122314 000000 120362
...SYS 105260  GEN     064240
```

(operator typed CTRL/O to suppress further output)

```
>ATL RMDEMO
RMDEMO 054222  GEN     056600 00204000-00217600  PRI - 225.  DPRI - 225.
   STATUS:  WFR PRV CAL
   TI - TT0:  IOC - 0.  EFLG - 000001 000000  PS - 170010  PC - 125072
   REGS 0-6  123354 000061 000000 125032 001601 131530 120352
```

BOOT (P)

The Boot command bootstraps a system that exists as a task-image file on a Files-11 volume. It provides a convenient means of terminating one system and initiating another, especially on minimum hardware configurations. For example, Boot can be used for terminating a real-time system and initiating a program development system.

The Boot command immediately terminates the system currently in operation.

Bootable systems are special task image files that have no task headers. Examples of such systems are RSX-11M, RSX-11S, and PRESRV.

Format:

```
BOO[T] [filespec]
```

where:

filespec - The standard RSX-11M file specifier of the form

```
dev:[g,m]filename.type;ver
```

that specifies the file from which a new system is to be loaded.

Defaults applied to the file specifier are:

dev:	- The system device LB:
[g,m]	- The UIC under which BOO is running
filename	- RSX11M
type	- SYS
version	- Latest version

Note that Boot is the only command that allows the file specifier to omit the filename field.

Restrictions:

- Virgin system Boot -- The boot device physical unit number (unit plug) must equal the logical unit number in the virgin system. Otherwise, the initial SAVe will not work.

  Under certain circumstances, it is not possible to boot a large, mapped, virgin Executive from an unmapped system. If the new Executive has loadable drivers above 27.5K words, the drivers will not be read into memory, thus causing problems when you try to run the system. This can happen because Boot moves the bootstrap code in an unmapped system to the top of memory less .5K. To an unmapped Boot, this location appears to be 28K. Boot then reads the virgin system image into memory starting at address zero up until the beginning of the bootstrap. On saved systems, Boot only reads the first block (512 bytes) of the saved system image and then passes control to the bootstrap code to read in the rest of the system.

On a mapped system, Boot reads in the entire mapped system because the KT11 is enabled, allowing Boot to recognize more than 28K words of memory.

- Saved system boot -- The physical unit number of the boot device does not have to be equal to the logical unit number. However, 5AV uses the corresponding physical unit number and searches the Unit Control Block (UCB) list for a match. When one is found, the UCB is used to determine the logical unit number. On multi-controller systems, it is possible that several devices have the same physical unit number (possible since the devices are on different controllers). Therefore, the search of the UCB may not find the UCB corresponding to the booted device.

  For example, if DK0 is physical unit 0 on the first RK11 controller and DK1: is physical unit 0 on a second RK11 controller, the system can be booted from DK0 and not from DK1:.

If the Executive Debugging Tool (XDT) has been generated into the system, XDT displays the following prompt when the system is booted for the first time:

```
XDT> BL22
XDT>
```

Typing the G key in response to the prompt returns control to RSX-11M, and causes the system to come up normally. After the system has been saved, XDT does not reappear unless a user sets a breakpoint or issues the BRK command. (See the *RSX-11M Guide to Writing an I/O Driver* for a description of XDT.)

Example:

```
>BOO TEST
```

Bootstrap from the system device the system contained in the file TEST.SYS.

Command Error Messages:

BOO -- DPB ERROR

A bad DPB was created by MCR. This error indicates that the system itself has faulted. If the error persists, submit a Software Performance Report (SPR) to DIGITAL.

BOO -- FILE NOT CONTIGUOUS

An attempt was made to load a system from a noncontiguous file. System images must be contiguous.

BOO -- INVALID LOAD DEVICE

Boot detected a device that was invalid as a system residence device.

BOO -- LABEL BLOCK READ ERROR

Boot could not read the label block of the system image.

BOO -- NO TRANSFER ADDRESS

Boot could not find a transfer address in a virgin system image (result of a task build).

BOO -- NOT SYSTEM IMAGE

Boot determined that the file is not a system image.

BRK

BREAKPOINT TO EXECUTIVE DEBUGGING TOOL (P)

The BRK command passes control to the Executive Debugging Tool (XDT) if it has been generated into the system. If XDT is not in the system, the command has no effect. See the RSX-11M Guide to Writing an I/O Driver for a description of XDT. Note that, if successful, the command halts all system activity; proceeding from a breakpoint normally restores the system to its previous state.

When XDT gains control, it prints the message

```
     BE:  nnnnnn
     XDT>
```

on TT0: to indicate that XDT now has control. All XDT commands are available for use in debugging the Executive or user-written drivers. To return control to MCR, the user enters the P command; MCR then prompts with >.

Format:

```
     BRK
```

Example:

```
     >BRK
     BE:  124560
     XDT>P
     >
```

BRO

BROADCAST (P & NP)

The Broadcast command causes the system to display a specified message at one or more terminals, as follows:

- A privileged user can display a message at all terminals connected to the system, or at all logged-in terminals (multiuser protection systems only).
- Both privileged and nonprivileged users can display a message at one specified terminal.

A user can also create an indirect file to contain one or more lines of input to the command. Each line specifies the target terminal and the message to be broadcast. (Privileged users can broadcast a message simultaneously to all terminals.) To use the command file, the user enters after the command name the file specification preceded by an at sign (@).

If the message cannot be broadcast within 10 seconds, the system displays the following message:

```
BRO -- TERMINAL IS BUSY -- ttn:
```

If the command specified multiple terminals (ALL: or LOG:), the system returns one error message for each busy terminal.

Formats:

- BRO[ADCAST] ttn:message
- BRO[ADCAST] @filespec
- BRO[ADCAST] ALL:message
- BRO[ADCAST] LOG:message

NOTE

Each format can be entered on two successive lines by pressing <CR> immediately after the command name. For example:

```
>BRO[ADCAST]  <CR>
BRO> ttn:message
```

where:

ttn: - Terminal to receive broadcast message.

message - Message to be broadcast, an ASCII character string that cannot exceed the length of one line (80 characters).

filespec - Specification of a file containing one or more message lines in the format:

ttn:message

or, for privileged users:

ALL:message or

LOG:message

ALL: - Privileged option. Indicates that specified message is to be sent to all terminals connected to the system.

LOG: - Privileged option. Indicates that the specified message is to be sent to all logged-in terminals. In a system that does not support multiuser protection, LOG: defaults to ALL:.

Examples:

-
  ```
  >BRO ALL:SYSTEM WILL SHUT DOWN 4PM TO 6PM
  ```

-
  ```
  >BRO <CR>
  BRO> TT3:PLEASE LOAD DISK1 ON DK3:
  ```

-
  ```
  >BRO @MESSAGE.CMD
  ```

Note:

- The Broadcast command uses the write-breakthrough feature of the terminal driver, a system generation option always generated with a multiuser protection system. If the system supports the feature, the broadcast message, under most circumstances, reaches all target terminals. If the system does not support the write-breakthrough feature, the write defaults to a write virtual block; the message cannot break through any type of I/O at the terminal.

Command Error Messages:

BRO -- COMMAND INPUT ERROR

The BROADCAST task did not receive the command line. (Usually an indirect file could not be found.) Retry command.

BRO -- COMMAND SYNTAX

The command line had an improper format.

BRO -- ILLEGAL DEVICE SPECIFIED

The specified device was not a terminal.

BRO -- PRIVILEGED COMMAND

A nonprivileged user entered a privileged option of the command, that is, attempted to broadcast one message to all connected or logged-in terminals.

BRO -- TERMINAL IS BUSY -- ttn:

The message could not be displayed within 10 seconds at the terminal specified by ttn:.

BYE

BYE

The Bye command logs the user off a multiuser protection system. The system writes a log off message on CO:, displays a terminating message on the initiating terminal, aborts any active nonprivileged tasks belonging to the user, dismounts any private volumes, and deallocates the user's private devices (if any).

Format:

```
     BYE
```

Example:

```
     >BYE
     >
     BYE -- DEVICE DK0:  BEING DISMOUNTED
     HAVE A GOOD MORNING
     25-AUG-77 10:04 TT15:  LOGGED OFF
```

When the user issued the BYE command, the system automatically deallocated and dismounted the disk DK0:, which the user had allocated and mounted. The system would also automatically abort any active nonprivileged tasks initiated by the user.

Depending on the time of day, the system displays HAVE A GOOD MORNING, HAVE A GOOD AFTERNOON, or HAVE A GOOD EVENING.

CAN

CANCEL (P & NP)

The Cancel command allows the operator to cancel time-based initiation requests for a task. These requests result from a RUN$ directive or any of the time-synchronized variations of the MCR Run command.

A Cancel command directed to a task not initiated from the entering terminal can only be issued from a privileged terminal.

Cancel does not affect the execution of the task while it is currently active. Cancel removes only the time-based schedule requests still in the queue.

Format:

```
     CAN[CEL] taskname
```

Example:

```
     >CAN XKE
```

Cancel all periodic rescheduling and time-based initiation requests for task XKE.

CLQ

CLQUEUE (P)

The CLQUEUE command displays on the entering terminal information about tasks currently in the clock queue. The information consists of the task names, the next time each task is to be run, and each task's reschedule interval, if one was specified. The system places a task in the clock queue if the task was activated by a Run command or a RUN$ directive that specified a time-based option.

Format:

```
CLQ[UEUE]
```

Example:

```
>CLQ
TASK1 SCHEDULED AT 18-DEC-76 10:24:30:00
TASK2 SCHEDULED AT 18-DEC-76 10:30:00:00 RESCHEDULE INTERVAL 30 MINS
>
```

The first task in the queue will be run at 10:24:30:00 (hour: minute: second: clock tick); it has not been rescheduled. The second task in the queue will next be run at 10:30:00:00 and is scheduled to run every 30 minutes.

DEA

DEALLOCATE (P & NP)

The Deallocate command releases a private (allocated) device, thereby allowing other users to access it. (This command applies only to systems that support multiuser protection.) Nonprivileged users can only deallocate private devices that they themselves have allocated. A privileged user can deallocate any device in the system. The system automatically deallocates all private devices when its owner logs off (that is, issues the Bye command).

Format:

```
DEA[LLOCATE] [ddn:]
```

where:

dn: - The device unit name and number of the private device to be deallocated. If the command does not specify a device-unit, the system deallocates all private devices owned by the user logged onto the requesting terminal.

Example:

```
>DEALL DK:
```

Deallocate disk DK:. The deallocated device can now be accessed by other users.

DEV

DEVICES (P & NP)

The Devices command displays on the entering terminal the symbolic names of all devices or, optionally, all of a particular device type known to the system. The device names appear in one column; second and subsequent columns contain additional information (when necessary) about each device.

Format:

```
DEV[ICES] [dd:]
DEV[ICES] [/LOG]
```

where:

dd: - Is the device unit name to be displayed.

/LOG - displays all logged on terminals.

Example:

```
>DEV
DB0:    PUBLIC MOUNTED LOADED
DB1:    MARKED FOR DISMOUNT
DB2:    LOADED
DB3:    OFFLINE LOADED
DK0:    LOADED
DK1:    TT1:   - PRIVATE       MOUNTED LOADED
DK2:    LOADED
DS0:    PUBLIC MOUNTED LOADED
DS1:    OFFLINE LOADED
DT0:    LOADED
DT1:    LOADED
LP0:    PUBLIC LOADED
MM0:    LOADED
MM1:    LOADED
PP0:    UNLOADED
PR0:    UNLOADED
TT0:     [7,16]  - LOGGED ON
TT1:     [301,316]       - LOGGED ON
TT2:
TT3:     [22,10] - LOGGED ON
TT4:     [304,351]       - LOGGED ON
CO0:    TT0:
CL0:    LP0:
SY0:    DB0:
>
>DEV    DK:
DK0:    LOADED
DK1:    TT1:   - PRIVATE       MOUNTED LOADED
DK2:    LOADED
>DEV /LOG
TT0:     [7,22]
TT1:     [301,316]
TT2:     [301,336]
TT4:     [7,5]
TT5:     [7,55]
TT13:    [301,365]
TT20:    [31,20]
TT21:    [301,341]
>
```

The following notes describe the messages that can appear in the subsequent columns. More than one message can appear on the same line.

- MOUNTED indicates that the device is mounted.

- PUBLIC indicates that the device has been set public.

- MARKED FOR DISMOUNT indicates that a mountable device has been requested to be dismounted, but the dismount operation has not yet been completed.

- OFFLINE indicates that, although the system tables contain entries for this device, the host configuration does not contain the related device.

- [uic] LOGGED ON indicates that the user identified by [uic] (the protection UIC) has logged onto the terminal.

- LOADED indicates that the driver for the device is loadable and is currently loaded.

- UNLOADED indicates that the driver for the device is loadable and is currently not loaded.

- A device name in the second column is the device to which the corresponding device in the first column has been redirected.

- A terminal name in the second column followed by the text "- PRIVATE" indicates that the device named in the first column has been allocated to the user logged on to the terminal in the second column.

DMOUNT (P & NP)

The Dismount command declares that the volume or communications channel specified in the command is logically offline. After a dismount operation, the device cannot be accessed by the associated Ancillary Control Processor (ACP). For a Files-11 volume (including magnetic tapes that conform to ANSI standards), a request is placed in the file-system queue to delete the Volume Control Block (VCB), and the volume is marked for dismount so that no additional files can be accessed on the volume. The VCB is a main memory-resident control block that controls volume access. The command is completed when the VCB is actually deleted; VCB deletion does not occur until all accessed files on the volume have been deaccessed. When the deaccessing process is complete, the system issues a message indicating that the dismount operation is complete. For a network device, the communication VCB is immediately deleted, and all additional processing on the channel is terminated.

A nonprivileged user can only dismount personally allocated devices (multiuser protection systems only). Otherwise, a user must issue the command from a privileged terminal.

When the dismount has been completed, either by the Files-11 disk system, ANSI magnetic tape system, or the network ACP, the message

```
*** dev:  DISMOUNT COMPLETE
```

is issued to CO:. There may be a considerable delay between the issuance of the command and the printing of this message if a number of I/O requests are pending and/or files accessed on the volume.

Format:

```
DMO[UNT] ddn:[volume-label]
```

where:

ddn: - Device-unit that holds the volume to be dismounted.

volume-label - The Files-11 volume label may be up to twelve characters in length for disk and DECtape, and up to six characters for magnetic tape. It is used to verify that the proper volume is being dismounted. A volume label must always be specified for ANSI magnetic tape, but must not be specified for a communications channel. For disks and DECtapes, a label is optional.

Default: NULL (if null, the system omits a volume label check).

Examples:

```
>DMO DK1:RVOLUME
```

This command causes the VCB for DK1: to be deleted.

```
>DMO XU:
```

This command causes the network ACP to terminate communication over the channel connected to device XU:.

```
>DMO MM0:PAYROL
```

This command causes the magnetic tape called PAYROL to be rewound and the VCB for the drive MM0: to be deleted.

Command Error Messages:

DMO -- ALREADY MARKED FOR DISMOUNT

The device-unit had been requested to be dismounted and was in the process of waiting for all accesses to the volume to complete.

DMO -- CHECKPOINT FILE STILL ACTIVE

The command attempted to dismount a volume that contained an active checkpoint file. The volume cannot be dismounted until the checkpoint file has been discontinued. Issue an ACS command to discontinue the file and reissue the DMO command when the system issues a message to indicate that the checkpoint file is inactive.

DMO -- HOME BLOCK CHECKSUM ERROR

The checksum in the home block and the calculated checksum did not agree. This message is usually caused by an I/O error.

DMO -- NO VOLUME LIST

The command specified a magnetic tape drive for which a mounted volume list does not exist.

DMO -- NOT MOUNTABLE DEVICE

The specified device was not a mountable device and therefore could not be dismounted.

DMO -- NOT MOUNTED

The specified device was not mounted.

DMO -- WRONG VOLUME

The volume label and the label specified in the command did not match.

FIX

FIX-IN-MEMORY (P)

The Fix-in-Memory command allows the operator to force a task to be loaded and locked into its partition. Subsequent requests for running the task can be serviced more quickly because the task is memory-resident and does not have to be loaded from the disk before it can run. The system can fix a task in memory only when the partition in which it is to be fixed becomes available.

Fixed tasks remain physically in memory even after they exit. Therefore, they do not have to be reloaded when a request is made to run them. (Note that reexecuting fixed tasks is not always practical.) Only an Unfix or Remove command can free the occupied memory.

The following restrictions apply to the fixing of tasks:

- Checkpointable tasks cannot be fixed.
- Active tasks cannot be fixed.
- A task must be installed before it can be fixed.

Format:

```
FIX taskname
```

Example:

```
>FIX XKE
```

FIX task XKE in its partition.

Command Error Messages:

FIX -- PARTITION BUSY

The partition in which the task was to be fixed was occupied, so the task could not be fixed.

FIX -- TASK BEING FIXED

The subject task was already being fixed.

FIX -- TASK ALREADY FIXED

The specified task was already fixed in memory.

FIX -- TASK CHECKPOINTABLE

The subject task was checkpointable. A checkpointable task cannot be fixed in memory.

HEL

HELLO

The Hello command logs a user onto a terminal to gain access to a multiuser system. Before a user logs in, the system rejects all MCR commands except Hello and Help. Parameters to the Hello command are the user's last name or User Identification Code (UIC) and a password; the system validates these parameters according to entries in an accounting file. If the user is authorized and the password matches the name or UIC supplied, the system:

- Sets the terminal to privileged or nonprivileged status according to the user's UIC (A privileged user has a group number less than or equal to 10(8).),
- Assigns the login logical-device name SY0: to the user's system disk (the disk that contains the user's files),
- Establishes the user's login UIC, which is the initial default UIC,
- Records the current time and date and other information about the terminal session in the accounting file,
- Displays a system identification, the current date, and the time the user logged onto the terminal,
- Optionally displays the contents of the file LB:[1,2]LOGIN.TXT (The user can suppress part of the display by typing CTRL/O),
- If SY:[login uic]LOGIN.CMD is present, it is submitted to MCR.
- Displays the default prompt.

Formats:

In the following command formats, the braces indicate that the user enters one of the two parameters.

-
  ```
  >HEL[LO]  {uic       /password <CR>
            {username
  ```

-
  ```
  >HEL[LO]  <CR>
  ACCOUNT OR NAME:    {uic       <CR>
                      {username
  PASSWORD:   password  <CR>
  ```

-
  ```
  >HEL[LO]  {uic       <CR>
            {username
  PASSWORD:   password <CR>
  ```

where:

uic - The user's User Identification Code (UIC). The command allows the following four UIC representations:

```
g,m
[g,m]
g/m
[g/m]
```

To suppress the display of the file LB:[1,2]LOGIN.TXT, the operator should use the slash form (that is, g/m or [g/m])

The variables g and m are octal numbers from 1 to 377 that represent the user's group and member numbers, respectively.

username - The user's last name, which can be entered in place of a UIC.

password - A 6-character alphanumeric password. If the user types the password in response to the prompt PASSWORD:, the system does not display the typed characters.

If the HELLO command was successful, the system displays the following message:

```
RSX-11M BL22 MULTI-USER SYSTEM

greeting
dd-mm-yy hh:mm LOGGED ON TERMINAL ttnn:

[login-text]
```

where:

greeting - is one of the following greetings, depending on the time of day:

GOOD MORNING
GOOD AFTERNOON
GOOD EVENING

dd-mm-yy - The current date

hh:mm - The time the user logged in

login-text - The contents of the file LB:[1,2]LOGIN.TXT. If this file does not exist, the message FILE SENDER ERROR -26 is displayed.

Examples:

-
  ```
  >HEL
  ACCOUNT OR NAME:  AMBER
  PASSWORD:

          RSX-11M BL22   MULTI-USER SYSTEM

  GOOD MORNING
  25-AUG-77 10:03 LOGGED ON TERMINAL TT20:

  24-AUG-77

          *** NEW DISK CONFIGURATION ***
          DB0:    System and User disk

  >
  ```

  (User typed CTRL/O to stop output.)

-
  ```
  >HEL [301/365] /BOB

            RSX-11M BL22   MULTI-USER SYSTEM

  GOOD MORNING
  25-AUG-77 10:06 LOGGED ON TERMINAL TT20:
  ```

-
  ```
  >HEL 301/365
  PASSWORD:

            RSX-11M BL22   MULTI-USER SYSTEM

  GOOD MORNING
  26-AUG-77 09:45 LOGGED ON TERMINAL TT17:
  >
  ```

-
  ```
  >HEL 301,365 /BOB

            RSX-11M BL22   MULTI-USER SYSTEM

  GOOD MORNING
  26-AUG-77 09:45 LOGGED ON TERMINAL TT17:

  24-AUG-77

            *** NEW DISK CONFIGURATION ***

  >

  (User typed CTRL/O)
  ```

Command Error Messages:

HEL -- ACCOUNT FILE OPEN FAILURE

The account file was open for another user; or the disk containing the account file was not mounted. Retry command.

HEL -- COMMAND INPUT ERROR

A system directive or the RUN command, rather than the HELLO command, has initiated the HELLO task.

HEL -- INVALID ACCOUNT

The name or UIC specified in the command is not stored in the account file; or the password specified does not match the name or UIC given.

HEL -- LOGINS ARE DISABLED

The system was in the process of shutting down; or the command SET /NOLOGON has been issued. A user cannot log onto a terminal at these times.

HEL -- MESSAGE FILE ERROR nnn.

The system could not open the file LOGIN.TXT for a reason indicated by the FCS code nnn. See Appendix I of the IAS/RSX-11 I/O Operations Reference Manual for a definition of the FCS code.

HEL -- OTHER USER LOGGED ON

The issuing terminal was currently logged by another user. Only one user at a time can be logged onto a terminal.

HEL -- TERMINAL ALLOCATED TO OTHER USER

The issuing terminal has been allocated to another user. A user cannot log onto a terminal allocated to someone else.

HELP

HELP

The Help command displays the contents of the file [1,2]HELP.TXT at the issuing terminal. Help is the only MCR command that a user can issue without logging on. The display usually tells the user how to log onto the terminal; but each installation can change the contents of its HELP.TXT file to suit specific needs.

Note that all four letters are required to distinguish Help from the Hello command.

The general format for the HELP command is:

```
     HELP qualifier qualifier2 ...  qualifier9
```

Another form of the HELP command displays the text contained in the HELP.TXT file of the user.

Format:

```
     HELP %qualifierl qualifier2...qualifier9
```

The user must be logged on to request HELP %. HELP % displays the contents of [current UIC] HELP.TXT.

The form of the HELP.TXT file is:

```
     1  UPPER-CASE-QUAL
        Text displayed if HELP UPPER-CASE-QUAL is typed.

     2  UPPER-CASE-QUAL
        Text displayed  if  HELP  UPPER-CASE-QUAL  UPPER-CASE-QUAL  is
        typed.

     2  UPPER-CASE-QUAL1
        Text displayed if  HELP  UPPER-CASE-QUAL  UPPER-CASE-QUAL1  is
        typed.

     1  ANOTHER-UPPER-CASE-QUAL
        Text displayed if HELP ANOTHER-UPPER-CASE-QUAL is typed.
```

The numerals are level numbers and can be from one through nine inclusive. Level numbers must be the first character of their line and the qualifiers that follow must be upper case. Arbitrary strings of blanks or tabs are allowed between level numbers and qualifiers. If a qualifier begins with a dollar sign ($) or a slash (/), the $ or / is ignored.

HELP interprets its command line in a strictly nested fashion. For example, if HELP UPPER-CASE-QUAL was typed, the line 1 UPPER-CASE-QUAL is searched for in HELP.TXT. If such a line is found, the text following 1 UPPER-CASE QUAL is displayed until another level number is encountered (text to the line 2 UPPER-CASE-QUAL). If any additional qualifiers were specified, the line 2 UPPER-CASE-QUAL is searched following the 1 UPPER-CASE-QUAL line. The search continues until 2 UPPER-CASE-QUAL is found or until a lower level number than 2 is found. The correspondence of qualifier position in the HELP command with level numbers is continued until the last qualifier is found. The desired text is then displayed.

The following is an example of a HELP.TXT file. The example file contains information on using the HELP command and how to create a HELP.TXT file.

One of the general forms of the HELP command is:

```
HELP qualifierl qualifier2...qualifier9
```

This displays the system help file (LB:[1,2]HELP.TXT).

The system help file presently provides information to enable you to create your own HELP.TXT file; that is the qualifier CREATE. Also included in this example HELP.TXT file is information on the MCR command ACTIVE; that is qualifier ACTIVE.

The other general form of the HELP command is:

```
HELP % qualifierl qualifier2...qualifier9
```

This displays information from a local (private) help file (SY:[current uic]HELP.TXT). This syntax is not legal until you are logged in.

1 ACTIVE

The general form of the ACtive Tasks list (short form) command is:

```
ACT /keyword
```

Keyword may be null, TERM=TTn:, or ALL. If null, the names of the active tasks for TI: are displayed.

ALL and TERM are qualifiers for ACT

2 ALL

ACT /ALL displays on TI: the names of all the active tasks in the system.

2 TERM

ACT /TERM=TTn: displays on TI: the names of all the active tasks for terminal ttn (tt is TT or TI and n is the unit number).

1 CREATE

NOTE

When creating a HELP.TXT file, the first line should be blank since it is not displayed.

By convention, each line should begin with a blank or tab; the lines cannot begin with the digits 1 through 9 since these would be interpreted as level numbers. Also, depending upon the types of devices in your configuration, some practical restrictions should be imposed. For example, to accommodate VT05Bs, lines of text should not be more than 72 decimal characters long, and a block of text should not be more than 20 decimal lines. If there are level n+1 qualifiers (like 2

UPPER-CASE-QUAL above), they should be mentioned in the text so that they are known to exist. Similarly, level one qualifiers should be added to the text displayed when only HELP is typed.

If the system contains the above HELP.TXT file, the following is the display in response to HELP, HELP CREATE, HELP ACT ALL (Note that a qualifier may be specified by the fewest initial characters that unambiguously define it).

```
>HELP

     One of the general forms of the HELP command is:

          HELP qualifier1 qualifier2...qualifier9

     This displays the system help file (LB:[1,2]HELP.TXT).

     The system help file presently provides information to enable you
     to create your own HELP.TXT file; that is the qualifier CREATE.
     Also included in this example HELP.TXT file is information on the
     MCR command ACTIVE; that is qualifier ACTIVE.

     The other general form of the HELP command is:

          HELP % qualifier1 qualifier2...qualifier9

     This displays information from a local (private) help file
     (SY:[current uic]HELP.TXT). This syntax is not legal until you
     are logged in.

>HELP CREATE


                              NOTE

          When creating a HELP.TXT file, the first line should be
          blank since it is not displayed.


     By convention, each line should begin with a blank or tab; the
     lines cannot begin with the digits 1 through 9 since these would
     be interpreted as level numbers. Also, depending upon the types
     of devices in your configuration, some practical restrictions
     should be imposed. For example, to accommodate VT05Bs, lines of
     text should not be more than 72 decimal characters long, and a
     block of text should not be more than 20 decimal lines. If there
     are level n+1 qualifiers (like 2 UPPER-CASE-QUAL above), they
     should be mentioned in the text so that they are known to exist.
     Similarly, level one qualifiers should be added to the text
     displayed when only HELP is typed.

>HELP ACT ALL

     ACT /ALL displays on TI: the names of all the active tasks in
     the system.

>
```

Command Error Messages:

HEL -- AMBIGUOUS QUALIFIER

The qualifier does not specify a unique part of the HELP.TXT file. Further information must be supplied. Examine the command line that help displayed below the error message and specify the last qualifier displayed more completely.

HEL -- IMPROPER HELP FILE FORMAT

The HELP.TXT file is not in proper format and cannot be used. Either edit the file or re-enter it in proper format.

HEL -- MESSAGE FILE ERROR nnn

The HELP.TXT file could not be opened for the reason specified by the FCS code nnn.

HEL -- SYNTAX ERROR

An error in the command format occurred.

HEL -- UNKNOWN QUALIFIER

The HELP.TXT file does not contain the last qualifier printed in the display of the command line below the error message.

INI

INITVOLUME (P & NP)

The function of the Initialize Volume command is to produce a Files-11 volume (see Section 3.1). On disk and DECtape, the command initializes the volume (destroys all existing files), writes a dummy bootstrap and a home block, and builds the directory structures. On magnetic tape, the command writes a volume label according to the ANSI standard and a dummy file that destroys all existing files.

In multiuser protection systems, a nonprivileged user can initialize a volume only on allocated, private devices (that is, a device the user has allocated). Otherwise, an operator must issue the command from a privileged terminal.

Selecting appropriate values for the command parameters requires an in-depth knowledge of Files-11. Refer to the IAS/RSX-11 I/O Operations Reference Manual for details of the Files-11 disk and ANSI magnetic tape structure.

Format:

```
INI[TVOLUME] ddn:volume-label[/keyword(s)]
```

Note that a hyphen (-) can be used as the last character on a line to extend the INI command line when selected keywords cause the command to exceed 80 characters (or whatever buffer size has been specified for the entering terminal). Any number of continuation lines is allowed, but the total command line cannot exceed 512. characters.

Keywords:

```
/BAD=[option]

/CHA=[characteristics]*

/DENS=bit-density

/EXT=block-count

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

/INDX=index-file-position

/INF=initial-index-file-size

/LRU=directory-preaccess-count

/MXF=file-count

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

/UIC*=[group,member]

/WIN=retrieval-pointer-count
```

(/DENS is the only keyword valid for initializing a magnetic tape volume.)

* This keyword is included for RSX-11D compatibility. Thus, volumes built on RSX-11M can be mounted on an RSX-11D system.

where:

ddn: - Device-unit name and number of the volume to be initialized.

volume-label - The volume label may be up to twelve characters in length for disk and DECtape, and up to six characters in length for magnetic tapes. The label identifies the volume and must be specified.

BAD - /BAD=[option]. The keyword /BAD indicates that bad-block processing is to be included in the volume initialization. Consequently, bad blocks on the volume will be marked as in-use and cannot be allocated to files.

The brackets surrounding each option are required syntax. The options are:

[OVR] - Include the last track in the BADBLK.SYS file. This option assumes that the bad block file is located on the last good block before the last track. The option is only valid for devices that support last tracks.

[MAN] - Accept a bad block list specified from the terminal.

[AUTO] - Read the bad-block file on the volume created by the diagnostic routine (BAD).

[AUTO,MAN] - Read the bad-block file and, when done, accept blocks specified from terminal.

If the command specifies [MAN], the program prompts for bad blocks as follows:

```
INI>BAD=
```

Bad blocks may be entered in two formats.

nnnnn - A single block

nnnnn,mmmm - A contiguous series of mmmm blocks beginning at nnnn. A null line (carriage return) terminates bad block input.

On an RK06, RL01, or RM03 disk, the default is BAD=[AUTO]. INI obtains the bad block information from a factory-recorded bad block file, located in the last track of the disk.

CHA - /CHA=[characteristics]. The characteristics words are ASCII strings separated by commas. The square brackets are required syntax. For the Initvolume command, two characteristics are defined:

[ATCH] - ATTACH/DETACH (device can be used by one task or group exclusively); included for RSX-11D compatibility. Unused by RSX-11M.

Default: no Attach/Detach.

[DCF] - Device Control Functions are permitted; included for RSX-11D compatibility. Unused by RSX-11M.

Default: no DCF.

DENS - /DENS=bit-density. The /DENS keyword determines the bit density (Bits Per Inch or BPI) of a magnetic tape loaded on either a TU16 or TU45 tape drive only. The drive must be capable of writing in either 800 BPI or 1600 BPI. Acceptable keyword values are:

/DENS=800

/DENS=1600

Default: /DENS=800

EXT - /EXT=block-count. The /EXT keyword specifies the number of blocks by which a file can be extended when the file has exhausted its allotted space.

Default: /EXT=5

FPRO - /FPRO=[system,owner,group,world]. The /FPRO keyword specifies the default protection for all files created on the volume being initialized. Access codes consist of four 4-code groups in the Access Rights word as follows:

R = Read
W = Write
E = Extend
D = Delete

In each instance, the absence of the code means that the user is denied the access right. The square brackets are required syntax.

Default: /FPRO=[RWED,RWED,RWED,R].

NOTE

Protection code subparameters (system, owner, group, world) are positional. That is, the location of the word in the parameter string defines the user to whom the code applies. The order is:

system, owner, group, world

The order of appearance of the characters R, W, E, and D is fixed.

R, if desired, must be first.
W, if desired, is next.
E, if desired, is next.
D, if desired, is last.

Thus, RWE and RE are acceptable, whereas WR and DEWR are not.

INDX - /INDX=index-file-position. The /INDX keyword specifies the index file logical block number. This keyword can be used to force the index file, the Master File Directory (MFD), and the storage allocation file to a specific volume location, usually for minimizing access time. Four possibilities are available:

BEG - Place the index file at the beginning of the volume.

MID - Place the index file at the middle of the volume.

END - Place the index file at the end of the volume.

BLK:nnn - Place the index file at the specified block number.

Default: /INDX=MID

INF - /INF=initial-index-file-size. The /INF keyword specifies the number of file headers to allocate initially in the index file. The five system files INDEXF.SYS, BITMAP.SYS, BADBLK.SYS, CORIMG.SYS, and 000000.DIR, are not included in the value for /INF.

Default: /INF=16.

LRU - /LRU=directory-preaccess-count. The /LRU keyword specifies the number of 512-byte buffers to be maintained in memory. The buffers will contain only the most recently accessed directories.

Default: /LRU=3

MXF - /MXF=file-count. The /MXF keyword specifies the maximum number of files (file headers in the volume's index file) permitted on the volume. The value for /MXF includes the five system files and must be less than or equal to the number specified for the device in the table below:

	Maximum
DECtape	144
RK05 Disk	1200
RK06 Disk	6781
RF Disk	256
RL01 Disk	2500
RM03 Disk	2200
RS03 Disk	256
RS04 Disk	512
RP02 Disk	10000
RP03 Disk	20000
RP04/05 Disk	42949
RP06 Disk	65536

	Defaults
DECtape	72
RK05 Disk	600
RK06 Disk	3390
RF Disk	128
RL01 Disk	1250
RM03 Disk	1100
RS03 Disk	128
RS04 Disk	256
RP02 Disk	5000
RP03 Disk	10000
RP04/05 Disk	21474
RP06 Disk	32768

NOTE

A Files-11 volume requires five files -- INDEXF.SYS, BITMAP.SYS, BADBLK.SYS, CORIMG.SYS, and 000000.DIR -- to create the on-disk structure (See Appendix E of the IAS/RSX-11 I/O Operations Reference Manual). Thus, MXF value must be computed by the maximum number of user UFD's and files plus the five files required for the Files-11 structure.

PRO* - /PRO=[system,owner,group,world]. The /PRO keyword specifies volume access rights. Access codes consist of four 4-code groups in the access rights word, as follows:

R = Read
W = Write
E = Extend
D = Delete

* RSX-11M does not protect a volume at the UIC level. It does, however, prevent direct access to mounted volumes by nonprivileged tasks. This feature permits volume interchangeability with RSX-11D.

Please cut along this line.

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. Problems with software should be reported on a Software Performance Report (SPR) form. If you require a written reply and are eligible to receive one under SPR service, submit your comments on an 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

-- Fold Here --

--- Do Not Tear - Fold Here and Staple ---

Postage will be paid by:

digital

Software Documentation
146 Main Street ML 5-5/E39
Maynard, Massachusetts 01754

In each instance, the absence of the code means that the user is denied the access right. The square brackets are required syntax.

Default: [RWED,RWED,RWED,RWED].

NOTE

Protection code subparameters (system, owner, group, world) are positional. That is, the location of the word in the parameter string defines the user to whom the code applies. The order is:

system, owner, group, world

The order of appearance of the characters R, W, E, and D is fixed.

R, if desired, must be first.
W, if desired, is next.
E, if desired, is next.
D, if desired, is last.

Thus, RWE and RE are acceptable, whereas WR and DEWR are not.

UIC* - /UIC=[group,member]. Keyword for the UIC parameter specifying the owner of the volume. Legal group and member numbers range from 1 to 377(8). The square brackets are required syntax.

Default: /UIC=[1,1]

WIN - /WIN=retrieval-pointer-count. The /WIN keyword specifies the number of mapping pointers to be allocated for file windows. A file window consists of a number of mapping pointers; it is stored in memory when the file is opened. (See Appendix F in the IAS/RSX-11 I/O Operations Reference Manual for a description of a mapping pointer.)

Default: /WIN=7.

Example:

```
>INI DK1:ICTSVOL2/UIC=[2,5]/INDX=BEG/FPRO=[RWED,RWE,RW,R] <CR>
```

The Initvolume parameters in the example above are:

DK1:	- Device-unit
ICTSVOL2	- Volume-label
/UIC	- User identification code of the volume owner: group=2, member=5

* RSX-11M does not protect a volume at the UIC level. It does, however, prevent direct access to mounted volumes by nonprivileged tasks. This feature permits volume interchangeability with RSX-11D.

/INDX=BEG - Index file location, forced to beginning of volume

/FPRO - Default file protection, specified by group as:

system:	Read,Write,Extend,Delete
owner:	Read,Write,Extend
group:	Read,Write
world:	Read-only

Command Error Messages:

INI -- ALLOCATION FOR SYS FILE EXCEEDS VOLUME LIMIT

The system was unable to allocate a system file from the specified block because of intermediate bad blocks or end of volume.

INI -- BAD BLOCK FILE CORRUPT - DATA IGNORED

Although automatic bad block recognition was selected, the bad block data on the disk was not in the correct format, and was therefore ignored.

INI -- BAD BLOCK FILE FULL

The disk had more than 102 bad regions on it.

INI -- BAD BLOCK HEADER I/O ERROR

An error was detected in writing out the bad-block file header.

INI -- BLOCK(S) EXCEED VOLUME LIMIT

The specified block (or blocks) exceeded the physical size of the volume.

INI -- BOOT BLOCK WRITE ERROR

An error was detected in writing out the volume boot block.

INI -- CHECKING DDnn

Not an error message. An automatic bad-block specification was proceeding, using the bad-block file provided by the Bad Block Locator utility program or, on an RK06, RL01, or RM03, the factory-written file from the last track of the disk.

INI -- CHECKPOINT FILE HEADER I/O ERROR

An error was detected in writing out the checkpoint file header.

INI -- COMMAND TOO LONG

The command, including continuation lines, exceeded the maximum length of 512. characters.

INI -- DATA ERROR

The command specified a bad-block number or contiguous region that was too large.

INI -- DISK IS ALIGNMENT CARTRIDGE

The last track on an RK06, RL01, or RM03 identified the volume as an alignment cartridge, which cannot be initialized as a Files-11 volume. An alignment cartridge is specifically formatted for aligning disk read/write heads.

INI -- DUPLICATE BLOCK(S) FOUND

A block that had been defined as bad was being defined as bad a second time.

INI -- FAILED TO READ BAD BLOCK FILE

The command was unable to read the bad block information from the last track of an RK06, RL01, or RM03 disk.

INI -- HOME BLOCK ALLOCATE WRITE ERROR

In overwriting a bad-home-block area, a write error occurred.

INI -- HOME BLOCK WRITE ERROR

An error was detected in writing out the volume home block.

INI -- ILLEGAL KEYWORD VALUE

A value entered for a keyword exceeded its limits.

INI -- INDEX FILE BIT MAP I/O ERROR

An error was detected in writing out the index-file bit map.

INI -- INDEX FILE HEADER I/O ERROR

An error was detected in writing out the index-file header.

INI -- I/O ERROR SIZING DISK

The system encountered an I/O error while sizing an RF11 disk.

INI -- MAGTAPE LABEL MUST BE SPECIFIED

The command attempted to initialize a magnetic tape without specifying the required volume label.

INI -- MAGTAPE DEVICE ERROR

While positioning the magnetic tape, the system encountered an I/O error.

INI -- MAGTAPE WRITE ERROR

While writing to the magnetic tape, the system encountered an I/O error.

INI -- MFD FILE HEADER I/O ERROR

An error was detected in writing out the Master File Directory (MFD) file header.

INI -- MFD WRITE ERROR

An error was detected in writing out a block in the Master File Directory (MFD).

INI -- NO BAD BLOCK DATA FOUND

Although automatic bad-block specification was selected, no bad-block file was found on the volume.

INI -- NOT FILES-11 DEVICE

The system does not support Files-11 on the specified device.

INI -- NULL FILE HEADER I/O ERROR

An error was detected in writing out null-file headers to the index file.

INI -- STORAGE BITMAP FILE HEADER I/O ERROR

An error was detected in writing out the storage allocation file header.

INI -- UNIT DOES NOT SUPPORT 1600 BPI

The command specified a tape drive that does not support 1600 BPI.

INI -- VOLUME MOUNTED

An attempt was made to initialize a mounted volume. Mounted volumes can not be initialized.

INI --VOLUME NOT READY

The command specified a volume that was not ready (not up to speed).

INI -- VOLUME WRITE LOCKED

The command specified a volume that was write-locked and therefore could not be initialized as a Files-11 device.

INI -- WARNING BLOCK 0 IS BAD

Block 0 of the specified volume, the boot block, was bad. A bootable image can therefore not be placed on this volume.

INS

INSTALL (P)

The Install command makes a specified task known to the system, or loads a resident library into a common partition. An installed task is dormant until the Executive receives a request for it to run. The optional keywords either provide parameters that were not specified at task build, or they override previously specified parameters.

When a task is installed, the system creates a Task Control Block (TCB) for the task in a memory-resident table called the System Task Directory (STD). The task parameters contained in the TCB enable the system to run the task when requested to do so by an MCR command or a system directive. A task cannot be run until it has been installed. (See Section 1.3.1.) (Note that the RUN command has an option that automatically installs and removes a task before and after running it.)

A task's TCB contains the address of the load-device Unit Control Block (UCB) and the Logical Block Number (LBN) of the task image file. The Executive can therefore load the task image in one transfer when an MCR command or system directive requests an installed task. (Loading does require more than one transfer if the task is greater than 32K words.)

When a user installs a task, the system stores the task's file identification in the task header. When a system is saved, the save task places the file identification, rather than the file's LBN, in the TCB. Then when the system is rebooted, it reopens the task file and stores the new current LBN of the task in the TCB. If a task has been deleted, the system cannot open the task file when the system is rebooted. The system then automatically removes the task's TCB from the STD.

NOTE

> If an installed task's file image is deleted and other data is written into that area, the system will crash. A crash also occurs if the disk from which the task was installed is removed and replaced with a different disk. This is because the task's file identification in the task header will then contain random data.

A saved system does not retain the physical addresses of installed tasks. The Disk Save and Compress (DSC) utility can compress and copy a disk containing a saved system, thus moving the task files to different physical locations. But since the TCB entries contain task file identifications, rather than LBNs, after a system save, the system can function normally when it is rebooted.

Format:

```
     INS[TALL] [$]filespec[/keyword(s)]
```

Keywords:

/CKP=option

/INC=size

/PAR=pname

/PMD=option

/PRI=number

/SLV=option

/TASK=taskname

/UIC=[group,member]

where:

filespec - The file specifier in the form:

```
dev:[group,member]filename.type;version
```

The type defaults to TSK; standard defaults apply to the remaining fields.

If $ is specified in the command line, then dev: defaults to LB:, [group,member] defaults to the current system UIC.

CKP - /CKP=option. Checkpoint option. If CKP=YES (default for checkpointable tasks), checkpointing is allowed. If CKP=NO, checkpointing is disabled for the task.

The /CKP keyword overrides the /CP switch specified at task build.

On systems that support the dynamic allocation of checkpoint space, the task image file does not need to contain allocated checkpoint space. The presence of the /CKP=YES keyword instructs the system to allocate checkpoint space for the task when required from a checkpoint file.

INC - /INC=size. The number of additional words of address space to be allocated to a task that is to run in a system-controlled partition. The extension size is usually expressed in octal but may also be expressed in decimal by appending a period. This keyword overrides the EXTTSK or PAR parameter used at task-build time. Tasks that specify the size of the partition in which they are running use this keyword to identify how much dynamic space they may allocate to symbol tables or buffers. Tasks such as MACRO-11, TKB, PIP, VFY, and AT. use the space from the end of the task image to the end of their address space as dynamic memory. In effect, the INC keyword defines the number of words in this dynamic region.

Default: size specified at task-build time, or zero.

PAR - /PAR=pname. A partition name that overrides the partition specified at task-build time. In an unmapped system, the specified partition must have the same base address as the partition for which the task was built.

If the keyword specifies a nonexistent partition, the system issues a warning message and attempts to install the task in the GEN partition.

Default: TKB uses the name GEN. (See the RSX-11M System Generation Manual for explanation of multiuser system defaults.)

PMD - /PMD=option. Post-Mortem Dump option. If PMD=YES, a post-mortem dump is requested for the task if it aborts because of an SST error condition. If PMD=NO (the default from a task-build), no dump is requested at SST abort time. To generate a dump, the post-mortem dump task PMD... must be installed in the system. This keyword overrides the /PM switch specified at task-build time.

Default: specified at task-build time.

PRI - /PRI=number. Priority number. The value range is 1 through 250(10). Number conventions apply.

Default: /PRI=50.

SLV - /SLV=option. Slave task option. The two options are YES and NO. If the command specifies /SLV=YES, data sent to the installed task is marked with the TI: of the sending task. When the installed task receives the data, the system sets the task's TI: to that associated with the data. This keyword overrides the /SL switch specified at task build.

If the command specifies /SLV=NO the TI: of the task does not change when receiving data.

Default: Specified at task build.

TASK - /TASK=taskname. The name by which the task is to be referred.

Default: first 6 characters of the filename or the name specified at task-build time. (See the RSX-11M System Generation Manual for explanation of multiuser system defaults.)

UIC - /UIC=[group,member]. The square brackets are required syntax. This UIC is the default for the task. It can be overridden in the Run command. The task UIC determines into what file-protection class (system, owner, group, world) a task belongs, and thus directly influences file access. Note that file protection may differ from file to file.

Default: default from task build. This default applies only to tasks started by a RUN$ directive or a Run command that specifies a time delay before runtime. Tasks started by typing the name as a command (for example, PIP and TKB) run under the default UIC of the terminal from which the command was issued. This UIC is defined by the Set command.

Examples:

```
>INSTALL SCAN
```

Install task SCAN. Priority and UIC defaulted.

```
>INSTALL DK1:[1,111]SCAN.TSK;4/PRI=103
```

Install task SCAN, file type TSK, version 4 on device DK1:, group number 1, member number 111, with a priority of 103(8).

```
>INS MAC/INC=4096./PAR=SYSCTL
```

Install task MAC in the system-controlled partition SYSCTL and increase the task size by 4096 decimal words.

Command Error Messages:

INS -- ADDRESSING EXTENSIONS NOT SUPPORTED

The command tried to install a task with a VSECT (virtual section) into a system that does not support the feature.

INS -- BASE ADDRESS MUST BE ON 4K BOUNDARY

The base virtual address of the task was not on a 4K boundary. This message is applicable only to mapped systems.

INS -- BASE MISMATCH COMMON BLOCK <common-name>

The base address of the partition did not match that of the common block. This message is applicable only to unmapped systems.

INS -- CHECKPOINT AREA TOO SMALL

The area allocated for checkpointing the task was smaller than the partition into which the task was being installed. The system does not support the dynamic allocation of checkpoint space.

INS -- CHECKPOINT SPACE TOO SMALL, USING CHECKPOINT FILE

A warning message. The checkpoint space allocated in the task image file is too small for the size of the task (usually because of the /INC keyword). The system supports the dynamic allocation of checkpoint space and will therefore use a checkpoint file to contain the rolled-out task.

INS -- COMMON BLOCK IS TASK PARTITION <common-name>

A task's request for access to a common block was rejected because the requested partition was a task partition.

INS -- COMMON BLOCK NOT LOADED <common-name>

The specified common block was linked to the task but had not been installed into the system. Install the common block, then install the task.

INS -- COMMON BLOCK OCCUPIED

An attempt was made to load a common block that was occupied.

INS -- COMMON BLOCK PARAMETER MISMATCH <common-name>

Parameters of a common block did not match those in the task's label block.

INS -- COMMON BLOCK SEGMENT COUNT ERROR

The command installed a resident library with overlays, but the number of overlay segments specified in the task header is different from the number specified in the segment descriptions.

INS -- FILE NOT CONTIGUOUS

An attempt was made to install a task from a non-contiguous file. Task images may exist only in contiguous files.

INS -- FILE NOT TASK IMAGE

Data in the label block was not correct, indicating that the file was not a task image.

INS -- ILLEGAL DEVICE ddnn:

The device specified at task build by the assign (ASG) options did not exist in the system. This is a warning message; the task will be installed.

INS -- ILLEGAL DEVICE/VOLUME

The specified device was not a valid task-residence device.

INS -- ILLEGAL FIRST APR

A privileged task, built to run using APR 4 or 5 as its base, was too large. Using APR 4, the task was larger than 12K; using APR 5, the task was larger than 8K. If the Executive has been built to support 20K of address space, APR 5 is the only valid base APR for tasks mapping into the Executive. A privileged task that does not map into the Executive must use APR 0 as its base. (See the RSX-11M Task Builder Reference Manual for a description of the /PR keyword.) This message applies only to mapped systems.

INS -- ILLEGAL PRIORITY

The value of the priority specified in the command was out of range (that is, not 1 to 250(10), inclusive).

INS -- INSTALLING PRIVILEGED TASK FROM NON-PRIVILEGED TERMINAL

This message applies to multiuser protection systems only. A nonprivileged user attempted to install a privileged task. Only a privileged user can install a privileged task.

INS -- LENGTH MISMATCH COMMON BLOCK <blockname>

The length parameter for the common block, as described in the label block for the task image, did not match the corresponding length parameter defined in the system. A task's label block data must match system data for that task before it can be installed.

INS -- NO CHECKPOINT SPACE, ASSUMING NOT CHECKPOINTABLE

This is a warning message. The command attempted to install the task as checkpointable; but the task was not built as checkpointable and therefore had no checkpoint space allocated in its task image file. Because the system does not support the dynamic allocation of checkpoint space, it considers the task not to be checkpointable.

INS -- NOT ENOUGH APRS FOR TASK IMAGE

The Task Builder allows the user to specify the virtual base address of a task image as a multiple of 4K. Privileged tasks start at virtual 100000(8) to map the first 16K of the Executive, or virtual 120000(8) to map the first 20K of the Executive, at the same time as the user task. If the virtual base address is set too high, the task image may not be able to be mapped with the remaining mapping registers. This message is applicable to mapped systems only.

INS -- NO ROOM AVAILABLE IN STD FOR NEW TASK

No dynamic memory was available to build the System Task Directory (STD) entry (Task Control Block [TCB]); therefore, no task could be installed.

INS -- PARTITION NOT COMMON

A partition specified for a common area or library is not defined as a common partition.

INS -- PARTITION par NOT IN SYSTEM, DEFAULTING TO GEN

This is a warning message. The specified partition does not exist; the system therefore attempts to install the task in the GEN partition.

INS -- WARNING -- PRIVILEGED TASK OVERMAPS THE I/O PAGE

A privileged task that is mapped into the Executive is usually also mapped into the I/O page in order to access the KT11 registers and other devices. Many tasks do not require this access and can use the additional 4K words of virtual address space for the task image. This message warns that a task that may require access to the I/O page might have grown too large.

INS -- PRIVILEGED TASK LARGER THAN 12K

A privileged task was larger than 12K. Privileged tasks on an unmapped system have a maximum size of 12K.

INS -- SPECIFIED PARTITION FOR COMMON BLOCK

A task was being installed into a common block.

INS -- SPECIFIED PARTITION TOO SMALL

The task being installed was larger than the partition into which it was being installed.

INS -- TASK AND PARTITION BASES MISMATCH

The base of the partition did not match the base of the task being installed. This message applies only to unmapped systems.

INS -- TASK IMAGE I/O ERROR

Install could not read the task-image file, or Install could not rewrite the task-image header. (The device is write-locked.)

INS -- TASK IMAGE ALREADY INSTALLED

The requested task image, which was checkpointable, had already been installed. On systems that do not support the dynamic allocation of checkpoint space, checkpointable tasks that have checkpoint space allocated can be installed only once; other tasks can be installed more than once.

INS -- TASK IMAGE VIRTUAL ADDRESS OVERLAPS COMMON BLOCK <block-name>

The virtual addresses reserved for the task image overlap those reserved for the common block specified in the message. A corruption of the task image file probably caused the overlap.

INS -- TASK NAME ALREADY IN USE

An attempt was made to install a task with the same name as one already in the system.

INS -- TOO MANY COMMON BLOCK REQUESTS

A task made too many common-block requests. A task is limited to seven common-block references.

INS -- TOO MANY LUNS

A task requested more than 255(10) LUN's to be assigned.

INS -- UNDEFINED COMMON BLOCK <blockname>

A task referenced a common block that had not been defined in the system. Usually, this message indicates that the task was built for another system.

LOA

LOAD (P)

The Load command reads a nonresident device driver into memory and constructs the data base and interrupt-vector linkages required to allow access to the device. This command eliminates the need to link infrequently used device drivers permanently into the Executive. (Executive support of loadable device drivers is a system generation option.)

The command includes an optional keyword that specifies the partition into which the driver is to be loaded. The specified partition cannot be a common partition; but all other types of partition are permissible. If the command does not name a partition, the system loads the driver into the partition chosen when the driver was built.

Format:

```
LOA[D] dd:[/PAR=parname]
```

where:

dd: - a 2-character ASCII device name

parname - the name of the partition into which the driver is to be loaded.

Examples:

```
LOAD   MM:                 !LOAD TU16 DRIVER (MMDRV)

LOA    LS:/PAR=DRVPAR      !LOAD LPS11 DRIVER (LSDRV) INTO DRVPAR
```

Notes:

- Guidelines for building loadable device-driver tasks are rigidly defined. These guidelines are described in the RSX-11M Guide to Writing an I/O Driver. Note that the system generation procedures include provisions to simplify the task building of drivers.

- It is not possible to LOAD all device drivers. Also, not all loadable drivers can have loadable data bases.

- The Load command requires that a driver's TSK and STB files reside under the system UIC on the system disk (LB:). For example, the command

  ```
  LOA  DT:
  ```

  requires that the files DTDRV.TSK and DTDRV.STB reside on LB: under the appropriate system UIC. The command SET /SYSUIC displays and defines the system UIC; the usual values, however, are [1,50] for an unmapped system and [1,54] for a mapped system.

- The Load command does not initiate checkpointing to free a partition.

- The Load command requires pool space as follows:

  1. If the driver data base is being loaded, the command needs pool space equal in size to the size of the data base.
  2. In a mapped system, the command needs pool space for an Interrupt Control Block (ICB) for each interrupt entry in the driver.
  3. If the driver is being loaded into a system-controlled partition, the command needs pool space for a Partition Control Block (PCB).

Command Error Messages:

LOA -- DEVICE dd: NOT IN SYSTEM

The data base for the device driver was neither in the system nor in the driver's task image.

LOA -- DEVICE NOT MOUNTED

The system device (LB:) was not mounted.

LOA -- DRIVER ALREADY RESIDENT

The specified device driver had already been loaded or is permanently resident.

LOA -- DRIVER BEING LOADED

The Executive was effecting another request to load the driver.

LOA -- DRIVER BUILT WITH WRONG EXECUTIVE STB FILE

The STB file for the driver was not compatible with the current Executive.

LOA -- DRIVER LOAD/UNLOAD CALLS NOT SUPPORTED

The requested driver included the symbols $ddLOA or $ddUNL in its source. These symbols are reserved for future use by the Load and Unload commands.

LOA -- INVALID DRIVER DATA BASE AT OFFSET xxxx IN FILE name

The driver had an invalid data base value at offset xxxx relative to the symbol $ddDAT.

LOA -- FILE name HAS ILLEGAL STB FORMAT

The driver's STB file contained an illegal object code or data.

LOA -- FILE name NOT A VALID DRIVER TASK IMAGE

The driver's task image was either overlaid or had a task header.

LOA -- FILE name NOT CONTIGUOUS

The file ddDRV.TSK was not contiguous. The file must be contiguous.

LOA -- ILLEGAL DRIVER TASK APR USAGE

A device driver must be built for APR 5 and must be less than 4K (e.g. PAR=DRVPAR:1200 00:20000). This message applies to mapped systems only.

LOA -- ILLEGAL VALUE FOR SYMBOL symname IN FILE name

The symbol symname is defined to an illegal value by file ddDRV.STB. This can occur for several reasons; some general and some specific to individual symbols. For example, symbols cannot have odd values; most symbols cannot be defined as zero; $ddTBL must exist and have a value less than $ddEND; mapped loadable drivers cannot reference $INTSV.

LOA -- INSUFFICIENT POOL SPACE

The required amount of pool space was not available.

LOA -- INTERRUPT VECTOR ALREADY IN USE

A driver interrupt vector does not point to one of the nonsense interrupt entries.

LOA -- INVALID INTERRUPT VECTOR

The interrupt vector address specified in the driver data base was too high for the current system.

LOA -- LOADABLE DRIVER SUPPORT NOT IN SYSTEM

The system generation option to support loadable device drivers was not chosen.

LOA -- PARTITION parname BUSY

The user-controlled partition or subpartition into which the driver was to be loaded was already occupied, or there was not enough room in a system-controlled partition. (The Load command does not initiate checkpointing.)

LOA -- PARTITION parname IS A COMMON

An attempt was made to load a device driver into a common partition. A device driver cannot be loaded into a common partition.

LOA -- PARTITION parname NOT IN SYSTEM

The partition specified either in the Load command or at task build was not in the system.

LOA -- PARTITION parname TOO SMALL

The partition into which the driver was to be loaded was not large enough to contain it.

LOA -- SYMBOL symname IS DOUBLY DEFINED BY FILE name

The symbol symname is defined twice by the file ddDRV.STB

LOA -- SYMBOL symname IS UNDEFINED BY FILE name

The symbol symname was found in the file ddDRV.STB, but it was not a symbol definition.

LOA -- TASK AND PARTITION parname BASES MISMATCH

The base address of the partition into which the driver was to be loaded did not match the base address of the partition address for which the task was built. This message applies to unmapped systems only.

LUN

LOGICAL UNIT NUMBERS (P & NP)

The LUN command displays at the entering terminal the static LUN assignments for a specified task. (Static assignments are those recorded in the task's disk-image file.) The display consists of a list of physical device units in one column, with the corresponding LUN's in an adjoining column.

When the specified task is running, the command display does not necessarily reflect the running task's assignments. An Executive directive issued from within the task (Assign LUN) could have altered the LUN assignments.

If the specified task was initiated by the install, run, and remove on exit option of the MCR RUN command (see Section 5.5) the task has no static LUN assignments. The message

```
LUN -- TASK NOT IN SYSTEM
```

is displayed.

Format:

```
LUN[S] taskname
```

Example:

```
>LUN XKE
CL0:  3
TT3:  4
TT3:  5
DK0:  6
TT3:  7
```

The display shows static assignments for LUN's 3, 4, 5, 6, and 7, as recorded in the task-image file header; no other LUN's are statically assigned to the task XKE.

Command Error Message:

LUN -- NO LUNS

The task that is the argument of the LUN's command did not have any logical units. This is not precisely an error message, but rather an indication that there were no assignments to display.

LUN -- TASK NOT IN SYSTEM

The specified task is not installed or, the task was initiated by the install, run, and remove n exit option of the MCR RUN command. This is not precisely an error message, but rather an indication that there were no assignments to display.

MOU

MOUNT (P & NP)

The Mount command logically connects devices to Ancillary Control Processors (ACP).

- For Files-11 volumes other than magnetic tapes, the Mount command creates the Volume Control Block (VCB) and declares that the volume is logically online for access by the file system. The VCB is allocated in dynamic memory and controls access to the volume.
- For Files-11 magnetic tapes, which conform to ANSI standards, the Mount command allocates the Volume Set Control Block (VSCB) and the mounted and unmounted volume lists. The command then requests the ACP to mount the volume set. It is necessary to MOUnt a magnetic tape only if it is an ANSI standard tape. FCS treats unmounted magnetic tapes as unit record devices.
- For network channels, the Mount command informs the network ACP that the device is available for use in the network. It allocates a communication VCB and issues a Mount request to the ACP.

In multiuser protection systems, a nonprivileged user can mount a volume on the user's private device (that is, a device that the user has allocated). Otherwise, the operator must issue the command from a privileged terminal.

To mount DECtapes as Files-11 volumes, the command defaults the ACP to DTAACP, or to F11ACP if DTAACP is not present in the system. (Note that the keyword /ACP always overrides the default.) The use of DTAACP has the advantage that it avoids tying up the disk ACP (F11ACP) during the relatively long searches inherent in DECtape processing.

If powerfail recovery support was selected during SYSGEN, there is a slight delay when MOU is specified for a disk that is off-line.

The description of the Mount command format consists of three parts which separately illustrate the parameters and keywords associated with Files-11 volumes other than magnetic tape, Files-11 ANSI magnetic tapes, and network devices, respectively. All the keywords are then collectively described below.

Files-11 Format:

```
MOU[NT] ddn:[volume-label][keyword(s)]
```

Note that a hyphen (-) can be used as a line terminator to extend the MOU command line when selected keywords cause the command to exceed 80 characters (or whatever buffer size has been specified for the entering terminal). Any number of continuation lines is allowed, but the total command line cannot exceed 512. characters.

Keywords:

/ACP=taskname

/EXT=block-count

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

/LRU=FCB-count

/OVR

/UIC=[uic]

/UNL

/VI

/WIN=retrieval-pointer-count

Mount keywords for Files-11 devices override corresponding values in the volume's home block.

where:

ddn: - Device-unit on which the volume is to be mounted.

Default: none; must be specified.

volume-label - The Files-11 volume label may be up to 12 characters in length. It is used to ensure that the correct volume is being mounted. In a multiuser protection system, the command must include a volume label unless the /OVR keyword is present.

Default: 12 nulls; no label check is performed.

Files-11 (ANSI) Magnetic Tape Format:

```
MOU[NT] ddn:  label [keyword(s)]
```

Keywords:

/ACP=taskname

/BYPASS

/DENS=tape-density

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

/NOLAB

/OVR

/OVRFSID

/OVREXP

/UIC=[uic] /VI

where:

dd - 2-character device name (e.g., MT)

n - A unit number. The number of units specified can be less than or equal to the number of tape volumes in the set. If the command specifies more than one unit number, the numbers must be separated by commas and enclosed in parentheses, e.g. (0,1,2).

label - A tape label. If the command specifies more than one label, the labels must be separated by commas and enclosed in parentheses. The number of labels can be less than, equal to, or greater than the number of units.

Initially, label-1 must be physically loaded on unit dd1:. The remaining labelled tapes may or may not be loaded on other units specified in the command. Volume labels must be specified in the order in which the volumes comprise the volume set because the system searches for and/or requests each successive volume in the specified label order. Since the system searches according to label, the second volume onward can reside on any of the specified units.

Network Format:

MOU[NT] ddn:[/keyword(s)]

Keywords:

/ACP=taskname

/PRTCL=network-protocol

/RCK=name of a redundancy check algorithm

/TEL=telephone number of a dial-up link

where:

ddn: - Device-unit to be made available for user in the network.

Keyword Definitions:

ACP - /ACP=taskname. The task name of an Ancillary Control Processor designed to process this volume.

Default: depends on device type, as indicated below:

For Files-11 volumes, /ACP=F11ACP
For Files-11 (ANSI) magnetic tape,
  /ACP=MTAACP
For network channels, /ACP=NT....

BYPASS - /BYPASS. The BYPASS keyword instructs the Mount command to bypass the tape label processor.

DENS - /DENS=tape-density. The density (Bits Per Inch or BPI) at which data is to be written to the tape. The density value is either 800 or 1600. /DENS=800 indicates 800 BPI NRZI; /DENS=1600 BPI Phase Encoded.

Default: /DENS=800

EXT - /EXT=block-count. The number of blocks by which a file that exhausts its space allocation can be extended.

Default: taken from the home block.

FPRO - /FPRO=[system,owner,group,world]. Keyword for default file protection. Access codes consist of four 4-code groups in the Access Rights word, as follows:

R = Read
W = Write
E = Extend
D = Delete

In each instance, the absence of the code means that the user is denied the access right. The square brackets are required syntax.

Default: values taken from volume home block or tape label.

NOTE

Protection code subparameters (system, owner, group, world) are positional. That is, the location of the word in the parameter string defines the user to whom the code applies. The order is:

system, owner, group, world

The order of appearance of the characters R, W, E, and D is fixed.

R, if desired, must be first.
W, if desired, is next.
E, if desired, is next.
D, if desired, is last.

Thus, RWE and RE are acceptable, whereas WR and DEWR are not.

LRU - /LRU=FCB-count. The number of directory File Control Blocks (FCB's) kept in memory by the ACP per volume. The more FCB's kept in memory, the faster a file can be found. If the file is in a directory for which the FCB is resident, the overhead required to open the directory is bypassed and thus the file can be found more quickly.

Default: /LRU=3.

NOLAB - /NOLAB. The tape volume set does not have a label.

OVR - Privileged option. /OVR. The OVR keyword instructs the Mount command to override label processing so that a volume can be mounted without specifying a volume label.

OVRFSID - /OVRFSID. The OVRFSID keyword instructs the Mount command to override existing file set identifiers.

OVREXP - /OVREXP. The OVREXP keyword instructs the Mount command to override the expiration date on the volume.

PRTCL - /PRTCL=network-protocol. The name of the network line protocol.

Default: /PRTCL=DDCMP.

RCK - /RCK=algorithm-name. The name of the redundancy checking algorithm routine.

Default: /RCK=CRC16.

TEL - /TEL=number. The telephone number of a line to a remote node. Up to 12 digits may be specified.

Default: Null.

UIC - /UIC=[uic]. User identification code - [group,member|.

UNL - /UNL. Specify the volume index file as unlocked. When locked, the index file is read-only; when unlocked, it is read/write.

Default: locked.

VI - /VI. Display the volume information on the entering terminal. The information displayed is that under which the volume was mounted. The volume information format is:

```
DEVICE = ddnn:
TYPE   = type
LABEL  = volume label
UIC    = [group,member]
ACCESS = [system, owner, group, world]
CHARAC = [ ]
```

WIN - /WIN=number. The number of mapping pointers to be allocated for file windows.

Default: If space is available and the I/O optimization option was selected at system generation, the number of pointers required for all index file retrieval pointers to reside in memory; otherwise, the number of pointers specified in the home block.

Examples:

```
>MOU DK1:
```

Mount the volume on device-unit DK1:. (In this example, the optional parameters have not been specified; therefore, the parameter values in the volume's home block are used.) No label checking is performed. No volume information is displayed.

```
>MOU MT(0,1,2):(VOL1,VOL2,VOL3,VOL4,VOL5)
```

Mount the volume set on units MT0:, MT1:, and MT2:. Initially, VOL1 must be physically loaded on MT0:; the first volume in a set must always be loaded on the first unit specified in the command. VOL2 and VOL3 may or may not be loaded when the command is issued. If they are loaded, they can reside on the remaining units in any order (for example, VOL3 could be loaded on MT1:). The system searches for the next volume in a set by volume label rather than by unit number; the order in which volumes are listed in the command is the order in which the system searches for and/or requests the volumes. If VOL2 and VOL3 are not loaded, the system requests each in turn, as well as VOL4 and VOL5. The volumes can be loaded on either MT0:, MT1:, or MT2:.

```
>MOU DK1:SYS004
```

Mount the volume labeled SYS004 on DK1:

```
>MOU XP:/TEL=16176468600
```

Mount the DP11 link and dial the remote node, whose telephone number is 16176468600.

Command Error Messages:

MOU -- ACP NOT IN SYSTEM

The task specified as ACP or default ACP was not installed in the system.

MOU -- ALREADY MOUNTED

The specified network device-unit was already mounted.

MOU -- DEVICE ATTACHED[-dev:]

The device-unit specified in the command was attached by a task and could not be mounted. For attempts to mount one or more magnetic tapes, the message includes a specific device-unit.

MOU -- DEVICE NOT DEFINED IN NETWORK

The specified device-unit was not defined in the current network topology and therefore could not be mounted.

MOU -- DEVICE NOT IN SYSTEM -dev:

The Mount command specified a tape drive that has not been generated into the system.

MOU -- DEVICE OFFLINE[-dev:]

The device specified in the command, although generated into the system, was not physically present in the host configuration. If the offline device is a magnetic tape drive, the message includes the device-unit.

MOU -- DEVICE SPECIFIED TWICE -- dev:

The Mount command specified the same device-unit twice.

MOU -- DRIVER NOT LOADED

The driver of the device to be mounted has not been loaded. Load the appropriate driver and reissue the Mount command. (See the description of the Load command.)

MOU -- FILE HEADER READ ERROR

Mount could not read either the index file header or the storage allocation file.

MOU -- HOME BLOCK READ ERROR

An I/O error was detected in trying to read the home block. This message usually indicates that the volume is not ready. Wait until it is ready and reissue the command.

MOU -- ILLEGAL KEYWORD COMBINATION

The Mount command specified conflicting keywords.

MOU -- MOUNT ERROR FROM ACP xxx.

The ACP detected an error while trying to mount the volume set. The following is a list of the ACP error code numbers and their respective meanings.

Decimal Value	Meaning
6	Get command line error
7	Syntax error
8	Bad device
11	Wrong volume (incorrect label)
12	All units must be TU16s
13	Not ANSI format
17	Privilege violation
18	Units have different track specifications (cannot mix 7- and 9-track units)
19	Message task (F11MSG) not installed

(See the IAS/RSX-11 I/O Operations Reference Manual for a further discussion of FCS error codes.)

MOU -- MULTI-UNIT DEVICE NOT TAPE

The command specified multiple units that are not tape drives.

MOU -- NT DEVICE NOT MOUNTED

The device NT: was not mounted. This device must always be mounted before any other device in a network.

MOU -- NOT MOUNTABLE DEVICE

The specified device was not supported as a Files-11 device (including ANSI magnetic tape) or a network device.

MOU -- OTHER VOLUME MOUNTED[-dev:]

An attempt was made to mount a volume on a device that already had a mounted volume. The message specifies the device-unit if it is a tape drive.

MOU -- RETRIEVAL POINTERS WRONG FORMAT

The index-file retrieval pointers were not in the correct format for RSX-11M.

MOU -- STORAGE BIT MAP FILE READ ERROR

An I/O error was encountered while reading the storage allocation file.

MOU -- TASK NOT ACP

The task specified as an ACP did not have the characteristics of an ACP.

MOU -- WRONG VOLUME

The volume label and the label specified in the command did not match.

MOU -- VOLUME STRUCTURE NOT SUPPORTED

RSX-11M did not support the Files-11 structure level of the volume being mounted.

OPE

OPEN REGISTER (P)

The Open Register command allows examination and optional modification of a word of memory. The command cannot open a location within a task unless the task is fixed in memory.

Format:

```
OPE[N] memory address [+ or -n][/Keyword]

(memory-address) (contents-of-address)/[value]<line-terminator>
```

The Open command, as shown above, is a 2-line command. The first line initiates the command. Its parameters are as follows:

memory-address - A 1- to 8-digit octal memory address.

+ or -n - One or more optional octal numbers to be added to or subtracted from the memory address.

Keywords:

```
/TASK=taskname
/PAR=partition name
/KNL
/DRV=dd:
```

The keywords apply only to mapped systems.

After accepting the first input line, Open finds the memory location to be accessed by using either:

- The keyword (in a mapped system), or
- The specified address as the actual address.

TASK causes the task partition to be located; PAR causes the named partition to be located; KNL causes the Executive memory (the first 32K of memory) to be accessed; and DRV causes the loadable driver for device dd: to be accessed. If no keyword is specified, the absolute address is opened.

When TASK, PAR, and DRV are used, the supplied memory address (a virtual address in the partition) is modified by + or - n. The absolute address thus formed is used to display the second line shown in the command format. In particular, the virtual address (task, partition, or kernel) and its contents are displayed. The operator may then enter an optional replacement value, followed by a line terminator.

In a mapped system, specifying an illegal location or accessing beyond the limits of the partition causes an error. The user is limited to the specified memory region (TASK, PAR, or DRV). However, if the command omits a keyword, it can access any address of physical memory. In an unmapped system, any location in physical memory can be accessed.

The line terminator directs the subsequent action of Open. Any value that is entered replaces the contents of the word whose address and current value are shown in the first part of the line-2 display. The slash (/) is part of the line-2 display.

Line Terminator Options:

ESC - ESCape or ALTMODE: ESC (ALTMODE) terminates acceptance of further input and is the only means of exit from the MCR Open function.

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

^CR - Up-arrow (circumflex on some terminals) carriage return: the previous location is opened.

*CR or @CR - Asterisk carriage return, or at-sign carriage return: the location pointed to by the contents of the opened location is opened.

>CR - Right angle bracket carriage return; the contents of the open location is treated as a branch instruction. The effective address of the branch is the next opened location. It is not permissible to supply a value (modify the current open location) and close the location with a >.

Example:

```
>OPE 4+10/TASK=CYCLE
000014 060014/350<ESC>
```

In this example, the value of the virtual address of the task CYCLE is 4, while n equals 10, yielding the effective task-relative address of 000014. The system responds by printing the virtual address (000014), and the contents of the address (060014), followed by the slash. The operator responds by entering the new value 350 and the ESCape character. The value 350 replaces the previous contents of task virtual location 000014.

Command Error Messages:

OPE -- BYTE ADDRESS

The address specified as the argument to the Open command was an odd address.

An even address (for example, 3000) is legal; an odd address (for example, 3001) is illegal.

OPE -- INPUT I/O ERROR

In attempting to read the next command, Open detected an error.

OPE -- INVALID ADDRESS

The address specified as an argument in the Open command referenced a nonexistent memory location, an address outside of the specified partition, or an address outside of the task's virtual address space.

OPE -- TASK NOT FIXED

An attempt was made to open a task virtual location. Since the task was not fixed, it may or may not have been in memory at the time the command was issued. This message indicates that the specified task was not in memory when the command was issued.

PAR

PARTITION DEFINITIONS (P & NP)

The Partition Definitions command displays on the entering terminal a description of each memory partition in the system.

The display consists of five columns that specify:

1. Partition name,
2. Partition base address (octal),
3. Partition size (octal),
4. Partition kind, main partition (MAIN) or subpartition (SUB), and
5. Partition type:

    TASK for user-controlled,
    COM for common,
    DEV for device registers,
    SYS for system-controlled,
    taskname for task region,
    DYNAMIC for dynamically created region, or
    DRIVER for region occupied by a loadable driver.

COM partitions are used for resident libraries and common data areas. DEV partitions are used to allow tasks to communicate with specific device registers, such as the UDC and ICS/ICR-11 industrial control subsystems. See Chapter 1 for a description of system and user-controlled partitions, as well as subpartitions. Dynamic and task regions are defined in the RSX-11M Executive Reference Manual. See the description of the Load command for information about loadable drivers.

Format:

```
PAR[TITIONS]
```

Examples:

```
Name   Base   Size   Kind Type

>PAR
LDR    000000 000000 MAIN TASK
SYSPAR 120000 010000 MAIN TASK
FCPPAR 130000 026000 MAIN TASK
PMDPAR 156000 020000 MAIN TASK
SPLPAR 156000 010000 SUB  TASK
DRVPAR 176000 014000 MAIN SYS
       176000 001600 SUB  DRIVER - DB:
       177600 000500 SUB  DRIVER - DS:
       200300 001000 SUB  DRIVER - DK:
       201300 001100 SUB  DRIVER - DT:
       202400 001000 SUB  DRIVER - LP:
       203400 003100 SUB  DRIVER - MM:
GEN    212000 546000 MAIN SYS
       212000 013400 SUB  (RMDEMO)
       225400 045700 SUB  (...EDI)
       302100 034200 SUB  (...AT.)
       336300 117100 SUB  (...TKB)
       455400 045700 SUB  (EDIT13)
       523300 010000 SUB  (...SYS)
       601500 117100 SUB  (...MAC)
```

```
 Name   Base    Size    Kind Type

>PAR
LDR     000000 000000 MAIN TASK
GEN     100000 360000 MAIN SYS
        100000 010000 SUB  (...MCR)
        110000 040000 SUB  (TT2    )
        150000 010000 SUB  (...SYS)
        173000 011100 SUB  (MES001)
        210000 024100 SUB  (F11ACP)
TPSPAR  460000 240000 MAIN SYS
SCBLIB  460000 000000 SUB  DYNAMIC
COM001  460000 004500 SUB  DYNAMIC
POL001  464500 110300 SUB  DYNAMIC
TPMLIB  720000 020000 MAIN COM
TSTLIB  740000 010000 MAIN COM
MESS    750000 010000 MAIN COM
```

REASSIGN (P)

The Reassign command allows the operator to reassign a task's Logical Unit Numbers (LUNs) from one physical device-unit to another. The reassignments affect only the static assignments recorded in the task's disk image file.

The command does not affect a memory-resident task. Hence, Reassign has no effect on the assignments of a currently executing task, nor on a task that is fixed in memory. The Redirect command (see next page) is used for dynamic device reconfiguration.

Format:

```
REA[SSIGN] taskname lun ddn:
```

where:

```
  lun  - Logical unit number
 ddn:  - New device-unit, which can be a physical, logical, or
         pseudo device.S
```

Examples:

```
>REA JOE 3 TT0:
```

Reassign LUN 3 of task JOE to device TT0:.

```
>REA BILL 2 TT1:
```

Reassign LUN 2 of task BILL to device TT1:.

```
>REA TEST 3 XX:
```

Reassign LUN 3 of task TEST to logical device XX:.

Command Error Message:

```
REA -- LUN OUT OF RANGE
```

An attempt was made to reassign a LUN that was greater than the maximum number of LUN's allocated during task build.

RED

REDIRECT (P)

The Redirect command allows the operator to redirect all I/O requests previously directed to one physical device-unit to another physical device-unit. The Redirect command is especially useful if I/O units required for a task are inoperable. The command does not affect any I/O requests aleady in the I/O queue.

The following device-units cannot be redirected:

- The device TI:,
- An attached device, and
- A device-unit containing a mounted volume.

In addition, one pseudo device cannot be redirected to another pseudo device.

Format:

```
RED[IRECT] Nddn:=Oddn:
```

where:

Nddn: - New device-unit to which subsequent requests will be redirected.

Oddn: - Old device-unit from which requests have been directed.

Examples:

```
>RED TT3:=TT6:
```

Redirect all I/O requests for device TT6: to device TT3:.

```
>RED TT:=LP:
```

Redirect all I/O requests for device LP0: to device TT0:.

Command Error Messages:

RED -- CIRCULAR REDIRECT ERROR

The attempt to redirect a device would result in a circular list of redirections.

RED -- DEVICE NOT REDIRECTABLE

An attempt was made to redirect a device that cannot be redirected.

RED -- NEW DEVICE NOT KNOWN TO SYSTEM

The new device in the Redirect command was not known to the system (did not exist in the device tables).

RED -- OLD DEVICE ATTACHED

An attempt was made to redirect an attached device.

RED -- OLD DEVICE MOUNTED

An attempt was made to redirect a mounted device.

RED -- OLD DEVICE NOT KNOWN TO SYSTEM

An attempt was made to redirect an unknown device (the device did not exist in the device tables).

RED -- PSEUDO DEVICE REDIRECT ERROR

An attempt was made to redirect one pseudo device to another pseudo device.

RED -- TI REDIRECT ERROR

An attempt was made to redirect the pseudo device TI:. This device may not be redirected.

REM

REMOVE (P)

The Remove command allows the operator to delete a task name from the System Task Directory (STD), and thereby remove the task from the system. The removal of a task from the system makes the task unknown to the system. This action is the complement of Install. All that remains of the task in the system is the task-image file.

To remove a task that is in execution, the user must first abort the task.

Format:

```
     REM[OVE] taskname
```

Example:

```
     >REM SCAN
```

Remove the task named SCAN from the system. The task-image file remains in the systemhe

Notes:

- If a task that is the object of a Remove command is fixed, the command automatically unfixes the task and then removes it.
- Remove automatically cancels all time-based schedule requests for the specified task.
- Remove automatically unlinks a task from all ICS/ICR interrupts.
- Remove automatically deallocates all receive-by-reference and receive-data packets for the task, and detaches all attached regions. (See the RSX-11M Executive Reference Manual.)

RES

RESUME (P & NP)

The Resume command allows the operator to continue the execution of a previously suspended task.

A Resume command directed to a task not initiated from the entering terminal can only be issued from privileged terminals.

Format:

```
RES[UME] taskname
```

Example:

```
>RES XKE

     Resume task XKE.
```

Note:

- A task cannot suspend any task other than itself. The Resume command allows such a suspended task to proceed.

Command Error Message:

```
RES -- TASK NOT SUSPENDED
```

The task used as the argument of the Resume command was not suspended.

RUN

RUN (P & NP)

The Run command initiates the execution of a task. The command can cause a task to be:

- Started immediately
- Started at a time increment from now
- Started at a time increment from clock unit synchronization
- Started at an absolute time of day
- Installed, immediately run, and removed on exit

All these options, except install-run-and-remove, are available with or without rescheduling.

Run has five format variations, as described below.

Formats:

1. Run immediately.

   RUN taskname [/RSI=magu][/UIC=[uic]]

   where:

   taskname - A 1- to 6-character taskname.

   RSI - The reschedule interval -- The format is the same as for dtime, which is detailed below under format 2. The reschedule interval specifies how often the task is to be rerun. Thus, each time the specified interval of time lapses, an initiation request is made for the specified task.

   Default: no rescheduling.

   UIC - (Privileged option in multiuser protection systems only.) The uic has the format [g,m] where g and m represent the group and member numbers respectively; legal group and member numbers range from 1 through 377(8). The square brackets are required syntax. This is the UIC under which the task will be requested. The UIC determines which files the task may access.

   Default: the value established for the terminal from which the Run command is entered.

2. Run at a time increment from now.

   RUN taskname dtime [/RSI=magu][/UIC=[uic]]

where:

taskname = A 1- to 6-character taskname.

dtime = The time at which the task will be initiated in time units from command issuance. It follows the form magu.

A time parameter consists of two fields:

a. A magnitude field

b. A units field

where:

mag = The magnitude
u = Units, one of the following letters:

T
S
M
H

The legal value of the magnitude is related to the value of the units field, which is encoded as:

T = Ticks -- A tick is a clock interrupt, and the rate at which interrupts occur depends on the type of clock installed in the system.

For a line frequency clock, the tick rate is either 50 or 60 per second, corresponding to the line frequency.

For a programmable clock, a maximum of 1000 ticks per second is available. (The frequency is selected at system generation.)

S = Seconds

M = Minutes

H = Hours

The magnitude is the number of units to be clocked, but the magnitude value cannot exceed 24 hours in the specified units. If the magnitude is equal to 0, then the RSI is set to 1. For example, /RSI = 0H is equivalent to /RSI = 1H.

Units = T -- Any positive value is valid (maximum of 15 bits).

Units = S -- Any positive value is valid (maximum of 15 bits).

Units = M -- The maximum m +tude is 1440(10).

Units = H -- The maximum magnitude is 24(10).

RSI = See Format 1, above.

UIC = See Format 1, above.

3. Run at a time increment from clock unit synchronization.

The task start-time is determined by first waiting for the next time unit's occurrence (hour, minute, second, tick) and then waiting for the specified time increment to elapse.

```
Run taskname sync [dtime][/RSI=magu][/UIC=[uic]]
```

where:

taskname = A 1- to 6-character taskname.

sync = H = synchronize on the next hour;
M = synchronize on the next minute;
S = synchronize on the next second, and
T = synchronize on the next tick.

dtime = As defined in Format 2, above; dtime, if present, is added to the synchronization unit to produce the actual run time for running the task.

RSI = See Format 1, above.

UIC = See Format 1, above.

4. Run at an absolute time of day.

```
RUN taskname atime [/RSI=magu][/UIC=[uic]]
```

where:

taskname = A 1- to 6-character taskname.

atime = Absolute time of day when this task will be initiated. Input format is hh:mm:ss.

hh = hours (decimal assumed);
mm = minutes (decimal assumed), and
ss = seconds (decimal assumed).

RSI = See Format 1, above.

UIC = See Format 1, above.

5. Install, run immediately, and remove on exit.

When Run tries to start a task as a type 1 request, it searches the system task directory (STD) for the task; if it finds the entry, it proceeds to run it. However, if Run does not locate the task, it attempts to install a task whose filename is given as the task name. The directory under which Run looks for this file is determined by the presence or absence of a leading $. If no $ is found, Run looks under the UIC on device SY: of the terminal requesting the Run. If $ is present, it looks under the system directory on device LB:. The SET /SYSUIC command defines the system directory; however, conventional values are [1,50] for an unmapped system and [1,54] for a mapped system.

```
RUN  [ddn:][$]filename [/keyword(s)]
```

Keywords:

```
/CKP=option

/INC=size

/PAR=pname

/PMD=option

/PRI=number

/SLV=option

/TASK=taskname

/UIC=[group,member]
```

where:

ddn: - Device-unit on which task image file resides.

$ - Specifies that the file is found under the system UIC on device LB: (if ddn: is not specified).

filename - The name of the task-image file to be run.

CKP - /CKP=option. Checkpoint option. If CKP=YES (default for checkpointable tasks), checkpointing is allowed. If CKP=NO, checkpointing is disabled for the task.

The /CKP keyword overrides the /CP switch specified at task build.

On systems that support the dynamic allocation of checkpoint space, the task image file does not need to contain allocated checkpoint space. The presence of the /CKP=YES keyword instructs the system to allocate checkpoint space for the task when required from a checkpoint file.

INC - /INC=words. Number of words by which the task is to be increased. This keyword applies to system-controlled partitions only. (See the description of the Install command for a more complete explanation.)

PAR - /PAR=parname. Parname specifies the partition in which the task is to be installed.

Default: Partition specified at task build.

PMD - /PMD=option. Post-Mortem Dump option. If PMD=YES, a post-mortem dump is requested for the task if it aborts because of an SST error condition. If PMD=NO (the default from a task-build), no dump is requested at SST abort time. To generate a dump, the post-mortem dump task PMD... must be installed in the system. This keyword overrides the /PM switch specified at task-build time.

Default: specified at task-build time.

PRI - /PRI=number. Priority number. The value range is 1 through 250(10). Number conventions apply.

Default: /PRI=50.

SLV - /SLV=option. Slave task option. The two options are YES and NO. If the command specifies /SLV=YES, data sent to the installed task is marked with the TI: of the sending task. When the installed task receives the data, the system sets the task's TI: to that associated with the data. This keyword overrides the /SL switch specified at task build.

If the command specifies /SLV=NO the TI: of the task does not change when receiving data.

Default: Specified at task build.

TASK - /TASK=taskname. The specified task name is to be assigned temporarily to the task while it is running. If taskname is not found in the STD, MCR names it TTnn by default, where nn is the unit number of the requesting terminal.

UIC - /UIC=[uic]. The square brackets are required syntax. This is the UIC under which the task will be requested.

Default: The value established for the terminal that issues the RUN command.

Examples:

```
>RUN XKE 15M
```

RUN task XKE 15 minutes from command issuance.

```
>RUN XKE 15M/RSI=90S/UIC=[3,1]
```

RUN task XKE 15 minutes from command issuance, rescheduling it every 90 seconds, with a UIC code of [3,1].

```
>RUN DK2:$MAC
```

Run task in file DK2:MAC.TSK from the system directory. The task is automatically installed, run, and removed on exit.

Notes:

- If the command is a run immediate, establish TI: as the terminal from which the task was initiated. Otherwise, the system sets the TI: terminal to CO:.
- A Run Immediately, if specified without the /RSI option and terminated with an ESCape character instead of a carriage return, echos the default prompt when the requested task exits and not when MCR has finished initiating the task. This facility enables the operator to determine that a task that is producing no output on a terminal has exited.
- The install, run, and remove form establishes the terminal from which the command was initiated to be the TI: terminal and causes the default prompt to be displayed on TI: when the task exits. This form of the command is especially useful when the space available in the dynamic storage region is small.

Command Error Messages:

RUN -- INVALID TIME PARAMETER

A specified time field was incorrect.

RUN -- TASK BEING ABORTED

A request for the execution of a task was made, but the subject task was being aborted either because of an unrecoverable error or an Abort directive issued by the operator.

SAV

SAVE (P)

The Save command writes into a task-image file the image of an RSX-11M system that has been resident in main memory. The command effectively saves the image so that a hardware bootstrap or the Boot command can later be used to reload and restart it. The saved system is written into the file from which it was originally booted. This command is intended to provide the user with a facility for building development systems that have tasks already installed, and thus eliminate the need for repetitive task installation following every system bootstrap.

The Save command removes any installed tasks that were not loaded from LB: and verifies that the system is inactive by making the following checks:

1. No tasks have outstanding I/O,

2. No devices are mounted,

3. No checkpoint files are active, and

4. Error logging has been turned off.

An error is reported if any of these checks fails.

All RSX-11M system images reside on a file-structured volume as a special format of task image. This special format is a task image without a task header. There may be more than one system image on a volume (for example, a program development system and a production or test system).

A system can either be booted via the hardware bootstrap, in which case a specified file is booted (see /WB), or it can be booted via the Boot command. Note that the unit number for the drive must be supplied at sysgen; the system must have support for the specified unit. Also, a system saved on one controller cannot be booted from another controller.

When a user installs a task, the system stores the task's file identification in the task header. When a system is saved, it places the file identification, rather than the file's Logical Block Number (LBN), in the Task Control Block (TCB). Then when the system is rebooted, it reopens the task file and stores the new LBN of the task in the TCB. If a task has been deleted, the system cannot open the task file when the system is rebooted, and the system automatically removes the task's TCB from the STD.

A saved system does not retain the physical disk addresses of installed tasks. The Disk Save and Compress (DSC) utility can compress and copy a disk containing a saved system, thus moving the task files to different physical locations. But since the TCB entries contain task file identifications, rather than LBNs after a system save, the system can function normally when it is rebooted.

When the bootstrap block is written, the physical disk-block address of the system-image file is stored with it. However, the file can be deleted. If there is file system activity, the blocks previously allocated to the system image may be reallocated to another file. A subsequent bootstrap that uses the boot block could possibly cause random data to be loaded. Also, when the disk save and compress utility copies a disk, the physical address of a system changes. The system therefore modifies the pointer to the system image address that appears in the boot block to reflect its new position.

Since Save is active when the memory-resident system image is copied to disk, Save appears in this image as an active task to VMR. This appearance is correct because Save is actually active and is the program that starts up the saved system after a disk boot.

Format:

```
SAV[E] [/WB]
```

Keyword:

/WB - Indicates that a boot block pointing to the system image is to be written out to the system device. The new boot block points to the file that is saved with the issuance of the command. Thus, on the next hardware bootstrap, this saved file will be loaded. If the command omits the /WB switch, the file previously pointed to by the boot block remains in effect; that is, it is not overwritten.

Note that the system does not recognize the presence of the keyword unless a space separates it from the command name.

Example:

```
>SAV
```

The current status of the system is saved on the system disk. System changes made by the Redirect command or other MCR commands are also saved with the system image that is resident in main memory.

Command Error Messages:

SAV -- BAD TASK VBN, TASK REMOVED - taskname

The system removed the specified task because it contained a bad virtual block number (VBN) in its file header.

SAV -- CHECKPOINT FILE STILL IN USE ON ddnn:

The system cannot be saved because a checkpoint file on the specified device is still active. Deallocate the checkpoint file (see the ACS command) and reissue the Save command.

SAV -- ERROR LOGGING STILL ACTIVE

A system cannot be saved while error logging is active. Run the task ERF to disable error logging.

SAV -- HOME BLOCK READ ERROR

A fatal error. The system detected an error when rebooting a saved system. The error was detected while reading the disk's home block.

SAV -- INDEX FILE HEADER CHECKSUM ERROR

While rebooting a saved system, the system detected a checksum error in the file header of the index file. The error causes the system to go into a hard loop.

SAV -- INDEX FILE HEADER READ ERROR

While rebooting a saved system, the system detected an error in the file header of the index file. The error causes the system to go into a hard loop.

SAV -- I/O OUTSTANDING

Tasks remained in the system with I/O requests waiting to complete. A system cannot be saved with I/O outstanding.

SAV -- LABEL BLOCK I/O ERROR

In attempting to save the system image, Save wrote the transfer address in the label block of the system-image file. An error occurred during this write attempt.

SAV -- LOAD DEVICE NOT LB:, TASK REMOVED - taskname

The system removed the specified task because the task had not been installed from LB:.

SAV -- MUST BE RUN FROM TERMINAL CO:

SAV is not being run from the terminal to which the pseudo device CO: is directed.

SAV -- NOT VALID SAVE DEVICE

A system may be saved only on a Files-11 device.

SAV -- RUNNING ABOVE SYSTEM IMAGE FILE LIMIT

The system image file created to contain the saved system is RSX11M.SYS. This error message appears when RSX11M.SYS is not large enough to hold the memory in which both the system and the Save task run. The Save task must be included in the saved system image because Save is used to reboot the system. Either run the Save task in a partition lower in memory or rebuild the system with a larger system image file.

SAV -- SYSTEM MAY NOT BOOT CORRECTLY

The [system uic] SAV.TSK was not found on LB:. If SAV must be checkpointed in order to redirect and mount the booted device, the system disk will be overwritten, causing unpredictable results.

SAV -- TASK FILE DELETED, TASK REMOVED - taskname

While rebooting a saved system, a file identification that appears in the Task Control Block (TCB) could not be found in the index file on LB:. The file has therefore been deleted; deleted files do not contain valid task images.

SAV -- TASK FILE HEADER ERROR, TASK REMOVED - taskname

While rebooting a saved system, a task's file header could not be found and the task was therefore removed.

SAV -- TASK HEADER READ ERROR

The system detected an I/O error while reading in all task headers to store file identifications in the TCB. The error causes the system to go into a hard loop.

SAV -- TASK FILE HEADER READ ERROR, TASK REMOVED - taskname

The system removed the specified task because it encountered a failure while reading the task's file header.

SAV -- VOLUME STILL MOUNTED ON ddnn:

The volume on device ddnn: was still mounted when the Save command was issued. A system cannot be saved with volumes mounted. Dismount the volumes and retry the Save command. If necessary, use the MCR Devices command to determine which volumes are mounted.

SAV -- VOLUME STRUCTURE NOT SUPPORTED

While rebooting a saved system, the system determined from the disk's home block that the disk's structure is not supported by the current version of RSX-11M. This error causes the system to go into a hard loop.

SET

SET (P & NP)

The Set command allows the user to alter dynamically a number of system-wide and local terminal characteristics. Some options are privileged. The following two lists describe all the available options under the appropriate privilege heading. The text in parentheses specifies keywords associated with the options.

Nonprivileged options:

- Establish certain terminal characteristics for the issuing terminal only (BUF, CRT, ESCSEQ, HOLD, LA30S, LA180S, LOWER, SLAVE, SPEED, VT05B),
- Establish the user's own private device as public (PUB),*
- Set the default UIC for the issuing terminal (UIC), and
- Display the status (determined by the keyword specified) of all terminals.
- Display statistics about system dynamic memory (POOL).

Privileged options:

- Establish device characteristics for any terminal in the system (BUF, CRT, ESCSEQ, HOLD, LA30S, LA180S, LOWER, REMOTE, SPEED, and VT05B),
- Establish the default UIC for any terminal in the system (UIC),
- Establish the system UIC used by the Load command and the install-run-remove option of the Run command (SYSUIC),
- Set the privilege status of any terminal in the system (PRIV),
- Set any terminal to slave status (SLAVE),
- Set any device to public status so that any one in the system can access the device (PUB),
- Create partitions or subpartitions (MAIN and SUB),
- Add space to the dynamic core pool (POOL),
- Enable write check on devices that support it (WCHK).
- Specify the maximum number of I/O packets to be preallocated to optimize I/O performance (MAXPKT),
- Specify the maximum size to which a task can extend itself by means of the EXTEND TASK system directive (MAXEXT), and
- Enable or disable the ability to log onto terminals in a multiuser protection system (LOGON).

* Valid in multiuser protection systems only.

Improper use of the Set command from a privileged terminal can result in a situation in which privileged operator input will not be accepted. The following two situations should be avoided:

- All privileged terminals in the system are set to slave and cannot enter unsolicited input to MCR. Non-privileged terminals cannot enter privileged commands and therefore, cannot reset the privileged terminals to non-slave. See the description of the /SLAVE keyword below.

- The buffer length of all the non-slaved, privileged terminals is shorter than 14 bytes. The terminal buffer is then too short to enter many important commands required to continue system operation to alter the status of other terminals. See the description of the /BUF keyword below.

A number of Set command options (CRT, ESCSEQ, HOLD, LA180S, LOGON, LOWER, MAXEXT, MAXPKT, PUB, and WCHK) are features that must be included at system generation. If support for a specified option has not been generated into the system, the Set command issued has no effect.

Format:

```
SET /Keyword=values
```

Only one keyword per command is permitted. Most keywords can be prefixed by NO to negate or disable the function of the keyword. Exceptions are BUF, MAXEXT, MAXPKT, POOL, SPEED, SYSUIC, and UIC; the system returns a syntax error whenever NO precedes one of these keywords.

Device Characteristics Keywords:

```
/BUF=dev:[size]

/CRT[=dev:]

/ESCSEQ[=dev:]

/HOLD[=dev:]

/LA30S[=dev:]

/LA180S[=dev:]

/LOWER[=dev:]

/PRIV[=dev:]

/PUB[=dev:]

/REMOTE[=dev:]

/SLAVE[=dev:]

/SPEED=dev:[recv:xmit]

/UIC[=uic[:dev:]]

/UIC[=dev:]

/VT05B[=dev:]

/WCHK[=dev:]
```

Keyword Definitions:

BUF - /BUF=dev:[size]. Set or display the default buffer size of the specified device. The specified size must be greater than 0 and less than or equal to 255(10). If dev: is a terminal, size must be greater than 2 and less than or equal to 255(10). BUF is particularly useful for defining line printer width (80 or 132 columns).

When size is omitted, the system displays at the issuing terminal the buffer size of the specified device.

CRT - /CRT[=dev:] where dev: is a Cathode Ray Tube (CRT) terminal. The CRT keyword defines the terminal as a CRT and enables the backspace deletion option. Typing the RUBOUT or delete key moves the printing position one space to the left and erases any character displayed in that position.

When the keyword omits =dev:, the system displays at the issuing terminal all the devices that are defined as CRT's.

NOCRT - /NOCRT[=dev:] where dev: is a Cathode Ray Tube (CRT) terminal. The NOCRT keyword deletes the CRT status of the specified device and disables the backspace deletion option.

When the keyword omits =dev:, the system displays at the issuing terminal all the terminals that are not defined as CRT's.

ESCSEQ - /ESCSEQ[=dev:] where dev: is a terminal. The ESCSEQ keyword enables the recognition of escape sequences from the specified terminal. The terminal driver treats the ESC character as the beginning of an escape sequence rather than as a line terminator. See the RSX-11M I/O Drivers Reference Manual for further information about escape sequences.

When the keyword omits =dev:, the system displays at the entering terminal all the terminals that recognize escape sequences.

NOESCSEQ - /NOESCSEQ[=dev:] where dev: is a terminal. The NOESCSEQ keyword disables the recognition of escape sequences at the specified terminal.

When the keyword omits =dev:, the system displays at the issuing terminal all the terminals that do not support the sending of escape sequences.

HOLD - /HOLD[=dev:] where dev: is a CRT display terminal. The /HOLD keyword enables hold-screen mode at the specified terminal. When hold-screen mode is enabled, the terminal displays output one screenful at a time. The screenful of output remains on the screen until the user types SHIFT/SCROLL (types the SCROLL key while pressing the SHIFT key). The terminal then refills the screen. The user can also advance the screen contents one line at a time by typing the SCROLL key.

Hold-screen mode is useful when copying files to terminals that normally transmit output at very fast baud rates. Note, however, that hold-screen mode may not be appropriate for use with certain types of programs (editors or terminals using different speed for input and output, for example).

If the keyword omits =dev:, the system displays at the issuing terminal all the terminals currently in hold-screen mode.

NOHOLD - /NOHOLD[=dev:] where dev: is a CRT display terminal. The /NOHOLD keyword disables hold-screen mode at the specified terminal.

If the keyword omits =dev:, the system displays at the issuing terminal all CRT terminals that have disabled hold-screen mode.

LA30S - /LA30S[=dev:]. Establishes the specified terminal device as an LA30S and generates fill characters required for an LA30S.

If =dev: is omitted, all terminals defined as LA30S are displayed.

NOLA30S - /NOLA30S[=dev:]. Resets the terminal to a device other than an LA30S.

If =dev: is omitted, all non-LA30S terminals are displayed.

LA180S - /LA180S[=dev:]. Establishes the specified terminal device as an LA180S and causes the system to suppress normal conversion of form feeds to 8-line feeds.

If the keyword omits =dev:, all terminals defined as LA180S are displayed.

NOLA180S - /NOLA180S[=dev:]. Resets the terminal to a non-LA180S device.

If the keyword omits =dev:, all non-LA180S terminals are displayed.

LOWER - /LOWER[=dev:]. Specifies that lower-case characters are not to be converted to upper case when they are received from the specified terminal.

If =dev: is omitted, all terminals for which lower-case characters are not converted are displayed.

NOLOWER - /NOLOWER[=dev:]. Resets the terminal characteristics so that lower-case characters received on input are automatically converted to upper case and echoed as upper case.

Absence of =dev: results in the display of all terminals for which conversion to upper case is specified.

PRIV - /PRIV[=dev:]. Sets the specified terminal to privileged status.

If =dev: is omitted, MCR displays all privileged terminals.

NOPRIV - /NOPRIV[=dev:]. Sets the identified terminal to nonprivileged status.

Absence of =dev: results in a display of all nonprivileged terminals.

PUB - /PUB[=dev:]. (Multiuser protection systems only). The keyword establishes the specified device as a public device. When the keyword does not specify a device name, the command displays all the current public devices at the issuing terminal. SET /PUB is not a privileged form of the command.

When the keyword specifies a device name, the command is privileged, except in the following case. A nonprivileged user can set as public (or nonpublic) the user's private (allocated) device. When the user deallocates the device, the device automatically loses its public status.

In all other cases, a user must be privileged to issue the SET /PUB=dev: form of the command.

NOPUB - /NOPUB=[dev:]. (Multiuser protection systems only). Causes the specified device to lose its public status. Nonprivileged users can specify their private devices that previously were made public. In all other cases, however, only privileged users can issue the SET /NOPUB=dev: command.

When the keyword omits =dev:, the system displays at the issuing terminal all private and unowned devices.

REMOTE - /REMOTE[=dev:] where dev: is a terminal with a line to a DZ11 multiplexer. The keyword declares that the specified terminal is connected to a modem and can be connected by means of a dial-up network.

When the keyword omits =dev:, the system displays at the issuing terminal all remote lines (including DH11 lines).

NOREMOTE - /NOREMOTE[=dev:] where dev: is a line to a DZ11 multiplexer. The NOREMOTE keyword establishes the terminal's line as a local line not connected to a modem.

When the keyword omits =dev:, the system displays at the issuing terminal all local lines (including DL11, DH11, DJ11, and DZ11 multiplexers).

SLAVE - /SLAVE[=dev:]. Establishes the specified terminal as one that can enter data only if it is solicited from a task; thus, the specified terminal always rejects unsolicited input (other than CTRL/O, CTRL/Q, and CTRL/S).

Absence of dev: results in the display of all terminals currently classified as slaves.

NOSLAVE - /NOSLAVE[=dev:]. Sets the specified terminal to nonslave status.

Absence of =dev: results in the display of all nonslave terminals.

SPEED - /SPEED=dev:[recv:xmit]. Establishes the receive and transmit baud rate for terminals attached to the system through a DH11 or DZ11 multiplexer. The argument "recv" is the baud rate at which characters are input to the computer from the terminal; and "xmit" is the baud rate at which the computer outputs characters to the terminal. Both must be specified in setting the speed. If neither is specified, the current settings will be displayed.

Valid baud rates for the DH11 are: 0, 110, 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, A (user-specified baud rate A), and B (user-specified baud rate B).

Valid baud rates for the DZ11 are: 0, 110, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, and 9600. The receive and transmit baud rates for the DZ11 must be the same; the DZ11 does not support split speeds.

UIC - /UIC[=uic[:dev:]]. The specified UIC is established as the default UIC for the entering terminal unless dev: is specified, in which case the default UIC is set for that terminal. If the system supports multiuser protection, and the user is privileged, the command also sets the terminal's protection UIC. (See Section 3.2.1.) All tasks run from the terminal are run with the UIC of the terminal, except in a multiuser protection system. In such a system, a privileged user can override the terminal UIC with the UIC switch in the Run command. Further, external MCR function tasks* are requested with this UIC.

If a UIC is not specified, the current UIC is displayed. If only dev: is specified, the UIC for the specified terminal is displayed, provided that the terminal issuing the command is privileged.

VT05B - /VT05B[=dev:]. Sets the specified terminal as a VT05B (a 2400 baud VT05) and generates fill characteristics of a VT05B.

If =dev: is omitted, all terminals classified as VT05B are displayed.

NOVT05B - /NOVT05B[=dev:]. Resets the terminal to other than a VT05B. If =dev: is omitted, all terminals classified as non-VT05B terminals are displayed.

WCHK - /WCHK[=dev:]. Specifies for all Files-11 devices except DX: and DL: and magnetic tape (MT:, MM:, MS:) that all writes are to be followed by a write check. This feature provides high reliability for data transfers to the specified disk. If =dev: is omitted, all Files-11 disk drives with write check enabled are displayed.

NOWCHK - /NOWCHK[=dev:]. Specifies that write check is to be disabled for the specified device. If =dev: is not specified, all Files-11 devices except DX: and DL: and magnetic tape (MT:, MM:, MS:) that have write check disabled are displayed.

Examples:

```
>SET /BUF=LP0:
BUF=LP0:132.
```

Display the current buffer size of LP0:

```
>SET /BUF=TT1:40.
```

Set TT1: buffer size to 40(10).

* These tasks are the MCR Mount, Install, Dismount, and UFD commands (which run as tasks) and all system-supplied software.

```
>SET /CRT=TT3:
```

Enable the backspace deletion option on the display terminal identified by TT3:.

```
>SET /CRT
CRT=TT3:
```

Display all terminals that support the backspace deletion option.

```
>SET /ESCSEQ=TT3:
```

Establish terminal TT3: as a terminal that can send escape sequences.

```
>SET /ESCSEQ
ESCSEQ=TT3:
```

Display all terminals that can send escape sequences.

```
>SET /NOESCSEQ=TT3:
```

Disable support of escape sequences on TT3:.

```
>SET /HOLD=TT3:
```

Enable hold-screen mode on CRT terminal TT3:. The terminal erases its screen and positions the cursor at the top left of the screen.

```
>SET /HOLD
HOLD=TT3:
```

Display all terminals that have hold-screen mode enabled.

```
>SET /NOHOLD
NOHOLD=TT1:
NOHOLD=TT2:
```

Display all CRT terminals that are not currently in hold-screen mode.

```
>SET /LA30S=TT2:
>SET /LA30S
LA30S=TT0:
LA30S=TT2:
```

Set TT2: as an LA30S.
Display all LA30S terminals.

```
>SET /NOLA30S=TT0:
```

Set TT0: to other than an LA30S terminal.

```
>SET /PRIV=TT0:
```

Set TT0: to a privileged terminal.

```
>SET /PRIV
PRIV=TT0:
PRIV=TT1:
```

Display all privileged terminals.

```
>SET /REMOTE=TT21:
```

Set the terminal line corresponding to TT21: as a remote dial-in line. The line must be connected to a DZ11 multiplexer.

```
>SET /REMOTE
REMOTE=TT1:
REMOTE=TT2:
REMOTE=TT3:
REMOTE=TT4:
REMOTE=TT21:
```

Display all terminals that are currently set as remote dial-in lines. TT1:, TT2:, TT3:, and TT4: are DH11 lines; TT21: is a DZ11 line.

```
>SET /SLAVE=TT3:
```

Set TT3: to a slave terminal.

```
>SET /VT05B
VT05B=TT4:
VT05B=TT5:
VT05B=TT6:
```

Display all VT05B terminals.

```
>SET /WCHK=DK1:
```

Enable write check on DK1:.

Enable or Disable Logon Keyword:

LOGON - /LOGON. Enables logging onto terminals in a multiuser protection system. The keyword is privileged.

NOLOGON - /NOLOGON. Disables logging onto terminals in a multiuser protection system. This is a privileged keyword. LOGON and NOLOGON clear and set a flag checked by the Hello command when a user attempts to log onto a terminal. This Set option provides a means of preventing users from accessing the system during system maintenance or account file maintenance activity, for example.

I/O Packet Keyword:

MAXPKT - /MAXPKT[=n]. The /MAXPKT keyword establishes the maximum number of I/O packets to be retained in a separate queue for use by the QIO directive. The specified number (n) of retained packets is used to optimize the servicing of QIO requests.

If the keyword omits n, the system displays at the issuing terminal the maximum number and the current number of available I/O packets in the format:

```
MAXPKT=m.:c.
```

where m. is the maximum number of and c. is the current number of available I/O packets.

Examples:

```
>SET /MAXPKT=12.
```

Set the maximum number of I/O packets retained for optimization of I/O servicing to 12(10).

```
>SET /MAXPKT
MAXPKT=12.:3.
```

The maximum number of I/O packets to be retained explicitly for QIO processing is 12(10). Three packets are currently available.

System UIC Keyword:

SYSUIC - /SYSUIC[=uic] where uic has the format [g,m]; g and m are octal numbers that represent a group and member number respectively. The /SYSUIC keyword establishes the UIC for the system and all system tasks. On most unmapped systems, the system UIC is [1,50]; on most mapped systems, the system UIC is [1,54]. These UIC values are the initial defaults on virgin systems. This UIC becomes the default UIC used by the Load command and the install-run-remove option of the Run command.

When the keyword omits =uic, the system displays at the issuing terminal the current system UIC.

Examples:

```
>SET /SYSUIC
SYSUIC=[1,54]
```

Display the current system UIC.

```
>SET /SYSUIC=[3,54]
```

Set the system UIC to [3,54].

Memory Allocation Keywords:

```
/MAIN=pname[:base:size:type]

/NOMAIN=pname

/SUB=mname:pname[:base:size]

/NOSUB=mname:pname

/MAXEXT[=size]

/POOL[=top]
```

These keywords are used as follows:

MAIN - The MAIN keyword establishes a main partition.

pname	- The 1- to 6-character alphanumeric partition name.
:base	- Physical base address of the partition specified as a number of 64-byte blocks.
:size	- Size of the partition specified as a number of 64-byte blocks.
:type	- TASK for a user-controlled task partition, SYS for a system-controlled task partition, COM for a common partition, or DEV for a common partition mapping into the device registers. User-controlled and system-controlled task partitions are allocated for user tasks. Common partitions are used for resident libraries and common blocks. Device partitions are used by nonprivileged tasks to access device registers in the external page, such as those for the UDC and ICS/ICR-11.

Base and size may be entered in any of the following formats; nnnn is a number (in bytes) to which a multiplication factor is applied to determine the value used.

Format	Calculated Value
nnnn	Octal (nnnn*100)
nnnn.	Decimal (nnnn.*64.)
nnnK	Octal K (nnn*4000)
nnn.K	Decimal K (nnn.*2048.)

Valid ranges of calculated values (in words) are:

Octal	0<=base<177777	0<size<2000
Decimal	0.<=base<65535	0.<size<1024.
Octal K	0K<=base<3777K	0K<size<40K
Decimal K	0.K<=base<2047.K	0.K<size<32.K

For system-controlled partitions, size may range from 0 to 1920.K or any valid representation of these limits.

Example:

Any of the following inputs for size will allocate a 1920. byte partition.

```
40
32.
1K
1.K
```

If the command does not specify base, size, and type, MCR displays the values of these parameters for the named partition. If the named partition is a main partition, MCR also displays the parameters of all its subpartitions.

NOMAIN - The NOMAIN keyword eliminates the named partition or library from the system.

SUB - The SUB keyword establishes a subpartition. Subpartitions may only be established in user-controlled main partitions.

mname - The 1- to 6-character main partition name. The subpartition being defined becomes a subpartition in the named main partition.

:pname - The 1- to 6-character subpartition name. The subpartition being defined becomes a subpartition of the main partition specified in mname.

:base - Physical base address of the subpartition, specified as a number of 64-byte blocks.

:size - Size of the subpartition, specified as a number of 64-byte blocks.

Base and size may be entered in any of the following formats; nnnn is a number (in bytes) to which a multiplication factor is applied to determine the value used.

Format	Calculated Value
nnnn	Octal (nnnn*100)
nnnn.	Decimal (nnnn.*64.)
nnnK	Octal K (nnn*4000)
nnn.K	Decimal K (nnn.*2048.)

Valid ranges of values (in words) are:

Octal	0<=base<10000	0<size<2000
Decimal	0.<=base<4096.	0.<size<1024.
Octal K	0K<=base<200K	0K<size<40K
Decimal K	0.K<=base<2047.K	0.K<size<32.K

Example:

Any of the following inputs for size will allocate a 2048. byte partition.

40
32.
1K
1.K

If base and size are omitted, the current values will be displayed for the named subpartition.

NOSUB - The NOSUB keyword eliminates the named subpartition from the system.

MAXEXT - /MAXEXT[=size]. The MAXEXT keyword establishes the maximum size to which a task can extend itself by means of the EXTEND TASK directive (see the RSX-11M Executive Reference Manual). The keyword is privileged and is only valid on mapped systems that chose to support the EXTEND TASK directive at system generation. The maximum size can be expressed in four formats:

```
n
nK
n.
n.K
```

See the description of /MAIN or /SUB for a definition of these size values.

If the keyword omits =size, the system displays at the issuing terminal the current maximum size in the format:

```
MAXEXT=m
```

where m is an octal number equal to the maximum size allowed for a task, in 64-byte blocks (for example, 1200 indicates 20K words).

POOL - The /POOL=top keyword increases the size of the dynamic storage region.

top - First location in memory to be used for user partitions. The location is specified in units of 64-byte blocks. It can be given in octal or decimal. If top is not given, the system will print out the current top of memory, the size of the longest block of pool space in words, and the total number of words in the pool.

The format is:

```
POOL = top:max:total
```

where top is given in units of 64-byte blocks. Max and total are in decimal words.

Notes:

- If a partition currently contains a task, and a command (for example, /NOMAIN or /NOSUB) attempts to eliminate the partition, the system rejects the command.
- A command that eliminates a main partition also eliminates all its subpartitions. A command that eliminates a specified subpartition eliminates only that specified subpartition.
- If a command attempts to eliminate a main partition that has subpartitions in which tasks are installed, the system rejects the command until those tasks are removed from the subpartitions.

- When defining a partition (main or sub) the name must not already be defined as a partition. In addition, a main partition can not overlap any other main partition; nor can a subpartition of a main partition overlap any other subpartition of the main partition.
- The numeric value convention applies to all the Set command numeric values.
- All partitions except LDR are considered user partitions by the POOL command. Thus, for example, SYSPAR is considered a user partition.
- Once space has been allocated to the dynamic storage region, the space can never be recovered for use in partitions.
- System-controlled partitions can not include subpartitions explicitly defined by the Set command. An attempt to define subpartitions of system-controlled partitions in this way will be rejected. The Executive dynamically defines subpartitions of a system-controlled partition as needed for tasks installed in the system-controlled partition.

Examples:

```
>SET /MAIN=SYSPAR:420:140:TASK
```

Define a main partition called SYSPAR as a user-controlled partition whose base address is 42000(8) and whose length is 14000(8) bytes.

```
>SET /SUB=SYSPAR:MCRPAR:445:56
```

Define a subpartition called MCRPAR of main partition SYSPAR whose base address is 44500(8) and whose length is 5600(8) bytes.

```
>SET /MAIN=SYSPAR
MAIN=SYSPAR:0420:0140:TASK
SUB=SYSPAR:MCRPAR:0445:0056
```

Display the parameters of the main partition SYSPAR and all of its subpartitions.

```
>SET /NOSUB=SYSPAR:MCRPAR
```

Eliminate subpartition MCRPAR.

```
>SET /NOMAIN=SYSPAR
```

Eliminate main partition SYSPAR.

```
>SET /MAIN=GEN:1000:6000:SYS
```

Define a main partition called GEN as a system-controlled partition whose base address is 100000(8) and whose length is 600000(8) bytes.

```
>SET /MAXEXT=20K
```

Establishes 20K as the maximum size to which the task can extend itself.

```
>SET /MAXEXT
MAXEXT=1200
```

```
>SET /POOL=420
```

Set the beginning of user partitions to 42000(8).

```
>SET /POOL

POOL=0420:150:430.
```

The first location for user partitions is 42000; the longest free block is 150. words, and the total of all pool space is 430. words.

Command Error Messages:

SET -- ALIGNMENT ERROR

An attempt was made to create a partition, but the base address or size conflicted with existing partitions or physical memory size.

SET -- DEVICE NOT VARIABLE SPEED MULTIPLEXER

An attempt was made to set the baud rate for a terminal that was not attached to a DH11 or DZ11 multiplexer.

SET -- DEVICE NOT TERMINAL

An attempt was made to set terminal characteristics for a nonterminal device.

SET -- INVALID SPEED

The multiplexer line specified does not support the requested speed; or the command specified unequal releive and transmit speeds for a DZ11. The DZ11 does not support split speeds.

SET -- FEATURE NOT SUPPORTED

The command keyword specifies an optional feature that was not incorporated into the system at system generation.

SET -- LINE NOT DZ11

The command attempted to set to remote a line that was not attached to a DZ11 multiplexer.

SET -- NON EXISTENT MEMORY

An attempt was made to define a partition in nonexistent memory.

SET -- PARTITION ALREADY EXISTS

An attempt was made to define a partition with a name already in use.

SET -- PSEUDO DEVICE ERROR

The command attempted to make a pseudo device public. Pseudo devices cannot be public devices.

SET -- SPACE USED

An attempt was made to create a partition or subpartition in a storage area already occupied.

SET -- TASK INSTALLED IN PARTITION

An attempt was made to eliminate a main partition or subpartition containing installed tasks.

SET -- TOO MANY SUBPARTITIONS

An attempt was made to define more than seven subpartitions. A main partition is limited to a maximum of seven subpartitions.

SET -- UNKNOWN MAIN PARTITION

An attempt was made to define a subpartition of a nonexistent main partition.

SET -- USER PARTITION >32K NOT PERMITTED

The command attempted to create a partition greater than 32K. User-controlled partitions greater than 32K cannot be created in systems that do not support extended logical address space, a system generation option.

SET -- WRITE CHECK NOT SUPPORTED

An attempt was made to enable write check on a device that was not a DB:, DF:, DK:, DM:, DP:, DR:, or DS:.

TAL

TASK LIST - ATL FORMAT (P & NP)

This command displays on the entering terminal the names and status of all tasks or of a specific task installed in the system in the format of the ATL command. See the ATL command for a description of the printout.

Format:

```
     TAL [taskname]
```

Example:

```
>TAL
. LDR. 052220  LDR     052164 00000000-00000000 PRI - 248.  DPRI - 248.
   STATUS:  -CHK FXD STP PRV
   TI - CO0:  IOC - 0.  EFLG - 000001 000000 PS - 170000  PC - 041350
   REGS 0-6  000162 004030 177777 105312 064254 105260 052132
TKTN   105010  SYSPAR 107734 00110000-00120000 PRI - 248.  DPRI - 248.
   STATUS:  -EXE OUT -CHK -PMD PRV
   TI - CO0:  IOC - 0.  EFLG - 000001 000000
RMDEMO 056214  GEN     053740 00217500-00233300  PRI - 225.  DPRI - 225.
   STATUS: WFR PRV CAL
   TI - TT0:  IOC - 0.  EFLG - 000001 000000 PS - 170010  PC - 125072
   REGS 0-6  123353 000070 000000 125032 001601 131530 120352
                                     .
                                     .
                                     .
PIPT2  061454 GEN      071524 00274300-00335300  PRI - 50.  DPRI - 50.
   STATUS:  OUT CKP TIO STP WFR -PMD REM MCR
   TI - TT2:  IOC - 0.  EFLG - 000003 040000

>TAL RMDEMO
RMDEMO 056214  GEN     053740 00217500-00233300 PRI - 225.  DPRI - 225.
   STATUS:  WFR PRV CAL
   TI - TT0:  IOC - 1.  EFLG - 000000 000000 PS - 170010  PC - 130224
   REGS 0-6  000050 000132 000000 125032 001601 131530 120350
>
```

TASKLIST (P & NP)

The Task List command displays on the entering terminal a description of each installed task. The display contains:

1. Task name,
2. Task version identification,
3. Partition name,
4. Task priority,
5. Size of task in bytes (octal),
6. Load device identification,
7. Disk address logical block number (octal), and
8. Task memory state.

Format:

```
TAS[KLIST]
```

Example:

```
...LDR 07.05  LDRPAR 248. 00002200 LB0:-01035303 FIXED
TKTN   03.3   SYSPAR 248. 00010000 LB0:-01126742
RMDEMO X03.03 GEN    225. 00013700 SY0:-00352100
MTAACP 0006   GEN    200. 00013000 DS0:-00000777
F11MSG V0010  GEN    200. 00005400 SY0:-01053030
MCR... 3.2    SYSPAR 160. 00010000 LB0:-01051626
...MCR 01     GEN    160. 00025000 LB0:-01051676
DCL... 0113A  SYSPAR 160. 00010000 LB0:-00567017 CHECKPOINTED
...DCL 0113A  GEN    160. 00040000 LB0:-01221102
TAST17 01     GEN    160. 00025000 LB0:-01051676
...STO 000005 SYSPAR 151. 00010000 SY0:-01136540
...BYE 01.5   GEN    150. 00040000 LB0:-01051337
F11ACP M0235  FCPPAR 149. 00040000 LB0:-01052445
SECFCP M0235  GEN    149. 00026200 SY0:-00055463
DTAACP M0235  GEN    149. 00026200 SY0:-01052554
...DMO 03.2   GEN    140. 00040000 LB0:-01051403
...INI 03.1   GEN    140. 00040000 LB0:-01051561
...MOU 03.01  GEN    140. 00040000 LB0:-01051653
...UFD V0407  GEN    140. 00040000 LB0:-010
```

(operator typed CTRL/O to suppress further output)

The display contains, in columnar form, from left to right: task name, task version identification, partition name, task priority, task size, load device identification, logical block number on the load device, and task memory state. On a processor with less than 124K words of memory, the size field (fifth column) contains 6-digit rather than 8-digit numbers.

FIXED indicates that the task is fixed in memory.

CHECKPOINTED indicates that the task has been swapped out of memory to make room for a higher priority task.

If the task version identification is missing (the rest of the line is offset to the left) or garbage, the task was installed from a disk that has since been removed from the drive or replaced by another cartridge.

TIM

TIME (P & NP)

The Time command makes it possible to:

- Set the current time of day (privileged),
- Set the current date (privileged), or
- Display the current time and date.

Format:

```
TIM[E] [hrs:mins[:secs]] [ml/day/year]

          or

TIM[E] [hrs:mins[:secs]][day/m2/year]
```

where:

hrs - Hours (range 0-23)

mins - Minutes (range 0-59)

secs - Seconds (range 0-59) (optional)

ml - Numeric representation of month (range 1-12)

m2 - 3-letter abbreviation for month

day - Day (range 1-31)

year - Year (range 0-99). Year is relative to 1900; thus, 1977 is entered as 77.

- If the command specifies neither time nor date, it displays the current time and date on the entering terminal.
- If the command specifies the time and date, it sets the clock and calendar. If only the time is specified, only the clock is set; if only the date is specified, only the calendar is set.
- The time and date can be specified in any order.
- All numeric values are decimal. No terminating period is necessary (or permitted).

Examples:

```
>TIM
10:23:31 03-NOV-77
```

Display the current time and date.

```
>TIM 1:30 11/7/77
```

Set the time to 01:30:00 and the date to 07-NOV-77

UFD

USER FILE DIRECTORY (P & NP)

This MCR command creates a User File Directory (UFD) on a Files-11 volume and enters its name into the Master File Directory (MFD). See Section 3.1.1 for further information about UFDs. The volume must have been subjected to Initvolume and Mount commands before UFDs can be defined. Once a volume has been initialized and mounted, UFDs can be added at any time.

In multiuser protection systems, a nonprivileged user can create a UFD on a volume mounted on the user's private device (that is, a device that the user has allocated). Otherwise, an operator must issue the command from a privileged terminal.

Format:

```
UFD ddn:[volume-label][group,member][/keyword][/keyword]
```

Keywords:

```
/ALLOC=number-of-entries

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

where:

ddn: - Device-unit containing the volume on which the UFD being created will reside.

Default: none; must be specified.

volume-label - If specified, the label is compared with the label on the volume. If the names match, a UFD can be allocated. If they do not match, the command is rejteted.

Default: nulls; no check is made.

[group,member] - The UIC for the UFD. The UIC establishes the owner of the UFD. The square brackets are required syntax.

Default: none; must be specified.

ALLOC - Number of directory entries for which space is to be allocated. The number provided is rounded up to the next multiple of 32(10).

Default: 32(10)

PRO - Establishes access rights for the directory file. Access codes consist of four 4-code groups in the Access Rights word, as follows:

R - Read
W - Write
E - Extend
D - Delete

In each instance, the absence of the code means that the user class is denied the access right.

Defaults: [RWED,RWED,RWED,R]

NOTE

Protection code subparameters (system, owner, group, world) are positional. That is, the location of the word in the parameter string defines the user to whom the code applies. The order is:

system, owner, group, world

The order of appearance of the characters R, W, E, and D is fixed.

R, if desired, must be first.
W, if desired, is next.
E, if desired, is next.
D, if desired, is last.

Thus, RWE and RE are acceptable, whereas WR and DEWR are not.

Example:

```
>UFD DK1:[1,1]
```

Create a user file directory (UFD) for the UIC [1,1] on DK1:. The name and file number of the UFD file are entered into the MFD. The name of the UFD is 001001.DIR;1, and its owner is UIC [1,1].

Command Error Messages:

UFD -- CAN'T READ MCR COMMAND BUFFER

The UFD was initiated by a Run command rather than as an MCR command.

UFD -- DIRECTORY ALREADY EXISTS

The requested UFD already existed on the volume.

UFD -- FAILED TO CREATE DIRECTORY

No space existed on the volume, or an I/O error occurred.

UFD -- FAILED TO ENTER IN MFD

There was no room in the MFD or on the volume, or an I/O error occurred on the volume.

UFD -- HOME BLOCK I/O ERROR

An error was detected in reading the volume home block.

UFD -- NOT FILES-11 DEVICE

The device on which the UFD was to be created was not a Files-11 device, and therefore could not support UFD's.

UFD -- WRITE ATTRIBUTES FAILURE

An error was encountered while writing the attributes of either the MFD or the newly created UFD.

UFD -- WRONG VOLUME

The volume label and the label specified in the command did not match.

UFD -- VOLUME NOT MOUNTED

The volume on which a UFD is to be created must be mounted before accessing the Files-11 structure.

UNF

UNFIX (P)

The Unfix command allows the operator to free a fixed task from memory. The effect is to allow tasks that have been waiting for the partition in which the task resides to compete for the partition. Unfix is the complement of the Fix command.

If a fixed task exits or aborts, it still occupies the physical memory of the partition.

Format:

```
     UNF[IX] taskname
```

Example:

```
     >UNFIX XKE
```

Unfix task XKE, freeing the partition in which it resides for task competition.

Command Error Message:

```
     UNF -- TASK NOT FIXED
```

An attempt was made to unfix a task that was not fixed.

UNL

UNLOAD (P)

The Unload command removes a loadable device driver from memory. If a device is mounted, attached, or has outstanding I/O, its driver cannot be unloaded.

To unload a device driver from memory, the driver's STB file must reside on the system device (LB:) under the system UIC. The SET /SYSUIC command displays and defines the system UIC; but usual values are [1,50] for unmapped systems and [1,54] for mapped systems.

The Unload command cannot remove a data base from memory, even if the data base was loaded by means of the Load command.

Format:

```
UNL[OAD] dd:
```

where:

dd: - A 2-character ASCII device name

Example:

```
>UNL LS:
```

Command Error Messages:

UNL -- DEVICE ddn: HAS UNITS ATTACHED, BUSY, AND/OR MOUNTED

Device ddn: is the first or only unit that is attached, mounted, and/or has outstanding I/O. Therefore, the driver cannot be unloaded.

UNL -- DEVICE ddn: NOT IN SYSTEM

No such device existed in the system.

UNL -- DEVICE NOT MOUNTED

The system device (LB:) was not mounted.

UNL -- DRIVER BUILT WITH WRONG EXECUTIVE STB FILE

The STB file for the driver was not compatible with the current Executive.

UNL -- DRIVER CANNOT BE UNLOADED

The Unload command attempted to remove a permanently resident device driver (that is, a driver linked to the Executive).

UNL -- DRIVER NOT LOADED

The device driver specified in the Unload command was not resident in memory.

UNL -- FILE name HAS ILLEGAL STB FORMAT

The driver's STB file contained illegal object code or data.

UNL -- ILLEGAL VALUE FOR SYMBOL symname IN FILE name

The symbol symname is defined to an illegal value by the file ddDRV.STB. Since ddDRV.STB had to pass inspection by LOA, there probably has been a procedural error or ddDRV.STB has been damaged.

UNL -- LOADABLE DRIVER SUPPORT NOT IN SYSTEM

The system generation option to support loadable device drivers was not chosen.

UNL -- SYMBOL symname IS DOUBLY DEFINED BY FILE name

The symbol symname was found in the file ddDRV.STB, but it was not a symbol definition. Since ddDRV.STB had to pass inspection by LOA, there probably has been a procedural error or ddDRV.STB has been damaged.

UNL -- SYMBOL symname IS UNDEFINED BY FILE name

The symbol symname was found in the file ddDRV.STB, but it was not a symbol definition. Since ddDRV.STB had to pass inspection by LOA, there probably has been a procedural error or ddDRV.STB has been damaged.

CHAPTER 5

INDIRECT COMMAND FILES

This chapter describes indirect command files and how to use them. The sections that follow describe the MCR indirect command file processor and its components and then define all the processor directives that control the execution of an MCR indirect command file.

5.1 INTRODUCTION

An indirect command file is a text file containing a series of commands exclusive to, and interpretable by, a single task. The interpreting task is usually a system-supplied component of RSX-11M/M-PLUS, such as MCR, MACRO-11, or the Task Builder.

There are two types of indirect command files: indirect task command files and indirect MCR command files.

An indirect task command file is a sequential file containing a list of task specific commands. Rather than typing commonly used sequences of commands, you can type the sequence once and store it in a file. The indirect task command file is specified in place of the command line(s) normally submitted to the task.

To initiate indirect task command files, replace the command string for a task with a file specifier, preceded by an at sign (@). The task requesting input then accesses the specified file and starts to read and respond to the commands contained within it. For example, to initiate a file of MACRO-11 commands, input the following:

```
>MAC @INPT.CMD
```

The MACRO-11 Assembler accesses the file and then executes the commands contained within the file INPT.CMD.

An indirect MCR command file contains a list of MCR commands. In addition, an indirect MCR command file can contain special commands that allow you to control command file processing (see Section 5.2).

To initiate an indirect MCR command file, enter the file specifier, preceded by an at sign (@), whenever MCR can accept input. For example:

```
>@MCRIPT.CMD
```

The default file type for indirect command files is .CMD. Thus the above command line could also be input as follows:

```
>@MCRIPT
```

An indirect file can contain any command that can be interpreted by the task to which it is directed.

Some tasks, including MCR, allow nested command files (one command file invokes another). See the appropriate task documentation for the maximum nesting depth allowed.

When the processor reaches the end of the highest level file, it displays the message

```
      >@ <EOF>
```

and then exits.

To exit the indirect command file processor (AT.) prematurely, type CTRL/Z in response to any request for input from the indirect file processor. The message

```
      >@ <EOF>
```

is then displayed.

5.2 MCR INDIRECT FILE PROCESSOR

Most tasks read and respond to commands contained in an indirect file as if the commands were entered directly from a terminal. MCR, however, has an indirect file processor for interpreting commands from a file. An MCR indirect command file can contain both normal MCR commands and special commands (known as directives) to be interpreted by the indirect file processor itself.

When processing an indirect command file, AT. first reads the command file and interprets each command line either as a command to be passed directly to MCR or as a request for action by the indirect command file processor itself. The directives to AT. are distinguished by a period (.) as the first character in the line. MCR commands have no special prefix characters.

The directives to AT. form a procedural language, which allows you to:

1. Define labels
2. Define and assign values to symbols of three types: logical, numeric, and string
3. Create and access data files
4. Control the logical flow within a command file
5. Perform logical tests
6. Enable or disable any of six operating modes
7. Increment or decrement a numeric symbol
8. Control time-based and parallel task execution

Two directives (.BEGIN and .END) permit you to block structure the command file and create Begin-End blocks. Modular, block structured command files are easier to debug and maintain. More importantly, Begin-End blocks isolate local symbol definitions as well as labels and thus conserve symbol table space. When you define a symbol, the indirect command file processor creates an entry in an internal symbol table. These symbol table entries retain their definitions throughout the command file execution if defined locally, or throughout all levels of command files if defined globally. Local symbols defined within a block, however, are defined only within that block; they are erased from the symbol table when AT. encounters an .END directive.

MCR displays on the requesting terminal every MCR command executed. However, if AT. is activated by @filename/-MCR, MCR commands in the indirect file are converted to comments with a leading exclamation point (!); the MCR commands are displayed, but not executed.

An MCR indirect file can also include comments, which the processor prints at the requesting terminal. Comments that begin a line interpreted by MCR have a leading semicolon (;). (The semicolon can be preceded on the same line by a directive to the indirect processor.) Comments that appear after the start of an MCR command have a leading exclamation point (!). All lines in an indirect command file that begin with a period followed by a semicolon (.;) are treated as comments, but are not displayed when the file is processed.

The indirect command file processor attaches the terminal while processing contiguous comment lines (only comments beginning with ;, not !). This permits you to type CTRL/O and suppress a lengthy comment. Output is resumed by typing another CTRL/O or is resumed at the next MCR command in the command file.

The AT. directives described in this chapter are listed below by category. A detailed description of each directive is presented in alphabetical order in Section 5.6.

Label Definition

.label: Assigns a name to a line in the command file so that the line may be referenced.

Symbol Definition

.ASK Defines or redefines a logical symbol and assigns the symbol a logical (true or false) value.

.ASKN Defines or redefines a numeric symbol and assigns the symbol a variable value.

.ASKS Defines or redefines a string symbol and assigns the string symbol a variable ASCII value.

.ERASE Deletes local or global symbol definitions.

.SETT/.SETF Defines or redefines a logical symbol and assigns the symbol a true/false value.

.SETN Defines or redefines a numeric symbol and assigns the symbol a numeric value.

.SETS Defines or redefines a string symbol and assigns the symbol a character string value.

File Access

.DATA	Specifies a single line of data to output to an output data file.
.CHAIN	Closes the current indirect file and uses commands from another file.
.CLOSE	Closes an output data file.
.OPEN	Creates and opens an output data file (if the file exists, creates a new version and opens it).
.OPENA	Opens an existing file and appends subsequent data (does not create a new version).

Logical Control

.BEGIN	Marks the beginning of a Begin-End block.
.END	Marks the end of a Begin-End block.
.EXIT	Terminates processing of either a Begin-End block or the current command file, returns control to the previous level, and optionally sets special symbol <EXSTAT> value.
.GOSUB	Branches to a label identifying a subroutine within the command file.
.GOTO	Branches to a label within the command file.
.ONERR	Branches to a label upon detecting a specific AT. error condition.
.RETURN	Effects an exit from a subroutine and returns to the line immediately following the subroutine call.
.STOP	Terminates indirect command file processing and optionally sets AT. exit status.

Logical Tests

.IF	Determines whether or not a symbol satisfies one of several possible conditions.
.IFACT/.IFNACT	Determines whether or not a task is active.
.IFDF/.IFNDF	Determines whether or not a symbol is defined.
.IFINS/.IFNINS	Determines whether or not a task is installed in the system.
.IFLOA/.IFNLOA	Determines whether or not a device driver is loaded.
.IFT/.IFF	Determines whether a logical symbol is true or false.
.TEST	Tests the length of a string symbol.

Enable/Disable An Operating Mode

.DISABLE — Disables substitution, data, global symbols, lower-case option, escape recognition, or echo control.

.ENABLE — Enables substitution, data, global symbols, lower-case option, escape recognition, or echo control.

Increment/Decrement Numeric Symbols

.DEC — Decrements the value of a numeric symbol by one.

.INC — Increments the value of a numeric symbol by one.

Execution Control

.DELAY — Delays the execution of an indirect command file for a specified period of time.

.PAUSE — Stops the indirect command file processor and allow user action.

.WAIT — Waits for a specified task to complete execution.

.XQT — Initiates a task, pass a command line to it, and continue indirect command file processing without waiting for the task to complete

5.3 SYMBOLS

The MCR indirect file processor enables you to define symbols. These symbols can then be tested or compared in order to control flow through the indirect command file. They may also be substituted for MCR commands, data records for data files, or comments to be displayed on the terminal.

Symbol names are ASCII strings from 1 to 6 characters in length. They must start with a letter (A-Z) or a dollar sign ($). The remaining characters must be alphanumeric or $.

There are three symbol types:

- Logical
- String
- Numeric

A logical symbol has either a true or false value.

A string symbol has as its value a string of ASCII characters, with a length between 0 and 80(10) characters inclusive.

A numeric symbol can have a numeric value in the range of 0-65535(10) inclusive. The symbol can be defined to have either a decimal or octal symbol value. The value (decimal or octal) is relevant only when the symbol is substituted (see Section 5.3.2).

A symbol's type (logical, numeric, or string) is defined by the first assignment directive that assigns a value to the symbol. Subsequent assignments may assign a different value, but cannot redefine the symbol type. Assignment directives can assign:

- A true or false value to define a logical symbol (Defined by .ASK, .SETT or .SETF.)

- An octal or decimal number to define a numeric symbol (Defined by .ASKN or .SETN.)

- A character string to define a string symbol (Defined by .ASKS or .SETS.)

5.3.1 Special Symbols

The MCR indirect command file processor automatically defines special symbols dependent upon specific system characteristics and the replies to queries presented during command file execution. As with symbols, special symbols can be compared, tested, or substituted and have three types: logical, numeric, and string. All special symbols have a common format; angle brackets (<>) enclose the special symbol name.

5.3.1.1 **Special Logical Symbols** - The special logical symbols are assigned a true or false value under the following circumstances.

Symbol	Value
<ESCAPE>	Set to true if last query was answered with a single escape character. Otherwise, <ESCAPE> is set to false.
<DEFAUL>	Set to true if answer to last numeric query (.ASKN) was defaulted (a single carriage return was entered).
<ALPHAN>	Set to true if last string entered in response to an .ASKS directive or tested with a .TEST directive contains only alphanumeric characters. An empty string also sets <ALPHAN> true. Note that a lowercase response to a query when lowercase is disabled sets <ALPHAN> false.
<RAD50>	Set to true if last string entered in response to an .ASKS directive or tested with a .TEST directive contains only Radix-50 characters. Radix-50 characters are the alphanumeric characters plus period (.) and dollar ($). A blank is not a Radix-50 character in this context. An empty string also sets <RAD50> true.
<MAPPED>	Set to true if the system on which AT. is running is mapped, false if the system is unmapped. Always true in an RSX-11M-PLUS system.
<RSX11D>	Always false in an RSX-11M/M-PLUS system.

5.3.1.2 **Special Numeric Symbols** - The following values are assigned to the special numeric symbols.

Symbol Value

<MEMSIZ> Assigned the value of the current system memory size in K words.

<SYUNIT> Assigned the unit number of the user's default device (SY:).

<EXSTAT> Assigned the value of 0, 1, 2, or 4 depending on the exit status from the last MCR command executed or from the last .WAIT tskname directive where tskname was activated by the .XQT directive.

This special numeric symbol is modified at the completion of a synchronous MCR command or at the completion of a .WAIT directive. The .EXIT directive can also modify <EXSTAT>. The value is returned from the task that has completed if the task exits with status. Otherwise, the value is returned from MCR. The values 0, 1, 2, and 4 indicate:

0	Warning
1	Success
2	Error
4	Severe error

<STRLEN> Assigned the length of the string entered in response to the last .ASKS directive or the string tested by the last .TEST directive.

<SPACE> Assigned the number, in octal, of free bytes in the internal symbol table of AT..

<SYSTEM> Assigned an octal number to represent the operating system on which AT. is running. The value 1 is for an RSX-11M system, the value 6 is for an RSX-11M-PLUS system.

5.3.1.3 **Special String Symbols** - The special string symbols are assigned the following string values.

Symbol Value

<CLI> Assigned the acronym (three letters) of the current Command Language Interpreter (DCL or MCR); format is MCR.

<DATE> Assigned the current date; format is dd-mon-yr.

<LIBUIC> Assigned the UIC of the current nonprivileged task library; format is [ggg,mmm]

<SYDISK> Assigned the device mnemonic (two letters) of the user's default device (SY:); format is DK.

Symbol	Value
<SYSUIC>	Assigned the system UIC; format is [ggg,mmm].
<TIME>	Assigned the current time; format is hr:min:sec.
<UIC>	Assigned the current UIC; format is [ggg,mmm]

5.3.2 Symbol Value Substitution

Substitution can occur in any line of a command file. Indirect MCR commands can use the values assigned to logical, numeric, string, or special symbols by replacing a normal parameter (for example, a device-unit) with the symbol name enclosed in apostrophes (for example, 'DEV'). When a previous directive has enabled substitution mode (.ENABLE SUBSTITUTION), AT. replaces the symbol name enclosed in apostrophes with the value assigned to the symbol.

When AT. encounters an apostrophe, it treats the subsequent text, up until a second apostrophe, as a symbol name. AT. then searches the table of symbols for the corresponding symbol, and substitutes the value of the symbol in the indirect command line in place of the symbol name and surrounding apostrophes.

For example, the first three lines below appear in an indirect command file. When the processor executes these lines, it displays the last two lines at the entering terminal.

```
.ASKS DEV MOUNT ON DEVICE?
.ENABLE SUBSTITUTION
MOUNT 'DEV'

>* MOUNT ON DEVICE? [S]: DK2:
>MOUNT DK2:
```

DK2: was entered in response to the displayed question. This reply assigned the string value DK2: to string symbol DEV. Then when the indirect file processor read:

```
MOUNT 'DEV'
```

it substituted for 'DEV' the value assigned to DEV, that is, DK2:. If substitution mode was not enabled, the processor would simply have passed the line to MCR as it appeared in the command file (that is, MOUNT 'DEV').

If substitution mode is enabled, an apostrophe signals the beginning of a string symbol. Thus, to include a single quote as text within a command line, rather than as the start of a symbol, you must replace the single quote with two contiguous apostrophes ('').

If substitution mode is enabled, the processor displays the command file line

```
! DON''T SHOOT
```

as

```
! DON'T SHOOT
```

5.3.3 Numeric And String Symbols

The MCR indirect command file processor assigns a value, either numeric or string, to symbols. This assignment defines the symbol as either a numeric symbol or a string symbol.

5.3.3.1 Numeric Symbols And Expressions - A numeric symbol is a string of digits representing a value in the range of 0-177777 octal (0-65535 decimal if immediately followed by a period). If an arithmetic operation yields a result outside of this range, a fatal error occurs and the message

```
    AT.--NUMERIC UNDER - OR OVERFLOW
```

is displayed.

When a numeric symbol is substituted into a string, the type (octal or decimal) is determined by the value assigned via an .ASKN or .SETN directive.

A numeric symbol or constant may be combined with another numeric symbol or constant by a logical or arithmetic operator to form a numeric expression. Arithmetic operators are used to add (+), subtract (-), multiply (*), and divide (/). Logical operators are the inclusive OR (!) and logical AND (&). No embedded blanks or tabs are permitted between operators.

Numeric expressions are evaluated from left to right unless parentheses are used to form subexpressions that are evaluated first. For example, the directive lines:

```
    .SETN N1 2
    .SETN N2 3
    .SETN N3 N1+N2*4
```

assigns numeric symbol N3 the value 24 octal whereas the directive lines

```
    .SETN N1 2
    .SETN N2 3
    .SETN N3 N1+(N2*4)
```

assigns numeric symbol N3 the value 16 octal.

The type of an expression is octal if all the operands are octal; otherwise the type is decimal.

For example, the directive lines:

```
    .SETN N1 2
    .SETN N2 3
    .SETN N3 N1+N2*4.
```

assigns numeric symbol N3 the decimal value 20.

Numeric expressions are permitted as second operands in numeric .IF and .SETN directives. They are also permitted as range and default arguments in .ASKN and .ASKS directives. The directives .EXIT and .STOP permit numeric expressions to represent exit status.

5.3.3.2 **String Symbols, Substrings, And Expressions** - A string constant is a string of any printable characters enclosed by quotes ("). Empty strings are also permitted. The number of characters cannot exceed 80.

Examples:

```
     "ABCDEF"
        ""
```

String symbols may have a value of any string constant. The value is assigned by a .SETS or .ASKS directive. For example, the directive lines

```
     .SETS    S1    "ABCDEF"
     .SETS    S2     S1
```

assigns string symbol S2 the value of string symbol S1 (that is, ABCDEF).

A substring facilitates the extraction of a segment from the value of a string symbol. You can use substrings only as second operands in .SETS and .IF directives. For example, the directive lines

```
     .SETS    S1    "ABCDEF"
     .SETS    S2     S1[1:3]
```

assign string symbol S2 the value of string symbol S1 beginning at character one and ending at character three (that is, ABC).

You can combine a string constant, symbol, or substring with another string constant, symbol, or substring by the string concatenation operator (+) to form a string expression.

String expressions are permitted as second operands in .SETS and .IF directives where the first operand is a string symbol. For example, the directive lines

```
     .SETS    S1    "A"
     .SETS    S2    "CDEF"
     .SETS    S3    S1+"B"+S2[1:3]
```

assign string symbol S3 the value of the concatenation of string symbol S1, string constant "B", and the first three characters of string symbol S2 (that is, ABCDE).

5.4 SWITCHES

The indirect file processor accepts three switches: /TR, /DE, and /MC.

Switch	Function
/TR	Displays a trace of each processed line on the entering terminal. This function is useful for debugging an indirect command file. Each command line, including those that begin with a period (directive), is displayed. The period on the first directive in the line is changed to an exclamation point (comment) and displayed. If the command causes some action to occur, the next printed line indicates the action; usually, this line consists of the MCR commands issued as a result of the previous directive.
/DE	Indicates that the indirect command file is to be deleted when processing is complete.
/MC	Indicates that commands are to be passed to MCR.

Use any combination of the switches in the command

```
@filename /switches
```

or directive

```
.CHAIN filename /switches
```

To negate the action of the switch, prefix it with a minus sign (-) or "NO" (for example, /NOMC suppresses sending of commands). The defaults are /NOTR, /NODE, and /MC.

The switches you specify in the terminal command that initiates indirect processing are used as defaults when executing the commands filename[/switches] or .CHAIN filename[/switches]. This does not apply, however, for the /DE switch; the default is always /NODE.

5.5 MULTI-LEVEL INDIRECT FILES

You can specify up to four levels of MCR indirect command files. A processor directive (.ENABLE GLOBAL) allows the definition of some symbols as global to all indirect file levels (see Section 5.6.12). Otherwise, each time the processor enters a deeper level, it masks all symbols defined by the previous level out of the symbol table, so that only symbols defined in the current level are available. These symbols are local only to the level of command file in which they are defined. When control returns to a previous level, the symbols defined in that level become available once again and the ones from the lower level are lost.

5.6 DIRECTIVES

Directives must be separated from their arguments and from MCR commands by at least one space.

You can insert any number of blanks and horizontal tabs in three places of a command line:

- At the start of the command line
- Immediately following the colon (:) of a label
- At the end of the command line

This allows you to format the command files for readability. The recommended procedure is to begin labels in column one and everything else in column nine (after one horizontal tab).

An important exception is the lines processed between .ENABLE and .DISABLE DATA directives; no blanks or tabs are removed from these lines. For example:

```
        .IFT Z .GOTO 10
        MAC @ASM
.10:    TKB @BLD
        .OPEN DATFIL
        .DATA XXXXX
.ENABLE DATA
THIS IS DATA
THAT GOES INTO
THE DATA FILE
.DISABLE DATA
        .GOTO 20
```

5.6.1 Define A Label

.label:

Labels always appear at the beginning of the line; they may be on a line with additional directives and/or an MCR command, on a line with a comment, or on a line by themselves. When control passes to a line with a label, the line is processed from the first character after the colon.

Commands do not have to be separated from the label by a space. Only one label is permitted per line. Labels are 1 to 6 characters long, and must be preceded by a period and terminated by a colon. A label may contain only alphanumeric characters and/or dollar sign ($).

It is also possible to define a label as a direct access label; once the label is found, its position in the command file is saved. This allows subsequent jumps to frequently called labels or subroutines to be effected very quickly. The first statement processed after a jump to a direct access label is the one on the next line. You can define up to 20(10) direct access labels within an indirect command file. If you define more than 20, the subsequent direct access labels replace the earliest and so on.

To declare a label as direct access, leave the line following the colon blank.

Examples:

```
.100:    .ASK A DO YOU WANT TO CONTINUE?

.IFT     A     .GOSUB  200
                 .
                 .
                 .

.200:

;THIS IS THE START OF A SUBROUTINE
                 .
                 .
                 .
.RETURN
```

5.6.2 Ask A Question And Wait For A Reply .ASK

The .ASK directive prints a question on the terminal, waits for a reply, and sets a specified logical symbol to the value of true or false, depending on the reply. If the symbol has not already been defined, the processor makes an entry in the symbol table. If the symbol has been defined, the processor resets its value (true or false) in accordance with the reply. The processor exits with a fatal error if the symbol was previously defined as a string or numeric symbol.

Format:

```
.ASK ssssss txt-strng
```

where:

ssssss = 1- to 6-character symbol to be assigned a true/false value.

txt-strng = any ASCII string of characters, preceded by at least one blank. The maximum length of txt-strng is 70.

When executing an .ASK directive, the processor displays txt-strng prefixed by an asterisk and suffixed with "? [Y/N]:". The processor recognizes four answers:

1. Y RET - set symbol ssssss to true.
2. N RET - set symbol ssssss to false.
3. RET - set symbol to false. <CR> indicates carriage return.
4. ESC - set symbol ssssss to true and set the special logical symbol <ESCAPE> to true only if escape recognition has been enabled. ESC indicates escape or altmode.

Example:

The directive line

```
     .ASK INSPIP DO YOU WANT TO INSTALL PIP
```

displays

```
     >* DO YOU WANT TO INSTALL PIP? [Y/N]:
```

on the terminal. Symbol INSPIP will be set to true or false after the user types Y, N, RET, or ESC (if escape recognition is enabled).

5.6.3 Ask For Definition Of A Numeric Symbol .ASKN

The .ASKN directive prints on the terminal a request for a numeric value, waits for it to be entered, optionally tests the range of the numeric response and/or applies a default value, and sets the specified symbol accordingly. If the symbol has not previously been defined, the processor makes an entry in the symbol table. If the symbol has already been defined, the processor resets its value in accordance with the reply. The processor exits with a fatal error if the symbol was previously defined as a logical or string symbol.

Format:

```
     .ASKN ssssss txt-strng
     .ASKN[low:high] ssssss txt-strng
     .ASKN[::def] ssssss txt-strng
     .ASKN[low:high:def] ssssss txt-strng
     (Brackets are required syntax)
```

where:

ssssss = 1- to 6-character symbol to be assigned a numeric value.

txt-strng = any ASCII string of characters, preceded by at least one blank. txt-strng is the text displayed by the indirect file processor. The maximum length of txt-strng is 70.

low:high = the inclusive limits for the response.

def = the default value.

The range and default arguments permit you to combine numeric symbols or constants with another numeric symbol or constant to form a numeric expression (see Section 5.3.3.1).

The command line cannot exceed 80 characters in length. When executing an .ASKN directive, the processor displays txt-strng prefixed by an asterisk and suffixed with [O]: to indicate that the response is considered as octal or [D]: to indicate that the response is considered as decimal. The reply must be a number either within the specified range or in the range 0-177777 octal by default, 0-65535 decimal.

If the response is outside the specified range, the message

```
     AT. -- VALUE NOT IN RANGE
```

is displayed and the query repeated.

If an arithmetic operation yields a result greater than 177777(8) when computing the actual value of any of the arguments low, high, or default, a fatal error occurs and the message

```
AT. -- NUMERIC UNDER- OR OVER-FLOW
```

is displayed.

If the response is an empty line (null) and a default value (def) was not specified, the processor applies a default of 0. Note that in this case, the range, if specified, must include 0.

The response may be either octal or decimal; a leading pound sign (#) forces octal, a trailing period (.) forces decimal. In the absence of both, the processor applies a default type. The default type is decimal if either the range or default values are decimal expressions (followed by a period). Otherwise, the default type is octal. AT. displays the default type as either [O] or [D].

To force a decimal default type without specifying a range argument, use the following construction:

```
.ASKN [::0.] A ENTER VALUE
```

When a numeric symbol is substituted into a string, AT. determines the type (octal or decimal) by the assigned value. The type of the source value is then copied to the destination symbol.

Examples:

The directive line

```
.ASKN  SYM  DEFINE NUMERIC SYMBOL A
```

displays

```
>*DEFINE NUMERIC SYMBOL A [O]:
```

on the entering terminal where:

[O] = the default type (octal).

The processor then defines symbol SYM according to the reply entered.

The directive line

```
.ASKN [2:35:16] NUMSYM DEFINE NUMERIC SYMBOL A
```

displays

```
>* DEFINE NUMERIC SYMBOL A [O R:2-35 D:16]:
```

in the format [x R:low-high D:def]

where:

x = O if the default radix is octal, or D if it is decimal.

R:low-high = the specified range.

D:def = the specified default.

The processor then checks that the response string is within the specified inclusive range.

The directive line

```
     .ASKN [NUMSYM+10:45:NUMSYM+10] SYM DEFINE NUMERIC SYMBOL B
```

displays (assuming the default value of 16 for NUMSYM)

```
     >* DEFINE NUMERIC SYMBOL B [O R:26-45 D:26]:
```

5.6.4 Ask For Definition Of A String Symbol .ASKS

THE .ASKS directive prints on the terminal a request for a string value to define a specified symbol, and optionally tests that the number of characters in the response string falls within the specified inclusive range. If the symbol has not previously been defined, the processor makes an entry in the symbol table. If the symbol has already been defined, the processor resets its value in accordance with the reply. The processor exits with a fatal error if the symbol was defined previously as a logical or numeric symbol. If the number of characters is out of the specified range, the message

```
     AT. -- STRING LENGTH NOT IN RANGE
```

is displayed and the query repeated.

Formats:

```
     .ASKS ssssss txt-strng
     .ASKS [low:high] ssssss txt-strng
     (Brackets are required syntax)
```

where:

ssssss = 1- to 6-character symbol to be assigned a string value.

txt-strng = any ASCII string of characters, preceded by at least one blank. txt-strng is the text displayed by the indirect file processor. The maximum length of txt-strng is 70.

low:high = the inclusive limits for the number of characters permitted in the response string.

The command line cannot exceed 80 characters in length.

When specifying range arguments, you may combine numeric symbols or constants with another numeric symbol or constant to form a numeric expression (see Section 5.3.3.1).

When executing an .ASKS directive, the processor displays txt-strng prefixed by an asterisk (*) and suffixes it with [S]:. The reply must be an ASCII character string.

Examples:

The directive line

```
.ASKS NAM PLEASE ENTER YOUR NAME
```

displays

```
>* PLEASE ENTER YOUR NAME [S]:
```

on the entering terminal. The processor then defines symbol NAM according to the string reply entered.

The directive line

```
.ASKS [1:15] MIDNAM PLEASE ENTER YOUR MIDDLE NAME
```

displays

```
>* PLEASE ENTER YOUR MIDDLE NAME [S R:1-15]:
```

in the format [S R:low-high]

where:

S = the symbol type (string).

R:low-high = the specified range for number of characters.

5.6.5 Begin Block .BEGIN

The .BEGIN directive marks the beginning of a Begin-End block. All local symbols following the directive are local to the block instead of the entire command file. An .ERASE LOCAL directive (see Section 5.6.14) erases all local symbols within the block.

Begin-End blocks can be nested up to a maximum depth of 127. AT. usually exhausts stack space before this limit can be reached; you may wish to rebuild AT. and specify more task stack space if an application requires the maximum nesting level.

Format:

```
.BEGIN
```

.BEGIN must be the only directive on a line. The .BEGIN directive cannot appear, for example, on the same line as an .IF directive.

Begin-End blocks isolate local symbol definitions and labels. All local symbols defined inside a block lose definition outside the block. However, all symbols defined outside a block retain definition throughout the command file. Note that symbols defined outside a block and then modified within a block assume and retain the value assigned in the block.

The block must be terminated by an .END directive.

5.6.6 Continue Processing Using Another File .CHAIN

The .CHAIN directive closes the current indirect file, masks all local symbols, and continues processing using commands fetched from another file. However, the .CHAIN directive does not close data files or change the file indirection level.

Format:

```
     .CHAIN filename [/switches]
```

where filename is the name of the file that contains the new commands and /switches are any of the optional switches described in Section 5.4.

Example:

The directive

```
     .CHAIN OUTPUT
```

transfers control to the file OUTPUT.CMD.

5.6.7 Close Secondary File .CLOSE

The .CLOSE directive closes the secondary file opened by an .OPEN directive.

Format:

```
     .CLOSE [#n]
     (Brackets not part of syntax.)
```

where:

#n = optional file number in the range 0-3 inclusive. Default is #0. You can substitute a numeric symbol for the value n by enclosing the symbol in apostrophes.

5.6.8 Output Data To Secondary File .DATA

The .DATA directive specifies text that is to be output to a secondary file previously opened by an .OPEN directive.

Format:

```
     .DATA [#N] text-string
     (Brackets not part of syntax.)
```

where:

text-string = text to be output to the secondary file.

#N = optional file number in the range 0-3 inclusive. Default is 0. You can substitute a numeric symbol for the value N by including single quotes.

The command line cannot exceed 80 characters. (The specified text cannot continue onto the next line.) If a secondary file is not open, an error condition exists; the indirect file processor issues an error message and begins error processing.

Example:

```
.SETS SEND "THIS IS DATA"
.OPEN TEMP
.DATA 'SEND'
.CLOSE
```

These directives output THIS IS DATA to the secondary file TEMP.DAT.

5.6.9 Decrement Numeric Symbol .DEC

The .DEC directive decrements a numeric symbol by 1. The processor exits with a fatal error if the symbol was defined previously as a logical or string symbol.

Format:

```
.DEC ssssss
```

where:

ssssss = 1- to 6-character numeric symbol.

Example:

```
.DEC N
```

This directive decrements by 1 the value assigned to the numeric symbol N.

5.6.10 Delay Execution For A Specified Period Of Time .DELAY

The .DELAY directive delays further processing of the indirect file for a specified period of time.

Format:

```
.DELAY nnu
```

where:

```
nn = Number of time units to delay.
 u = T - ticks
     S - seconds
     M - minutes
     H - hours
```

The parameter nn is octal by default, or decimal if terminated with a period. For example:

```
10S is 10(8) seconds
10.S is 10(10) seconds
```

When the DELAY directive is executed, the processor issues the message

```
AT. -- DELAYING
```

When the time period expires and the task resumes, the processor issues the message

```
AT. -- CONTINUING
```

Example:

The directive

```
.DELAY 20M
```

delays processing for 20(8), or 16(10) minutes.

5.6.11 Disable Option .DISABLE

The .DISABLE directive disables a specified operating mode (substitution, data, global symbol, lower-case, escape recognition, or echo control) previously activated by an .ENABLE directive. See Enable Option (Section 5.6.12) for operating mode information.

Formats:

```
.DISABLE SUBSTITUTION

.DISABLE DATA [#n]
 (Brackets not part of syntax.)
```

where:

#n = optional file number in the range 0-3. Default is 0. You can substitute a numeric symbol for the value n by enclosing the symbol in apostrophes.

```
.DISABLE GLOBAL

.DISABLE LOWERCASE

.DISABLE ESCAPE

.DISABLE QUIET
```

5.6.12 Enable Option .ENABLE

The .ENABLE directive is used to invoke one of six operating modes: lower to upper-case character conversion; data mode; global symbol mode; substitution mode; recognition of an escape character; or echo control. Each mode is independent of the others; all six can be active simultaneously. When the indirect file processor starts to process a command file, the initial settings are:

Lower case	enabled
Data	disabled
Global	disabled
Substitution	disabled
Escape	disabled
Quiet	disabled

In lower-case mode, characters read from the terminal in response to .ASKS directives are stored in the string symbol without lower- to upper-case conversion. The representation of characters is significant when comparing strings (see Section 5.6.18) since the .IF directive treats lower-case and upper-case characters as distinct. Also, note that if lower-case mode is disabled and the response to a query is in lower-case, the special logical symbol <ALPHAN> is set to false.

In data mode, the processor outputs lines that follow the directive line .ENABLE DATA to a secondary file. (The .DATA directive sends a single line of text to a secondary file; see Section 5.6.8.)

In global symbol mode, symbol names that begin with a dollar sign ($) are defined as global to all levels of indirect files; once such a symbol has been defined, all levels recognize it. Symbols that do not begin with a dollar sign are always recognized only by the level that defines them.

In substitution mode, the processor substitutes a string in place of a symbol. The symbol must begin and end in apostrophes ('SYMBOL'). For example, if the symbol A has been assigned the string value THIS IS A TEST, then every A will be replaced by THIS IS A TEST. When substitution mode is active, the processor performs substitutions on each line before scanning the line for directives and MCR commands. (While obeying a .GOTO label directive, however, the processor ignores any undefined symbols encountered before the target line, that is, the line containing the specified label.)

Escape recognition permits the response to an .ASK, .ASKN, or .ASKS directive to be an escape character. A question answered with a single escape character sets the special logical symbol <ESCAPE> to true. The escape character must be used only as an immediate terminator to the question; if one or more characters precede the escape, an error condition exists. In this case, the message

```
AT.--INVALID ANSWER OR TERMINATOR
```

is printed and the query repeated. Note that if you type ESC in response to an .ASK directive, the specified logical symbol (ssssss of .ASK ssssss txt-strng) is also set to true.

Quiet mode is valid only in systems that support parent-offspring tasking; the directive performs no function in systems that do not support parent-offspring tasking. In quiet mode, the processor does not echo Command Line Interpreter (CLI) command lines. The command lines are executed normally and, if they return a message or display, the message or display is printed on the terminal.

Formats:

```
.ENABLE LOWERCASE

.ENABLE SUBSTITUTION

.ENABLE DATA [#n]
```

(Brackets not part of syntax.)

where:

#n = optional file number in the range 0-3. Default is 0. You can substitute a numeric symbol for the value n by enclosing the symbol in apostrophes.

```
.ENABLE GLOBAL

.ENABLE ESCAPE

.ENABLE QUIET
```

Examples:

- Substitution mode -

  ```
  .ENABLE SUBSTITUTION
  .ASKS FIL SPECIFY SOURCE FILE
  MAC  'FIL'='FIL'
  ```

  When the indirect file is executing, the corresponding lines displayed at the terminal are:

  ```
  * SPECIFY SOURCE FILE [S]:SOURCE
  >MAC SOURCE=SOURCE
  ```

- Lower-case mode -

  ```
  .ENABLE SUBSTITUTION
  .ENABLE LOWERCASE
  .ASKS A DEFINE STRING SYMBOL A
  ;'A'
  ```

  When the file is executing, the corresponding lines displayed at the terminal are:

  ```
  >* DEFINE STRING SYMBOL A [S]:RSX is in caps
  >;RSX is in caps
  ```

- Global symbol mode -

  The following two lines appear in an indirect command file BOB1:

  ```
  .ENABLE GLOBAL
  .SETS $X "TEST"
  ```

  A file called BOB2.CMD contains the following lines:

  ```
  .ENABLE SUBSTITUTION
  @BOB1
  MAC '$X','$X'='$X'
  ```

  Therefore, MCR displays the following when the file BOB2.CMD is run:

  ```
  >@BOB1
  >MAC TEST,TEST=TEST
  >@ <EOF>
  ```

- Data mode

  ```
  .OPEN SECFILE.DAT
  .ENABLE DATA
        .
        .
        .
  .DISABLE DATA
  ```

  The lines that fall between the .ENABLE and .DISABLE directives are written to the secondary file SECFILE.DAT.

- Escape recognition mode -

```
;IF YOU WANT A LIST OF OPTIONS, TYPE <ESC>
.ENABLE ESCAPE
.ASKS A ENTER OPTION
.IFT <ESCAPE> .GOTO LIST
        .
        .
        .
.LIST: ;OPTIONS ARE: A (ADD), S (SUBTRACT), M (MULTIPLY)
```

If the user types the escape key (which is not displayed) in response to ENTER OPTION, the corresponding lines displayed at the terminal are:

```
>;IF YOU WANT A LIST OF OPTIONS, TYPE <ESC>
>* ENTER OPTION [S]: ESC
>;OPTIONS ARE: A (ADD), S (SUBTRACT), M (MULTIPLY)
```

- Quiet mode -

```
.ASK QUIET DO YOU WANT COMMAND LINES SUPPRESSED
.IFT QUIET .ENABLE QUIET
.IFF QUIET .DISABLE QUIET
ACT
```

If the response is affirmative, the processor displays the active tasks, but not the Active command. For example:

```
>DO YOU WANT COMMAND LINES SUPPRESSED? [Y/N]:Y
MCR...
ACTT17
AT.T17
```

5.6.13 End Block .END

The .END directive marks the end of the Begin-End block. If AT. encounters more .END directives than .BEGIN directives, command file processing terminates and displays the following message:

```
AT. -- ILLEGAL NESTING
```

Format:

```
.END
```

as the only directive on a line.

5.6.14 Delete Symbols .ERASE

The .ERASE directive deletes all local or global symbol definitions. When you define a symbol, either locally (by defining a symbol value) or globally (by enabling global symbol mode and preceding the symbol name with a dollar sign ($)), AT. creates an entry in a symbol table. The .ERASE directive erases either all local or all global entries in the table.

Following an .ERASE directive, you can redefine symbol values as well as symbol type.

Formats:

```
.ERASE LOCAL
.ERASE GLOBAL
```

An .ERASE LOCAL directive outside of a Begin-End block erases all local symbols. An .ERASE LOCAL directive within a Begin-End block erases only those local symbols defined in the block. An .ERASE GLOBAL, either outside of or within a Begin-End block, erases all global symbols.

Example:

```
.ERASE LOCAL
```

This directive erases all local symbol definitions for the indirect command file.

5.6.15 **Exit Current Command File** .EXIT

The .EXIT directive terminates processing of the current command file or Begin-End block and returns control to the previous level command file or, if the directive is executed within a block, to the line following an .END directive. If the directive is encountered at the uppermost indirect nesting level, AT. exits and passes control to MCR (see the .STOP directive, Section 5.6.29).

The directive also allows you to optionally specify a value to copy into the special symbol <EXSTAT>.

Format:

```
.EXIT [value]
(Brackets  not part of syntax.)
```

where value is an optional numeric expression to be copied to the special symbol <EXSTAT>.

Examples:

The following line appears in an indirect command file BOB1:

```
@BOB2
```

The file BOB2.CMD contains the following line:

```
.EXIT
```

When AT. encounters the .EXIT directive in BOB2, control returns to BOB1.CMD.

If the .EXIT directive in BOB2.CMD includes a numeric expression, for example:

```
.EXIT N+2
```

AT. evaluates the expression and copies the value into <EXSTAT>.

5.6.16 Call A Subroutine .GOSUB

The .GOSUB directive saves the current position in an indirect file and then branches to a label. The label identifies an entry point to a subroutine which is terminated by a .RETURN directive.

When you issue a .GOSUB directive from within a Begin-End block, AT. saves the current block context and then scans down the file searching for the first occurrence of the subroutine label. Note that during the scan, AT. ignores any intervening .BEGIN or .END directives. The .RETURN directive restores previous block context. Thus, the subroutine can be contained within a Begin-End block.

The maximum nesting depth for subroutine calls is eight.

Format:

```
     .GOSUB label
```

where label is the label that designates the first line of a subroutine, but without the leading period and trailing colon. The label must be preceded by at least one space.

Example:

The directive

```
     .GOSUB EVAL
```

transfers control to the subroutine labeled .EVAL:.

5.6.17 Branch To A Label .GOTO

The .GOTO directive causes a branch from one line in an indirect file to another. All commands between the .GOTO directive and the specified label are ignored. Branches can go forward or backward in the file.

When executing a .GOTO directive, AT. ignores all lines within a nested Begin-End block. Thus, the target label of a .GOTO directive cannot be contained within a block (unlike the target label of a .GOSUB directive). Likewise, the target of a .GOTO branch from within a Begin-End block must be contained in that block; the .GOTO directive cannot branch into or out of a block. Note, however, that AT. permits branches across nested blocks. When AT. encounters a .GOTO directive within a Begin-End block, it scans for the label within that block. Since the label scan starts at the .BEGIN directive and continues to the .END directive, multiply defined labels are permitted within a block.

Format:

```
     .GOTO label
```

where label is the name of the label, but without the leading period and trailing colon. The label must be preceded by at least one space.

Example:

The directive

```
     .GOTO 100
```

transfers control to the line containing the label .100:.

5.6.18 Logical Test IF

A number of directives make tests; if the test is true, AT. processes the remainder of the command line. Logical tests can be combined into a compound logical test by using the .AND and .OR directives.

5.6.18.1 Test If Symbol Meets Specified Condition

.IF - The .IF directive compares a numeric or string symbol with another expression of the same type to determine if one of several possible conditions is true. If the condition is satisfied, the processor executes the remainder of the command line. When comparing a string symbol with a string expression, the indirect processor compares the ASCII values of each operand's characters (from left to right) one by one. An operand is considered greater if the first non-equal character has a greater value than the corresponding character in the other operand. Numeric symbols are compared strictly on the basis of magnitude.

Format:

```
.IF symbol relop expr
```

where:

symbol = 1- to 6-character name of numeric or string symbol.

relop = one of the following relational operators:

EQ or =	- equal to
NE or <>	- not equal to
GE or >=	- greater than or equal to
LE or <=	- less than or equal to
GT or >	- greater than
LT or <	- less than

expr = expression of same type as symbol.

Examples:

```
.SETS X "A"
.SETS Y "a"
.IF X LT Y .GOTO 200
```

The ASCII value of string symbol X is less than the ASCII value of string symbol Y, which satisfies the less than condition. Thus, control passes to the line containing the label .200:.

```
.SETN N1 2
.SETN N2 7
.IF N1 <= N2 PIP /LI
```

With the condition satisfied (numeric symbol N1 less than or equal to numeric symbol N2), the PIP command is processed.

```
.SETS S1 "AAb"
.SETS S2 "AA"
.SETS S3 "BBBB"
.IF S1 >= S2+S3[1:1] .INC N
```

Since the condition where string symbol S1 is greater than or equal to string symbol S2 concatenated with the first character of string symbol S3 (AAb >= AAB) is satisfied, the processor increments numeric symbol N.

5.6.18.2 **Test If Task Is Active Or Dormant** .IFACT/.IFNACT - The .IFACT or .IFNACT directive tests whether a task is active (.IFACT) or dormant (.IFNACT). If the test is true, the rest of the command is processed. If the specified task is not installed, the processor assumes the dormant condition.

Formats:

```
     .IFACT  tttttt

     .IFNACT  tttttt

     where:

          tttttt = a task name.
```

The task name tttttt must be preceded and followed by at least one blank.

Examples:

```
     .IFACT   REPORT  .GOTO 350

     .IFNACT  REPORT  RUN REPORT
```

5.6.18.3 **Test If Symbol Is Defined Or Not Defined** .IFDF/IFNDF - The .IFDF or .IFNDF directive tests whether a logical, numeric or string symbol has been defined (.IFDF) or not defined (.IFNDF). If the test is true, the rest of the command line is processed. This directive does not test the value of the symbol.

Formats:

```
     .IFDF  ssssss

     .IFNDF  ssssss

     where:

          ssssss = 1- to 6-character symbol being tested.
```

At least one blank must precede and follow the symbol.

Examples:

```
     .IFDF  A  .GOTO 100

     .IFNDF  A  .ASK A DO YOU WANT TO SET TIME
```

5.6.18.4 **Test If Task Is Installed Or Not Installed** .IFINS/.IFNINS - The .IFINS or .IFNINS directive tests whether a task is installed (.IFINS) or not installed (.IFNINS) in the system. If the test is true, the rest of the command line is processed.

Formats:

```
.IFINS  tttttt

.IFNINS  tttttt
```

where:

tttttt = a task name.

The task name tttttt must be preceded and followed by at least one blank.

Examples:

```
.IFINS  PIP  .GOTO 250

.IFNINS  PIP  INS [1,50]PIP
```

5.6.18.5 **Test If Driver Is Loaded Or Not Loaded** .IFLOA/.IFNLOA - The .IFLOA or .IFNLOA directive tests whether a driver is loaded (.IFLOA) or not loaded (.IFNLOA) in the system. If the test is true, the rest of the command line is processed. Note that for purposes of this directive, resident drivers are assumed to be loaded.

Formats:

```
.IFLOA dd:

.IFNLOA dd:
```

where:

dd: = a device driver.

Examples:

```
.IFLOA DK:  .GOTO 250

.IFNLOA DK:  LOA DK:
```

5.6.18.6 **Test If Symbol Is True Or False** .IFT/.IFF - The .IFT or .IFF directive tests whether a logical symbol is true or false. If the test is true, AT. processes the remainder of the directive line.

The processor exits with a fatal error if the symbol being tested was previously defined as a numeric or string symbol.

Formats:

```
.IFT  ssssss

.IFF  ssssss
```

where:

ssssss = 1- to 6-character logical symbol being tested.

At least one blank must precede and follow the logical symbol.

Examples:

```
.IFT  A  .GOTO  100

.IFF  B  .GOTO  200
```

5.6.18.7 **Compound Tests** - You can combine "If" tests by using the .AND and .OR directives. In addition, an implied .AND is effected when more than one IF appears on the same line without being separated by an .AND directive. The compound operators .AND and .OR must be preceded and followed by at least one blank.

The .AND directive takes precedence over the .OR directive as shown in the following example:

```
.IFT A .OR .IFT B .AND .IFT C .GOTO D
```

That is, the processor reads the line as:

```
.IFT A .OR (.IFT B .AND .IFT C) .GOTO D
```

Examples:

```
.IFT A .AND .IFF B  .GOTO HELP
```

If the logical symbol A is true and the logical symbol B is false, control passes to the line containing the label .HELP:.

```
.IFT A .IFF B  .GOTO HELP
```

Same effect as the previous directive (.AND implied).

```
.IFT A .OR  .IFF B RUN PIP
```

If the logical symbol A is true or if the logical symbol B is false, the MCR Run command is issued.

5.6.19 Increment Numeric Symbol

.INC

The .INC directive increments a numeric symbol by 1. The processor exits with a fatal error if the symbol was previously defined as a logical or string symbol.

Format:

```
.INC ssssss
```

where:

ssssss = 1- to 6-character numeric symbol.

Example:

```
.INC N
```

Increment by 1 the value assigned to the numeric symbol N.

5.6.20 Define Logical End Of File /

The logical end of file directive (/) terminates file processing and exits. The message

```
>@<EOF>
```

is then displayed.

Format:

```
/
```

as the first non-blank character on a line.

You can use this directive at any location in the command file to quickly terminate file processing, but care should be taken to avoid an inadvertent exit.

Example:

```
                  .ASK CONT DO YOU WISH TO CONTINUE
                  .IFT CONT .GOTO 100
                  /
      .100:
```

5.6.21 Branch To Label On Detecting An Error .ONERR

If the processor detects one of the following errors:

- Task not installed in system (.XQT, .WAIT, .IFACT, .IFNACT)
- Undefined symbol
- Bad syntax (.XQT, .WAIT, .DELAY)
- Unrecognized command
- String substitution error (any command)
- Symbol type error (.IF, .IFT, .IFF, .INC, .DEC)
- Redefinition of a symbol to a different type (.ASK, .ASKN, .ASKS, .SETT, .SETF, .SETN, .SETS)
- Data file error (.OPEN, .OPENA, .DATA, .CLOSE, between .ENABLE DATA and .DISABLE DATA)

control passes to the line containing the specified label. This feature provides you with a means of gaining control in order to terminate command file processing in an orderly manner.

Note that the .ONERR directive applies only to the error conditions listed above; errors returned from a task external to AT. (for example, an MCR syntax error) are not processed by the .ONERR directive.

Format:

```
.ONERR  label
```

Upon detecting an error, the processor passes control to the line starting with .label:. The .ONERR directive must be issued before AT. encounters the error condition. If the directive is executed (one of the above errors is encountered), error processing passes to the specified label. Subsequent errors pass control to the same label until a new .ONERR directive is executed. If the label specified by the .ONERR directive does not exist and an error condition has occurred, indirect command file processing terminates.

Example:

```
.ONERR  100
```

Upon detecting one of the above error conditions, the indirect command file processor passes control to the line labeled .100:.

5.6.22 Open Secondary File .OPEN

The .OPEN directive opens a specified secondary file as an output file.

Format:

```
.OPEN [#n] filename
(Brackets not part of syntax.)
```

where:

filename = a file to be opened as an output file. Default file type is .DAT.

#n = optional file number in the range 0-3 inclusive, allowing up to four data files to be open simultaneously. Default is #0. You can substitute a numeric symbol for the value n by enclosing the symbol in apostrophes.

Example:

```
.OPEN SECOUT
```

This directive opens the file SECOUT.DAT as an output file.

5.6.23 Open Secondary File For Append .OPENA

The .OPENA directive opens a secondary file and appends all subsequent data to the file.

Format:

```
.OPENA [#n] filename
(Brackets not part of syntax)
```

where:

filename = A secondary file to be opened with subsequent data appended to it. Default file type is .DAT.

#n = optional file number in the range 0-3 inclusive, allowing up to four data files to be open for append simultaneously. Default is #0. You can substitute a numeric symbol for the value n by enclosing the symbol in apostrophes.

Example:

```
.OPENA SECOUT
```

This directive opens the file SECOUT.DAT as an output file and appends subsequent data to it.

5.6.24 Pause For Operator Action .PAUSE

The .PAUSE directive interrupts processing of an indirect file to wait for user action. A .PAUSE directive causes the indirect file processor task to stop itself. The user may then perform some operations and subsequently cause the task to resume.

Format:

```
.PAUSE
```

When the indirect file processor task stops itself, it displays the following message on the entering terminal:

```
AT. -- PAUSING.  TO CONTINUE TYPE "UNS tttttt"
```

where:

tttttt = the name of the indirect file processor task.

The operator then types the following command to resume the task:

```
>UNS  tttttt
```

The indirect file processor then displays the message:

```
AT. -- CONTINUING
```

and it continues processing where it left off.

5.6.25 Return From A Subroutine .RETURN

The .RETURN directive signifies the end of a subroutine and returns control to the most recently saved position in the indirect file (the line immediately following the .GOSUB directive that initiated the subroutine).

Format:

```
     .RETURN
```

5.6.26 Set Symbol To True Or False .SETT/.SETF

The .SETT/.SETF directives define or change the value of a specified logical symbol. If the symbol has not been defined, the processor makes an entry in the symbol table and sets the logical symbol to the value specified. If the symbol has already been defined, the processor resets the symbol accordingly. The processor exits with a fatal error if the logical symbol was defined previously as a numeric or string symbol.

Formats:

```
     .SETT  ssssss

     .SETF  ssssss
```

where:

ssssss = 1- to 6-character logical symbol to be assigned a true or false value.

The logical symbol ssssss must be preceded by at least one blank.

Examples:

```
     .SETT  X
```

This directive sets the logical symbol X to true.

```
     .SETF  ABCDE
```

This directive sets the logical symbol ABCDE to false.

5.6.27 Set Symbol To Numeric Value .SETN

The .SETN directive defines or changes the numeric value of a specified symbol. If the symbol has not been defined, the processor makes an entry in the symbol table and sets the symbol to the numeric value specified. If the symbol has already been defined, the processor resets the symbol accordingly. The processor exits with a fatal error if the numeric symbol was previously defined as a logical or string symbol.

Format:

```
     .SETN ssssss numexp

     where:

          ssssss =   1- to 6-character numeric symbol.

          numexp =   a numeric expression.  (See Section 5.3.3.1.)
```

When specifying a numeric value to assign to a symbol, you may combine a numeric symbol or constant with another numeric symbol or constant to form a numeric expression. If numeric expressions are used, no embedded blanks or tabs are permitted. Evaluation is done from left to right unless parentheses are used to form subexpressions that are evaluated first. The type of an expression is octal if all the operands are octal; otherwise the type is decimal.

Examples:

```
     .SETN NUMBER 27
```

This directive assigns to the numeric symbol NUMBER the value 27 (octal).

```
     .SETN A1 3*(A2-5)
```

This directive assigns the numeric symbol A1 the value of symbol A2 minus 5 multiplied by 3.

5.6.28 Set Symbol To String Value .SETS

The .SETS directive defines or changes the string value of a specified string symbol. If the symbol has not been defined, the processor makes an entry in the symbol table and sets the symbol to the specified string value. If the symbol has been defined, the processor resets the symbol accordingly. The processor exits with a fatal error if the symbol was defined previously as a logical or numeric symbol.

Format:

```
     .SETS ssssss strexp

     where:

          ssssss  = 1- to 6-character string symbol.

          strexp  = any string expression.  (See Section 5.3.3.2.)
```

The processor assigns to the specified symbol the string value represented by the string expression strexp. If a string constant is used in strexp, the constant must be enclosed by quotes ("strexp").

You can combine a string symbol, constant, or substring with another string symbol or substring by the string concatenation operator (+) to form a string expression.

Examples:

```
     .SETS A "ABCDEF"
```

This directive assigns to the string symbol A the string value ABCDEF.

```
     .SETS STR2 "ZZZ"
```

This directive assigns string symbol STR2 the value ZZZ.

```
     .SETS X STR2+"ABC"
```

This directive assigns string symbol X the value of symbol STR2 plus ABC (that is ZZZABC).

```
     .SETS X STR2+A[1:3]
```

This directive is equivalent to the previous directive; it assigns the string symbol X the string value of STR2 plus the first three characters of string A (that is ZZZABC).

```
     .SETS MYFILE <UIC>+"MYFILE.TXT"
```

This directive assigns the string symbol MYFILE the string value of the current UIC plus the string MYFILE.TXT (for example, if the current UIC is [303,3], string symbol MYFILE is assigned the string value [303,3]MYFILE.TXT).

5.6.29 Terminate Command File Processing .STOP

The .STOP directive immediately terminates command file processing and exits. The message

```
     >@ <EOF>
```

is then displayed.

The .STOP directive allows you to optionally set the exit status for AT. execution.

Format:

```
     .STOP [value]
```

where value is an optional numeric expression to serve as the exit status for the indirect command file processor. If you do not specify an exit status value, the .STOP directive is identical to the logical end of file directive (/).

Example:

```
     .STOP 0
```

This directive terminates command file processing and sets AT. exit status to zero.

5.6.30 Test String Symbol .TEST

The .TEST directive moves the length of a string symbol into the special numeric symbol <STRLEN>. It also tests the characters of the string and sets the special logical symbols <ALPHAN> and <RAD50> accordingly (see Section 5.3.1).

Format:

```
     .TEST strsym
```

where strsym is the string symbol to be tested.

Example:

The directive

```
     .TEST A
```

moves the number of characters in string A into <STRLEN> and sets <ALPHAN> and <RAD50> accordingly. The special numeric symbol <STRLEN> is then available to compare the length of string symbol A to some numeric constant or expression.

5.6.31 Wait For A Task To Finish Execution .WAIT

The .WAIT directive suspends processing of the indirect command file until a particular task has terminated.

Format:

```
     .WAIT  tttttt
```

The task name tttttt must be preceded by at least one blank.

If the taskname is omitted, the indirect processor assumes the taskname applied by the last previous "RUN filename" MCR command. This name is specified as

```
     TTnn
```

where:

TT = the invoking terminal.

nn = the terminal number.

If the specified (or default) task is not installed, the indirect processor ignores the .WAIT directive. The .WAIT directive performs no function if the /NOMCR switch is in effect.

Example:

```
     .WAIT  PIP
```

This directive discontinues processing of the command file until the terminal-initiated task PIP exits.

5.6.32 Initiate Parallel Task Execution .XQT

The indirect file processor usually passes a command to MCR and waits until the command's execution has completed. However, it is possible for the processor to initiate a task and not wait for it to complete before executing the next indirect file command. The MCR Run command initiates tasks, and the indirect file processor can continue as soon as the Run command is processed. However, you cannot use the Run command to pass a command string to the target task. The .XQT directive allows you to start a task, to pass a command string to it, and to continue processing command lines in parallel with the initiated task. AT. permits up to 10(10) successive .XQT directives.

Format:

```
.XQT  tsk-name command-string
```

where:

tsk-name = the name of the task (for example, MAC, PIP, DMO).

command-string = the command to the task.

The .XQT command provides a facility to initiate parallel processing of tasks. Use the .WAIT directive to synchronize the execution of parallel tasks; command file processing is suspended until the specified task completes.

Example:

```
.XQT  MAC TEST,TEST=TEST
.XQT  TKB BLD,BLD=BLD
.WAIT MAC
.WAIT TKB
```

The example starts an assembly and a task build executing in parallel and then waits for the two tasks to complete.

5.7 TASK NAME REFERENCES

References to task names in an indirect command file follow the same rules as for MCR. If the task was started as an external MCR task (for example, MAC, PIP, DMO), it may be referenced by its 3-character name. Thus, the .WAIT, .IFINS, and .IFACT directives need only specify the 3-character task name; the indirect command file processor can then find the correct task. Of course, you can always refer to a specific task by using its full 6-character name.

5.8 EXAMPLE COMMAND FILE AND ITS EXECUTION

```
;
;       EXAMP.CMD
;
        .ENABLE SUBSTITUTION
        .ENABLE GLOBAL
        .;-----------------------------------------------------
        .;
        .; This procedure calculates the day of the week from the
        .; date obtained from the system.
        .;
        .;-----------------------------------------------------
        .;
        .; DAYS OF THE WEEK.
        .;
        .SETS DAYS "SUN MON TUESWED THURFRI SAT SUN"
        .;
        .; MONTHS OF THE YEAR.
        .;
        .SETS MONTHS "JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC"
        .;
        .; NUMBERS FOR DAYS OF THE MONTH.
        .;
.SETS DAYMON "01020304050607080910111213141516171819202122232425262728293031"
        .;
        .; NUMBERS FOR THE YEARS.
        .;
        .SETS YEARNM "798081828384858687888990919293949596979899"
        .SETS DAT "'<DATE>'"                        ! GET TODAYS DATE.
        .SETS TIM "'<TIME>'"                        !GET THE TIME.
        .SETS DAY DAT[1:2]                          ! EXTRACT THE DAY
        .SETS MONTH DAT[4:6]                        ! EXTRACT THE MONTH
        .SETS YEAR DAT8.:9.]                        ! EXTRACT THE YEAR
        .SETN DAYN 1.                               ! DAY OF MONTH COUNTER.
        .SETN NDX 1.                                ! DAY OF MONTH INDEX.
.DLOOP:
        .SETS TEMP DAYMON[NDX:NDX+1]                ! SET TO DAY OF MONTH
        .IF TEMP = DAY .GOTO 10                     ! BRANCH IF CORRECT DAY
        .INC DAYN                                   ! BUMP COUNTER
        .SETN NDX NDX+2.                            ! POINT TO NEXT DAY
        .GOTO DLOOP                                 ! TRY NEXT DAY
.10:    .SETN MONCNT 1.                             ! INIT MONTH COUNTER
        .SETN NDX 1.                                ! INIT MONTH INDEX
.MLOOP:
        .SETS TEMP MONTHS[NDX:NDX+2.]               ! SET TEMP TO MONTH
        .IF TEMP = MONTH .GOTO 20                   ! BR IF CORRECT MONTH
        .INC MONCNT                                 ! BUMP MONTH COUNTER
        .SETN NDX NDX+3.                            ! POINT AT NEXT MONTH
        .GOTO MLOOP                                 ! TRY NEXT MONTH
.20:    .SETN YEARN 79.                             ! INIT YEAR COUNTER.
        .SETN NDX 1.                                ! INITIALIZE YEAR INDEX.
.YLOOP:
        .SETS TEMP YEARNM[NDX:NDX+1]                ! SET TEMP TO MONTH
        .IF TEMP = YEAR .GOTO 30                    ! BRANCH IF CORRECT YEAR
        .INC YEARN                                  ! BUMP YEAR COUNTER
        .SETN NDX NDX+2.                            ! POINT AT NEXT YEAR
        .GOTO YLOOP                                 ! TRY NEXT YEAR
        .;
        .;
        .;
.30:    .SETN T1 MONCNT+10.
        .SETN T2 T1/13.
.;      .SETN T3 (13*(T1-(T2*12.))-1.)/5.
```

```
         .SETN T3 (T1-(T2*12.))
         .SETN T3 ((13.*T3)-1)/5.
         .SETN T3 T3+DAYN+77.
.;       .SETN T4 YEARN+((MONCNT-14.)/12.)
         .SETN T4 YEARN
         .IF MONCNT < 3 .DEC T4
         .SETN T5 T4/100.
         .SETN T6 T4/400.
.;       .SETN T7 (5.*(T4-(T5*100.)))/4.
         .SETN T7 (T4-(T5*100.))
         .SETN T7 (5.*T7)/4.
         .SETN T8 T3+T7+T6-T5-T5+1.
         .SETN DAYNDX T8-((T8/7.)*7.)              ! DAY INDEX (SUNDAY = 0)
         .SETN DAYNDX (DAYNDX*4.)+1.               ! INDEX INTO DAYS LIST
         .SETS $DAY DAYS[DAYNDX:DAYNDX+3.]         ! GET DAY OF THE WEEK
         ;
         ; The date is - 'DAT',
         ; The time is - 'TIM', and
         ; Today is '$DAY'DAY.
         ;
         ; See you tomorrow.
         ;
```

COMMAND FILE EXECUTION

```
>@EXAMP
>;
>;     EXAMP.CMD
>;
>;
>; The date is - 17-APR-79,
>; The time is - 13:25:37, and
>; Today is TUESDAY.
>;
>; See you tomorrow.
>;
>@ <EOF>
>
```

DOESN'T PRINT DAY NAME OUT CORRECTLY.

APPENDIX A

MCR AND TKTN MESSAGES

Messages from MCR and TKTN appear in this chapter in alphabetical order. Some messages include remedial action to be taken by the operator. This is not included with messages for which the required action is obvious or contained in the description of the message.

11 40 F.P. EXCEPTION

Explanation: This is a TKTN message. The task encountered a floating point exception while executing on a PDP-11/40, and no SST routine was specified to handle the trap.

ABORTED BY DIRECTIVE OR MCR

Explanation: Either MCR or an Executive directive issued by another task caused the task to be aborted.

ABORTED VIA MCR

Explanation: MCR aborted the task and requested a post-mortem dump.

XXX -- ACCESS TO COMMON BLOCK DENIED

Explanation: Common block protection requirements were not met.

XXX -- ACCOUNT FILE OPEN FAILURE

Explanation: The account file was open for another user; or the disk containing the account file was not mounted.

User Action: Retry command.

XXX -- ACCOUNT FILE RECORD(S) LOCKED

Explanation: The account file or a specific record was locked. Retry the command.

ACCOUNTING SEC. POOL ALLOCATION FAILURE

Explanation: This is an RSX-11M-PLUS TKTN message. The resource accounting system could not find sufficient space in secondary pool to allocate a block.

XXX -- ACP NOT IN SYSTEM

Explanation: The specified Ancillary Control Processor was not installed in the system.

XXX -- ADDRESSING EXTENSIONS NOT SUPPORTED

Explanation: The command tried to install a task with a VSECT (virtual section) into a system that does not support the feature.

XXX -- ALIGNMENT ERROR

Explanation: An attempt was made to create a partition, but the base address or size conflicted with existing partitions or physical memory size.

XXX -- ALLOCATION FOR SYS FILE EXCEEDS VOLUME LIMIT

Explanation: The system was unable to allocate the system file from the specified block because intermediate bad blocks or end-of-volume was encountered.

XXX -- ALREADY MARKED FOR DISMOUNT

Explanation: The device-unit has already been requested for dismount and the associated ACP was waiting for all accesses to the volume to complete.

XXX -- ALREADY MOUNTED

Explanation: The specified network device was already mounted.

XXX -- AMBIGUOUS QUALIFIER

Explanation: The qualifier does not specify a unique part of the Help file.

User Action: Further information must be supplied.

XXX -- AMBIGUOUS SYNONYM

Explanation: A qualifier in the Help file references an ambiguous shared entry.

XXX -- ASSIGN FAILURE

Explanation: The Hello command could not complete login logical assignments. For example, the MCR dispatcher is not installed.

AST ABORT. BAD STACK

Explanation: This is a TKTN message. An AST cannot be effected because the AST parameters cannot be pushed onto the task's stack.

XXX -- BAD BLOCK FILE CORRUPT - DATA IGNORED

Explanation: The bad block file contained bad data.

XXX -- BAD BLOCK FILE FULL

Explanation: The disk had more then 102 bad regions on it.

XXX -- BAD BLOCK HEADER I/O ERROR

Explanation: A write error was detected in writing the bad-block file header.

XXX -- BAD COMMAND OR SYNTAX

Explanation: The directive was unknown or the syntax of the directive was incorrect.

XXX -- BAD TASK FILE VBN, TASK REMOVED - taskname

Explanation: The system removed the specified task because it contained a bad Virtual Block Number (VBN) in its file header.

XXX -- BASE ADDRESS MUST BE ON 4K BOUNDARY

Explanation: The base address of the task is not on a 4K boundary. This message is applicable only to mapped systems.

XXX -- BASE MISMATCH COMMON BLOCK <common-name>

Explanation: The base address of the common block, as recorded in the task image, did not match the base address of the resident common block.

XXX -- BINARY OPERATOR ERROR

Explanation: The .AND or .OR indirect-file-processor directive was not used correctly.

XXX -- BLOCK(S) EXCEED VOLUME LIMIT

Explanation: The specified block (or blocks) exceeded the physical size of the volume.

XXX -- BOOT BLOCK WRITE ERROR

Explanation: An error was detected in writing the volume boot block.

XXX -- BOOTED DEVICE CANNOT BE BROUGHT ONLINE

Explanation: An RSX-11M-PLUS message. There was an error when Save attempted to bring either the unit or controller of the booted device online. The Executive is corrupted, the device driver does not work, or the device does not work.

XXX -- BOOTED DEVICE'S DRIVER NOT LOADED

Explanation: VMR has been used to unload the driver for the system (booted) disk.

XXX -- BOOTED DEVICE NOT IN SYSTEM -- dd nnn mmmmmm

Explanation: When the system was booted, Save could not find the booted device (device name dd, physical unit number nnn, CSR mmmmmm) in the system data structures. For example, the system was booted from DB3:, but the system data structures include only DB0: and DB1:.

XXX -- BYTE ADDRESS

Explanation: The address specified as the argument to the Open command was an odd address. For example, the address 3000 is a legal address; however, the address 3001 is not.

XXX -- CANNOT INSTALL PRIVILEGED TASK FROM NONPRIVILEGED TERMINAL

Explanation: This message applies to multiuser protection systems only. A nonprivileged user attempted to install a privileged task. Only a privileged user can install a privileged task.

XXX -- CANNOT LOAD/UNLOAD A PSEUDO DEVICE

Explanation: The device you attempted to load is a pseudo device.

XXX -- CAN'T READ MCR COMMAND BUFFER

Explanation: UFD was started by a Run command rather than by an MCR command; therefore, the command could not be parsed.

XXX -- CHECKING ddnn:

Explanation: Automatic bad-block specification is proceeding, using the bad-block file produced by the Bad Block Locator utility program.

XXX -- CHECKPOINT AREA TOO SMALL

Explanation: The area allocated for checkpointing the task was smaller than the partition into which the task was being installed. This message applies only to systems that do not support the dynamic allocation of checkpoint space.

CHECKPOINT FAILURE. READ ERROR

Explanation: This is a TKTN message. The task could not be read back into memory after being checkpointed.

XXX -- CHECKPOINT FILE ALREADY IN USE ON DEVICE

Explanation: A previous ACS command established a checkpoint file on the volume mounted on the specified device-unit. A volume can contain only one checkpoint file.

XXX -- CHECKPOINT FILE HEADER I/O ERROR

Explanation: An error was detected in writing out the checkpoint file header.

XXX -- CHECKPOINT FILE NOW INACTIVE

Explanation: This message appears after an ACS command has been issued and the file does not contain any checkpointed tasks. The message indicates that use of the file was discontinued immediately after the command was issued. If the file were still in use, TKTN would issue a message when the file was finally discontinued.

*** ddnn: -- CHECKPOINT FILE NOW INACTIVE

Explanation: The message, issued by TKTN, indicates that a request to discontinue use of a checkpoint file on device ddnn: has finally been satisfied. The need to return tasks checkpointed to the discontinued file back into memory caused the delay between the request and this message.

XXX -- CHECKPOINT FILE STILL ACTIVE

Explanation: The command attempted to dismount a volume that contained an active checkpoint file. The volume cannot be dismounted until the checkpoint file has been discontinued.

User Action: Issue an ACS command to discontinue the file and reissue the DMO command when the system issues a message to indicate that the checkpoint file is inactive.

XXX -- CHECKPOINT FILE STILL IN USE ON dev:

Explanation: The system cannot be saved because a checkpoint file on the specified device is still active.

User Action: Deallocate the checkpoint file (see the ACS command) and reissue the Save command.

XXX -- CHECKPOINT SPACE TOO SMALL, USING CHECKPOINT FILE

Explanation: A warning message. The checkpoint space allocated in the task image file is too small for the size of the task (usually because of the /INC keyword). The system supports the dynamic allocation of checkpoint space and will therefore use a checkpoint file to contain the rolled-out task.

*** ddnn: CHECKPOINT WRITE ERROR

Explanation: This is a TKTN message. A write failure occurred while the system was attempting to checkpoint a task.

XXX -- CIRCULAR REDIRECT ERROR

Explanation: An attempt was made to redirect a device, which would have caused a circular list of redirects to result.

XXX -- CO: REDIRECTED TO CO:

Explanation: The console output driver is active.

User Action: If it is not active, use the Open command to redirect CO: to the terminal from which you are running.

XXX -- COMMAND INPUT ERROR

Explanation: Circumstances prevented the BROADCAST task from receiving the command line (usually an indirect file could not be found); or, a system directive or the Run command, rather than the Hello command, has initiated the HELLO task.

User Action: In the case of BROADCAST, retry the command.

XXX -- COMMAND I/O ERROR

Explanation: An I/O error was generated during a read to an indirect file, or an attempt by another task to obtain a command from MCR failed.

XXX -- COMMAND SYNTAX ERROR

Explanation: The command line had an improper format.

XXX -- COMMAND TOO LONG

Explanation: An INITVOL or Mount command, including continuation lines, exceeded the maximum length of 512. characters.

XXX -- COMMON BLOCK IS TASK PARTITION common-name

Explanation: A task's request for access to a common block was rejected because the requested partition was a task partition.

XXX -- COMMON BLOCK NOT LOADED common-name

Explanation: The specified common block was linked to the task but had not been loaded into the system.

User Action: Install the specified common block, then install the task.

XXX -- COMMON BLOCK OCCUPIED

Explanation: An attempt was made to load a common block that was already occupied.

XXX -- COMMON BLOCK PARAMETER MISMATCH common-name

Explanation: Parameters (partition name and PIC attributes) of a common block did not match those in the task's label block.

XXX -- COMMON BLOCK SEGMENT COUNT ERROR

Explanation: The command installed a resident library with overlays, but the number of overlay segments specified in the task header is different from the number specified in the segment descriptors.

XXX -- COMMON common-name NOT INSTALLED FROM AN LB:

Explanation: An RSX-11M-PLUS message. All checkpointable commons must be installed from an LB:.

User Action: Remove the common and retry the Save command.

XXX -- COMMON, DRIVER OR TASK ABOVE SYSTEM IMAGE FILE LIMIT

Explanation: A system cannot be saved if a memory resident structure is higher than the amount of main memory that is written to the system image file.

User Action: Use the Partitions command to determine the cause of the problem.

XXX -- COMMON NOT IN SYSTEM

Explanation: The specified or required common does not exist in the current system.

XXX -- CONTINUING

Explanation: This is not an error message. This message indicates that the indirect file processor is resuming execution after a .PAUSE or .DELAY directive.

XXX -- CTB name DOES NOT EXIST

Explanation: The controller table name specified with the /CTB keyword does not exist in the resident database.

XXX -- CTB NAME name IS A DUPLICATE

Explanation: The loadable database contains a Controller Table (CTB) with a name that is the same as a CTB name in the resident database. CTB names must be unique.

XXX -- CTB name IS NOT SUPPORTED BY DRIVER-NOT LOADED

Explanation: The device driver's dispatch table does not have all the Controller Table mnemonics that the rest of the database (the DCBs and CTBs) implies it to have or the CTB is defective. A defective CTB may contain a wrong name, or its linkage to the DCB may be wrong.

XXX -- DCB TABLE FOR CTB name IS FULL

Explanation: Load attempted to write the address of your Device Control Block (DCB) for the loadable database into the DCB table of the Controller Table. There were no null entries in the DCB table. (Refer to the RSX-11M or RSX-11M-PLUS Guide to Writing an I/O Driver.)

XXX -- DATA ERROR

Explanation: The specified bad-block number or the specified contiguous region size is too large.

User Action: Reenter properly.

XXX -- DATA FILE ERROR x.

Explanation: The MCR indirect command file processor encountered an error while processing an .OPEN, .CLOSE, .DATA directive or a data mode access to the secondary file. See Appendix I in the IAS/RSX-11 I/O Operations Reference Manual for a definition of the numeric code x.

XXX -- DELAYING

Explanation: This is not an error message; it indicates that a .DELAY directive has just been executed by the indirect file processor.

XXX -- DEVICE ATTACHED

Explanation: The specified device was currently attached by a task and could not be mounted; or the specified device cannot be allocated because it is attached to a running task.

XXX -- DEVICE ATTACHED[-dev:]

Explanation: The device-unit specified in the command was attached by a task and could not be mounted. For attempts to mount one or more magnetic tapes, the message includes a specific device-unit.

XXX -- DEVICE ddn: HAS UNITS ATTACHED, BUSY, AND/OR MOUNTED

Explanation: Device ddn: is the first or only unit that is attached, mounted, and/or has outstanding I/O. Therefore, the driver cannot be unloaded.

XXX -- DEVICE NOT ALLOCATED TO THIS TERMINAL

Explanation: The command line specified a private device that was allocated to other than the issuing terminal.

XXX -- DEVICE NOT DEFINED IN NETWORK

Explanation: The specified device had not been defined in the network topology and therefore could not be mounted.

XXX -- DEVICE NOT IN SYSTEM

Explanation: The device specified in the command was not generated into the system. Devices to be used in the system must be specified during system generation.

XXX -- DEVICE NOT IN SYSTEM -dev:

Explanation: The Mount command specified a tape drive that has not been generated into the system.

XXX -- DEVICE ddn: NOT IN SYSTEM

Explanation: The data base for the device driver was neither in the system nor in the driver's task image; or specified device driver does not exist in the system.

XXX -- DEVICE NOT MOUNTED

Explanation: The device specified in the file specification was not mounted. In response to a Load or Unload command, the system device (SY:) was not mounted.

XXX -- DEVICE NOT REDIRECTABLE

Explanation: The specified device may not be redirected.

XXX -- DEVICE NOT READY - ddnn:

Explanation: The command specified a volume that was not ready (not up to speed).

XXX -- DEVICE NOT TERMINAL

Explanation: An attempt was made to set terminal characteristics for a non-terminal device.

XXX -- DEVICE NOT VARIABLE SPEED MULTIPLEXER

Explanation: An attempt was made to set the baud rate for a terminal that was not attached to a DH11 or DZ11 multiplexer.

XXX -- DEVICE OFFLINE[-dev:]

Explanation: The device specified in the command, although generated into the system, was not physically present in the host configuration. If the offline device is a magnetic tape drive, the message includes the device-unit.

XXX -- DEVICE SPECIFIED TWICE -- dev:

Explanation: The Mount command specified the same device-unit twice.

XXX -- DEVICE WRITE LOCKED - ddnn:

Explanation: The command specified a volume that was write-locked.

XXX -- DIRECTIVE PARTITION UNFIXED OR NONEXISTENT

Explanation: VMR has been used to make the contents of the directive partition disappear. Use VMR to restore it.

XXX -- DIRECTORY ALREADY EXISTS

Explanation: The requested UFD already exists on the volume.

XXX -- DISK IS ALIGNMENT CARTRIDGE

Explanation: The last track on an RK06 disk identified the volume as an alignment cartridge, which cannot be initialized as a Files-11 volume. An alignment cartridge is specifically formatted for aligning disk read/write heads.

*** ddnn: DISMOUNT COMPLETE

Explanation: This is a TKTN message. This message indicates that the device requested for dismount is now logically disconnected from the system (that is, all files are deaccessed and the VCB is deallocated).

XXX -- DPB ERROR

Explanation: A bad DPB was created by MCR. This error indicates that the system itself has faulted.

User Action: If the error persists, submit a Software Performance Report (SPR) to DIGITAL.

XXX -- DRIVE LOAD/UNLOAD CALLS NOT SUPPORTED

Explanation: The requested driver included the symbols $ddLOA or $ddUNL in its source. These symbols are reserved for future use by the Load and Unload commands.

XXX -- DRIVER ALREADY RESIDENT

Explanation: The specified device driver had already been loaded.

XXX -- DRIVER BEING LOADED OR UNLOADED

Explanation: The Executive was effecting another request to load the driver.

XXX -- DRIVER BUILT WITH WRONG EXECUTIVE STB FILE

Explanation: The STB file for the driver was not compatible with the current Executive.

XXX -- DRIVER CANNOT BE UNLOADED

Explanation: The Unload command attempted to remove a permanently resident device driver (that is, a driver linked to the Executive).

XXX -- DRIVER DISPATCH TABLE IS INCONSISTENT

Explanation: The device driver's dispatch table is illegally formatted. Refer to the <u>RSX-11M-PLUS Guide to Writing an I/O Driver</u>.

XXX -- DRIVER LOAD/UNLOAD CALLS NOT SUPPORTED

Explanation: The requested driver included the symbols $ddLOA or $ddUNL in its source. These symbols are reserved for future use by the Load and Unload commands.

XXX -- DRIVER NOT LOADED

Explanation: The device driver specified in the Unload command was not resident in memory; or the Mount command specified a device for which a driver was not loaded.

XXX -- DUPLICATE BLOCK(S) FOUND

Explanation: A block that had been specified as bad was being defined as bad a second time.

XXX -- DYNAMIC ALLOCATION OF CHECKPOINT SPACE NOT SUPPORTED

Explanation: The ACS command was issued on a system that did not include support of dynamic allocation of checkpoint space at system generation.

XXX -- ERROR LOGGING STILL ACTIVE

Explanation: A system cannot be saved while error logging is active. Run the task ERF to disable error logging.

XXX -- FAILED TO ATTACH DEVICE - ddnn:

Explanation: The specified device could not be attached.

XXX -- FAILED TO CREATE A DIRECTORY

Explanation: No space existed on the volume, or an I/O error occurred.

XXX -- FAILED TO ENTER IN MFD

Explanation: No space existed in the Master File Directory (MFD) or on the volume, or an I/O error occurred on the volume.

XXX -- FAILED TO READ BAD BLOCK FILE

Explanation: The INITVOL command was unable to read the bad block information from the last track of an RK06 disk.

XXX -- FATAL I/O ERROR

Explanation: An input or output operation could not be completed because of the unavailability of a device or dynamic memory, or a device error occurred.

XXX -- FEATURE NOT SUPPORTED

Explanation: The command keyword specifies an optional feature that was not incorporated into the system at system generation.

XXX -- FILE name CONTAINS INVALID DRIVER DATA BASE

Explanation: The driver had an invalid data base.

XXX -- FILE name HAS ILLEGAL STB FORMAT

Explanation: The driver's STB file contained illegal object code or data.

XXX -- filename FILE HEADER I/O ERROR

Explanation: An error was detected in writing out the file header for the specified system file.

XXX -- FILE HEADER READ ERROR

Explanation: Mount could not read either the index file or the storage allocation file.

User Action: Determine the problem; if it is a hardware error, contact the local Field Service office.

XXX -- FILE I/O ERROR nnn.

Explanation: The ACS command detected an error when it tried to allocate or open the checkpoint file. The code nnn. is an FCS error code that defines the cause of the error. See Appendix I of the IAS/RSX-11 I/O Operations Reference Manual.

XXX -- FILE filename HAS INCONSISTENT DRIVER LOAD INFORMATION

Explanation: The partition for loadable drivers is logically beyond the end of the system image file.

XXX -- FILE name NOT A VALID DRIVER TASK IMAGE

Explanation: The driver's task image was either overlaid or had a task header.

XXX -- FILE NOT CONTIGUOUS

Explanation: An attempt was made to boot a system or install a task from a noncontiguous file. System images and task images must be contiguous.

XXX -- FILE name NOT CONTIGUOUS

Explanation: The file specified was not contiguous. The file must be contiguous.

XXX -- FILE NOT FOUND

Explanation: The requested file was not in the directory on the specified volume.

XXX -- FILE NOT IN USE

Explanation: A checkpoint file was not in use on the device specified in the ACS command.

XXX -- FILE NOT TASK IMAGE

Explanation: Data in the label block was not correct, indicating that the file was not a valid task image.

XXX -- FILE READ ERROR

Explanation: An error was detected in reading an indirect file. This error is usually caused by records that are more then 80. bytes long.

XXX -- HOME BLOCK ALLOCATE WRITE ERROR

Explanation: A write error occurred while the system was overwriting a bad home block area.

XXX -- HOME BLOCK CHECKSUM ERROR

Explanation: The checksum in the home block and the calculated checksum did not agree. This condition was probably caused by an I/O error.

XXX -- HOME BLOCK I/O ERROR

Explanation: An error was detected in reading or writing the volume home block.

XXX -- HOME BLOCK READ ERROR

Explanation: An error was detected in reading the home block. This error usually indicates that the volume is not ready in the drive. Wait until it is ready and reissue the command.

XXX -- HOME BLOCK WRITE ERROR

Explanation: A write error was detected in writing the volume home block.

XXX -- IIST CANNOT BE BROUGHT ONLINE

Explanation: There was an error when Save attempted to place the IIST controller online. The Executive is corrupted, the driver does not work, or the IIST does not work. There is no recovery procedure.

XXX -- ILLEGAL DEVICE ddnn:

Explanation: The device specified at task-build time by the Assign command keyword option (Devices) did not exist in the system.

XXX -- ILLEGAL DEVICE SPECIFIED

Explanation: Specified device was not a terminal.

XXX -- ILLEGAL DEVICE/VOLUME

Explanation: The specified device was not a valid task-residence device.

XXX -- ILLEGAL DRIVER TASK APR USAGE

Explanation: A device driver must be built for APR 5 and must be less than 4K (e.g., PAR=DRVPAR:120000:20000). This message applies to mapped systems only.

XXX -- ILLEGAL FIRST APR

Explanation: A privileged task, built to run using APR 4 or 5 as its base, was too large. Using APR 4, the task was larger than 12K; using APR 5, the task was larger than 8K. If the Executive has been built to support 20K of address space, APR 5 is the only valid base APR for tasks mapping into the Executive. A privileged task that does not map into the Executive must use APR 0 as its base. (See the RSX-11M/M-PLUS Task Builder Manual for a description of the /PR keyword.) This message applies only to mapped systems.

XXX -- ILLEGAL FUNCTION

Explanation: A command was entered that MCR could not recognize.

XXX -- ILLEGAL KEYWORD COMBINATION

Explanation: The command specified conflicting keywords.

XXX -- ILLEGAL KEYWORD VALUE

Explanation: A value entered for a keyword exceeded its limits.

User Action: Enter a legal value.

XXX -- ILLEGAL PRIORITY

Explanation: The value of the priority in the command was out of range (that is, not 1 to 250(10), inclusive).

XXX -- ILLEGAL UIC

Explanation: The command line specified an illegal UIC.

XXX -- ILLEGAL VALUE FOR SYMBOL symname IN FILE name

Explanation: The symbol symname is defined to an illegal value by file ddDRV.STB. This can occur for several reasons; some general and some specific to individual symbols. For example, symbols cannot have odd values; most symbols cannot be defined as zero; $ddTBL must exist and have a value less then $ddEND; mapped loadable drivers cannot reference $INTSV.

XXX -- IMPROPER HELP FILE FORMAT

Explanation: The Help file is not in proper format and cannot be used. Either edit the file or re-enter it in proper format.

XXX -- INDEX FILE BIT MAP I/O ERROR

Explanation: An error was detected in writing out the index file bit map.

XXX -- INDEX FILE HEADER CHECKSUM ERROR

Explanation: While rebooting a saved system, the system detected a checksum error in the file header of the index file. The error causes the system to halt.

XXX -- INDEX FILE HEADER READ ERROR

Explanation: While rebooting a saved system, the system detected an error in the file header of the index file. The error causes the system to halt.

XXX -- INDEX FILE I/O ERROR

Explanation: An I/O error was detected in writing out the index-file header.

XXX -- INPUT I/O ERROR

Explanation: In attempting to read the next command, the Open command detected an error.

XXX -- INSUFFICIENT ICB POOL SPACE FOR CPU CPx

Explanation: There is not enough pool space to allocate an Interrupt Control Block for the CPU designated in the error message. This message applies only to Multiprocessor, I and D space systems.

XXX -- INSUFFICIENT POOL SPACE

Explanation: The required amount of pool space was not available.

XXX -- INTERRUPT VECTOR ALREADY IN USE

Explanation: A loadable device driver interrupt does not point to one of the nonsense interrupt entries. This message can appear when a user issues the Load command.

XXX -- INVALID ACCOUNT

Explanation: The name or UIC specified in the command is not stored in the account file; or the password specified does not match the name or UIC given.

XXX -- INVALID ADDRESS

Explanation: The address specified as an argument in the Open command referenced a nonexistent memory location or an address outside the boundaries of the specified partition.

XXX -- INVALID ANSWER

Explanation: In response to a question from the indirect file processor directive .ASK, the operator entered a response other than Y, N, or null, followed by carriage return. The processor will repeat the question.

XXX -- INVALID DRIVER DATA BASE AT OFFSET xxxx IN FILE name

Explanation: The driver had an invalid data base value at offset xxxx relative to the symbol $ddDAT.

XXX -- INVALID KEYWORD

Explanation: A keyword was encountered that was not recognized by the specific command processor.

XXX -- INVALID INTERRUPT VECTOR

Explanation: The interrupt vector address specified in the driver data base was too high.

XXX -- INVALID LOAD DEVICE

Explanation: The Boot or Install command detected a device that was invalid as a system or task-residence device.

XXX -- INVALID SPEED

Explanation: The multiplexer line specified does not support the requested speed; or the command specified unequal receive and transmit speeds for a DZ11. The DZ11 does not support split speeds.

XXX -- INVALID TIME PARAMETER

Explanation: A time field was incorrect.

XXX -- INVALID UIC

Explanation: A value of zero was detected for either the group number or the member number.

IOT EXECUTION

Explanation: This is a TKTN message. The subject task executed an IOT instruction, and no SST routine was specified to process the trap.

XXX -- I/O ERROR SIZING DEVICE - ddnn:

Explanation: The system encountered an I/O error while sizing an RF11 disk.

XXX -- I/O OUTSTANDING

Explanation: Tasks that have I/O requests waiting to complete remain in the system. A system cannot be saved with I/O requests outstanding.

User Action: Wait for the I/O requests to complete and reenter the command.

XXX -- KRB name NOT IN LOADABLE DATA BASE

Explanation: You are attempting to load the driver for a MASSBUS device with a loadable data base. The special symbol (KRBname) which defines the location of the KRB is not defined in the loadable data base. Refer to RSX-11M-PLUS Guide to Writing an I/O Driver.

XXX -- KRB TABLE OF CTB name WILL NOT ACCEPT KRB name

Explanation: You are attempting to load the driver for a MASSBUS device with a loadable data base. If this operation is to be successful, two conditions must exist:

1. The appropriate slot in the CTB (controller table) must exist.

2. The slot in the CTB must be unused.

When you get this error message, one of these conditions does not exist. Refer to the RSX-11M-PLUS Guide to Writing an I/O Driver.

XXX -- LABEL BLOCK I/O ERROR

Explanation: In saving the system image, the Save command writes the transfer address in the label block of the system image file. An error occurred during this write attempt.

XXX -- LABEL BLOCK READ ERROR

Explanation: The Boot command could not read the label block of the system image.

XXX -- LENGTH MISMATCH COMMON BLOCK <common-name>

Explanation: The length parameter for the common block, as described in the label block for the task image, did not match the corresponding length parameter defined in the system. A task's label-block data must match system data for that task before it can be installed.

XXX -- LINE NOT DZ11

Explanation: The command attempted to set to remote a line that was not attached to a DZ11 multiplexer.

XXX -- LISTING DEVICE NOT AVAILABLE

Explanation: The device-unit on which MCR was attempting to display information was attached and unavailable for use.

XXX -- LOAD DEVICE NOT LB:, TASK REMOVED - taskname

Explanation: The system removed the specified task while saving a system because the task had not been installed from LB:.

LOAD FAILURE. READ ERROR

Explanation: This is a TKTN message. The requested task could not be loaded because of a hardware error.

XXX -- LOADABLE DRIVER SUPPORT NOT IN SYSTEM

Explanation: The system generation option to support loadable device drivers was not chosen.

XXX -- LOGICAL DEVICE NOT IN SYSTEM

Explanation: The specified logical device was not defined in the logical device tables.

XXX -- LOGINS ARE DISABLED

Explanation: The system was in the process of shutting down; or the command SET /NOLOGON has been issued. A user cannot log onto a terminal at that time.

XXX -- LUN OUT OF RANGE

Explanation: An attempt was made to reassign a LUN that had a value greater than the maximum number of LUNs allocated for the task at task build.

XXX -- MAGTAPE LABEL MUST BE SPECIFIED

Explanation: An INITVOL command attempted to initialize a magnetic tape without specifying the required volume label.

XXX -- MAGTAPE DEVICE ERROR - ddnn:

Explanation: While positioning a magnetic tape before initializing it, the system encountered an I/O error.

XXX -- MAGTAPE WRITE ERROR - ddnn:

Explanation: While writing to the magnetic tape during initialization, the system encountered an I/O error.

XXX -- MAXIMUM INDIRECT FILES EXCEEDED

Explanation: An attempt was made to access an indirect file at a depth greater than four levels.. MCR indirect files may nest to a maximum depth of four levels.

XXX -- MCR IS NOT INSTALLED

Explanation: When a system is booted, Save uses MCR to mount the system disk. MCR... must be installed using VMR.

MEMORY PARITY ERROR IN PARTITION pname

Explanation: This is a TKTN message. The error logging subsystem detected a memory parity error in the specified partition. TKTN prints this message as notification of such.

MEMORY PROTECT VIOLATION

Explanation: This is a TKTN message. The task encountered a memory protect violation, and no SST routine was specified to process the trap.

XXX -- MESSAGE FILE ERROR nnn.

Explanation: The system could not open the file LOGIN.TXT for a reason indicated by the FCS code nnn. See Appendix I of the <u>IAS/RSX-11 I/O Operations Reference Manual</u> for a definition of the FCS code.

XXX -- MFD FILE HEADER I/O ERROR

Explanation: An error was detected in writing out the Master File Directory (MFD).

XXX -- MFD WRITE ERROR

Explanation: An error was detected in writing out the master file directory.

MICRO CODE LOADER NOT INSTALLED

Explanation: For systems that include the KMC11 line printer driver (LKDRV), this message indicates that the task LPINIT is not installed in the system.

XXX -- MOUNT ERROR FROM ACP - xxx.

Explanation: The ACP detected an error while trying to mount the volume set. See the IAS/RSX-11 I/O Operations Reference Manual for a definition of standard FCS error codes.

XXX -- MULTI-UNIT DEVICE NOT TAPE

Explanation: The command specified multiple units that are not tape drives.

XXX -- MUST BE RUN FROM TERMINAL CO:

Explanation: Save is not being run from the terminal to which the pseudo device CO: is directed. This is not a requirement on an RSX-11M-PLUS system, therefore, the message does not apply.

XXX -- NEW DEVICE NOT KNOWN TO SYSTEM

Explanation: The new device in the Redirect command was not known to the system (that is, it does not exist in the device tables).

XXX -- NO BAD BLOCK DATA FOUND

Explanation: Although automatic bad-block specification was selected, no bad-block file could be found on the volume. This is a warning message.

XXX -- NO CHECKPOINT SPACE, ASSUMING NOT CHECKPOINTABLE

Explanation: This is a warning message. The command installed a task that was designated checkpointable at task build, but that had no checkpoint space allocated in its task image file. Because the system does not support the dynamic allocation of checkpoint space, it considers the task not to be checkpointable.

XXX -- NO LUNS

Explanation: A LUN command specified a task that had no current LUN assignments.

XXX -- NO POOL SPACE

Explanation: The Executive could not currently satisfy MCR's request for dynamic memory.

XXX -- NO ROOM AVAILABLE IN STD FOR NEW TASK

Explanation: No dynamic storage was available for making an entry in the System Task Directory (STD); therefore, the specified task could not be installed.

XXX -- NO TRANSFER ADDRESS

Explanation: The Boot command could not find a transfer address in a virgin system image (result of task build).

XXX -- NO VOLUME LIST

Explanation: The command specified a magnetic tape drive for which a mounted volume list does not exist.

XXX -- NON EXISTENT MEMORY

Explanation: An attempt was made to define a partition in nonexistent memory.

NON RSX EMT EXECUTION

Explanation: This is a TKTN message. The task executed an EMT instruction having an argument other than 377(8), and no SST routine was specified to process the trap.

XXX -- NOT ENOUGH APRS FOR TASK IMAGE

Explanation: The Task Builder allows the user to specify the virtual base address of a task image as a multiple of 4K. Privileged tasks start at virtual 100000(8) to map the first 16K of the Executive, or virtual 120000(8) to map the first 20K of the Executive, at the same time as the user task. If the virtual base address is set too high, the task image may not be able to be mapped with the remaining mapping registers. This message is applicable to mapped systems only.

XXX -- NOT FILE STRUCTURED DEVICE

Explanation: The system does not support Files-11 on the specified device.

XXX -- NOT FILES-11 DEVICES

Explanation: The command specified a device that did not contain a Files-11 disk. The specified device-unit must contain a mounted Files-11 disk.

XXX -- NOT MOUNTABLE DEVICE

Explanation: The specified device was not supported as a Files-11 device (including ANSI magnetic tape) or a network device.

XXX -- NOT MOUNTED

Explanation: The specified device was not mounted.

***   dnn: NOT READY

Explanation: This is a TKTN message. The specified device was not ready.

XXX -- NOT SYSTEM IMAGE

Explanation: The Boot command determined that the file was not a system image.

XXX -- NOT VALID SAVE DEVICE

Explanation: The device specified in the Save command was not valid. The Save command can specify only random access devices in Files-11 format.

XXX -- NT DEVICE NOT MOUNTED

Explanation: The device NT: was not mounted. This device must be mounted before any other network devices.

XXX -- NULL FILE HEADER I/O ERROR

Explanation: An error was detected in writing out null-file headers to the index file.

ODD ADDRESS OR OTHER TRAP FOUR

Explanation: This is a TKTN message. The task executed a word instruction with an odd address or referenced a nonexistent memory location in an unmapped system, and no SST routine was specified to process the trap.

XXX -- OLD DEVICE ATTACHED

Explanation: An attempt was made to redirect an attached device-unit.

XXX -- OLD DEVICE MOUNTED

Explanation: An attempt was made to redirect a mounted device-unit.

XXX -- OLD DEVICE NOT KNOWN TO SYSTEM

Explanation: An attempt was made to redirect an unknown device-unit.

XXX -- OPEN FILE(S) ON ddnn:

Explanation: There are open files on ddnn:. Insure that the queue manager is stopped.

XXX -- OTHER USER LOGGED ON

Explanation: The issuing terminal was currently logged by another user. Only one user at a time can be logged onto a terminal.

XXX -- OTHER VOLUME MOUNTED[-dev:]

Explanation: An attempt was made to mount a volume on a device that already had a mounted volume. The message specifies the device-unit if it is a tape drive.

XXX -- OUTPUT ERROR

Explanation: An MCR terminal-write operation failed.

XXX -- OVERLAP SEEKS NOT SUPPORTED

Explanation: The command attempted to enable overlapped seek support for other than a DM, DP, or RH device controller.

XXX -- PARAMETER CONFLICTS WITH MOUNTED VOLUME

Explanation: An attempt was made to mount a mounted volume with keyword parameters that conflict with those values specified when the volume was initially mounted.

PARITY ERROR

Explanation: This is a TKTN message. A parity error occurred during task execution. The task is fixed in memory so that the memory cannot be reused for another task.

XXX -- PARTITION ALREADY EXISTS

Explanation: An attempt was made to define a partition with a name already in use.

XXX -- PARTITION BUSY

Explanation: The partition in which a task was to be fixed was occupied by a running task; thus, the requesting task could not be fixed in that partition.

XXX -- PARTITION parname BUSY

Explanation: The user-controlled partition or subpartition into which the driver was to be loaded was already occupied, or there was not enough room in a system-controlled partition. (The Load command does not initiate checkpointing.)

XXX -- PARTITION parname IS A COMMON

Explanation: An attempt was made to load a device driver into a common partition. A device driver cannot be loaded into a common partition.

XXX -- PARTITION parname IS NOT A COMMON

Explanation: A partition specified for a common or resident library was not defined as a common partition.

XXX -- PARTITION NOT IN SYSTEM

Explanation: The partition name specified as an argument in a command, or during task build, could not be found in the system.

XXX -- PARTITION OR REGION name NOT IN SYSTEM

Explanation: An attempt was made to load a driver into a partition that does not exist in the system, or Load found a symbol $xxCOM in the driver and partition xxCOM does not exist in the system.

XXX -- PARTITION parname NOT IN SYSTEM

Explanation: The partition specified either in the Load command or at task build was not in the system.

XXX -- PARTITION par NOT IN SYSTEM, DEFAULTING TO GEN

Explanation: This is a warning message. The specified partition does not exist; the system therefore attempts to install the task in the GEN partition.

XXX -- PARTITION parname NOT IN SYSTEM, TASK REMOVED - taskname

Explanation: This is an RSX-11M error message. If a task is installed in more than one system, the partition control block addresses in its header mapping windows may not be for the booted system. The task's label block indicates that the task maps to parname, but parname is not in the current system. The label block has probably been corrupted.

XXX -- PARTITION parname TOO SMALL

Explanation: The partition into which the driver was to be loaded was not large enough to contain it.

XXX -- PARTITION TOO SMALL TO BUFFER INFORMATION

Explanation: The specified listing command resulted in excessive output to the buffer. The information in excess of the buffer size is not displayed.

XXX -- PAUSING. TO CONTINUE TYPE "RES tttttt"

Explanation: This is not an error message. The indirect file processor just executed a .PAUSE directive.

PMD NOT POSSIBLE

Explanation: This is a TKTN message. A post-mortem dump was requested (or the task was built specifying a PMD), but the dump could not be generated. For example, the PMD task was not installed, the PMD task and the aborted task were in the same partition and not checkpointable, or there was not enough memory available.

PMD WILL BE GENERATED

Explanation: This is a TKTN message that indicates that a post-mortem dump was requested and will be generated.

XXX -- PRIVILEGED COMMAND

Explanation: A command that was available only to privileged terminals was issued from a nonprivileged terminal.

XXX -- PRIVILEGED TASK LARGER THAN 12K

Explanation: A privileged task was larger than 12K. Privileged tasks on an unmapped system have a maximum size of 12K.

XXX -- PROCESSOR x IS ONLINE

Explanation: In an RSX-11M-PLUS multiprocessor system, all processors except one must be offline before you can boot a new system.

XXX -- PROCESSOR x IS NOT STOPPED

Explanation: A multiprocessor system can be saved only if one processor is active. CPU x is active in addition to the processor on which Save is running. Use CON to place processor x offline and retry the Save command.

XXX -- PROPER CONTROLLERS AND/OR UNITS ARE NOT OFFLINE

Explanation: When a system is saved, all device units and device controllers except the TI: of Save and the load device must be offline. Use the MCR Devices or the CON Display command to determine which devices are online and then issue a CON offline ALL command.

XXX -- PSEUDO DEVICE ASSIGNMENT ERROR

Explanation: An attempt was made to assign a logical device name to a pseudo device.

XXX -- PSEUDO DEVICE ERROR

Explanation: The specified device is a pseudo device. Pseudo devices cannot be allocated or made public.

XXX -- PSEUDO DEVICE REDIRECT ERROR

Explanation: An attempt was made to redirect a pseudo device to another pseudo device.

XXX -- PUBLIC DEVICE

Explanation: The command attempted to allocate a public device. Public devices cannot be allocated.

*** ddnn: READ FAILURE. CHECK HARDWARE STATUS

Explanation: This is a TKTN message. The card reader driver detected a hardware error.

XXX -- RECORD LARGER THAN 80. BYTES

Explanation: An indirect file for MCR contained a record having a length greater than 80. bytes.

XXX -- REDEFINING SPECIAL SYMBOL

Explanation: An attempt was made to change the value of an indirect file processor special symbol (a symbol bracketed with < and >). Special symbols may not be redefined.

XXX -- REDEFINING SYMBOL TO DIFFERENT TYPE ssssss

Explanation: An .ASK or a .SET type directive attempted to set the specified, already defined symbol to a different type. The first definition of a symbol determines its type (logical, numeric, or string); subsequent redefinitions must conform to the original type.

XXX -- REGION/PARTITION name IS A COMMON

Explanation: An attempt was made to load a device driver into a common partition. A device driver cannot be loaded into a common partition.

RESERVED INST EXECUTION

Explanation: This is a TKTN message. The task executed an illegal instruction, and no SST routine was specified to process the trap.

XXX -- RETRIEVAL POINTERS WRONG FORMAT

Explanation: The index-file retrieval pointers were not in the correct format for RSX-11M. Refer to the IAS/RSX-11 I/O Operations Reference Manual for details.

XXX -- RUNNING ABOVE SYSTEM IMAGE FILE LIMIT

Explanation: The system image file created to contain the saved system is RSX11M.SYS. This error message appears when RSX11M.SYS is not large enough to hold the memory in which both the system and the Save task run. The Save task must be included in the saved system image because Save is used to reboot the system. Either run the Save task in a partition lower in memory or rebuild the system with a larger system image file.

*** ddnn: SELECT ERROR

Explanation: This is a TKTN message. The selected device was not ready, or more than one drive had the same unit number.

XXX -- SHADOW RECORDING IS ACTIVE ON ddnn:

Explanation: This is an RSX-11M-PLUS message. The target device specified in a Dismount command was being used in shadow recording. Shadow recording requires mounted devices.

XXX -- SPACE USED

Explanation: An attempt was made to create a partition or a subpartition in a main storage area already occupied.

XXX -- SPECIFIED PARTITION FOR COMMON BLOCK

Explanation: An attempt was made to install a task in a common block.

XXX -- SPECIFIED PARTITION TOO SMALL

Explanation: The task being installed was larger than the partition into which it was being installed.

SST ABORT. BAD STACK

Explanation: This is a TKTN message. An SST cannot be effected because the SST parameters cannot be pushed onto the task's stack, or a stack overflow was detected in an unmapped system, as indicated by a nonzero value in the header guard word.

XXX -- STORAGE BIT MAP FILE I/O ERROR

Explanation: An error was detected in writing out the storage allocation file header.

XXX -- STORAGE BIT MAP FILE READ ERROR

Explanation: An error was encountered while attempting to read the storage allocation.

XXX -- STRING SUBSTITUTION ERROR

Explanation: The MCR indirect command file processor encountered an error while performing a substitution. A probable cause for the error is either the omission of a second single quote or the fact that the symbol name did not correspond to a string symbol.

XXX -- SYMBOL symname IS DOUBLY DEFINED BY FILE name

Explanation: The symbol symname is defined twice by the file ddDRV.STB.

XXX -- SYMBOL symname IS UNDEFINED BY FILE name

Explanation: The symbol symname was found in the file ddDRV.STB, but it was not a symbol definition.

XXX -- SYMBOL TABLE OVERFLOW ssssss

Explanation: The symbol table of the indirect file processor was full; there was no space for symbol ssssss.

XXX -- SYMBOL TYPE ERROR ssssss

Explanation: An .IF directive attempted to compare two types of symbol; ssssss is one of the compared symbols. Only symbols of the same type can be compared.

XXX -- SYNTAX ERROR

Explanation: This message generally indicates that the required information was entered incorrectly.

User Action: The user can usually correct this condition by retyping the arguments according to the command specifications.

XXX -- SYSTEM MAY NOT BOOT CORRECTLY

Explanation: [system uic]SAV.TSK was not found on the boot device. If SAV must be checkpointed in order to redirect and mount the booted device, the system disk will be overwritten, causing unpredictable results.

XXX -- SYSTEM MAY NOT WORK - CORRUPTED FILE FOR COMMON comname

Explanation: When Save attempted to convert to file ID for the common, Save found that the task image of the checkpointable common was corrupted or had been deleted. Save altered the system data structures to prevent checkpointing of the common into what may now be another task image.

XXX -- TASK ACTIVE

Explanation: The task used as the argument of the command was active.

XXX -- TASK ACTIVE IN ANOTHER SYSTEM, TASK REMOVED - taskname

Explanation: The task is installed in another system as well as in the booted system. Since taskname is active and mapped to dynamic regions in the other system, it cannot be run in the booted system.

XXX -- TASK ALREADY FIXED

Explanation: The task used as the argument of the Fix command was fixed in memory.

XXX -- TASK AND PARTITION BASES MISMATCH

Explanation: The base of the partition did not match that of the task being installed. This message is applicable only to unmapped systems.

XXX -- TASK AND PARTITION parname BASES MISMATCH

Explanation: The base address of the partition into which the driver was to be loaded did not match the base address of the partition address for which the task was built. This message applies to unmapped systems only.

XXX -- TASK BEING ABORTED

Explanation: A request for the execution of a task was made, but the task was being aborted either because of an error or because of an MCR Abort command.

XXX -- TASK BEING FIXED

Explanation: An attempt was made either to fix or to alter the priority of a task that was being fixed.

XXX -- TASK CHECKPOINTABLE

Explanation: An attempt was made to fix a checkpointable task in memory. A checkpointable task cannot be fixed in memory.

TASK EXIT WITH OUTSTANDING I/O

Explanation: This is a TKTN message. Tasks should terminate all I/O operations before exiting, even though the system performs all outstanding I/O when necessary.

XXX -- TASK FILE DELETED, TASK REMOVED - taskname

Explanation: While rebooting a saved system, a file identification that appears in the task control block (TCB) could not be found in the index file on the booted device. This indicates that the file has been deleted; deleted files do not contain valid task images.

XXX -- TASK FILE HEADER ERROR, TASK REMOVED - taskname

Explanation: While rebooting a saved system, a task's file header could not be found and the task was therefore removed.

XXX -- TASK FILE HEADER READ ERROR, TASK REMOVED - taskname

Explanation: The system removed the specified task because it encountered a failure while reading the task's file header.

XXX -- TASK HEADER READ ERROR

Explanation: The system detected an I/O error while reading in all task headers to store file identifications in the TCB. The error causes the system to loop.

XXX -- TASK IMAGE ALREADY INSTALLED

Explanation: The requested task image, which was checkpointable, had already been installed. On systems that do not support the dynamic allocation of checkpoint space, checkpointable tasks that do not have checkpoint space allocated can be installed only once; other tasks can be installed more than once.

XXX -- TASK IMAGE I/O ERROR

Explanation: Install could not read the task-image file, or Install could not rewrite the task-image header. (The device is write-locked.)

XXX -- TASK IMAGE I/O ERROR IN FILE filename

Explanation: The device is probably write-locked. MCR could not read the task-image file or could not rewrite the task image header.

XXX -- TASK IMAGE CURRENTLY INSTALLED

Explanation: The requested task image was already installed. This message is applicable only to checkpointable tasks.

XXX -- TASK IMAGE VIRTUAL ADDRESS OVERLAPS COMMON BLOCK blockname

Explanation: The virtual addresses reserved for the task image overlap those reserved for the common block specified in the message. A corruption of the task image file probably caused the overlap.

TASK INSTALLED IN MORE THAN ONE SYSTEM

Explanation: This is a TKTN message. Since the task is active and mapped to dynamic regions in one system, it cannot be installed in another.

XXX -- TASK INSTALLED IN PARTITION

Explanation: An attempt was made to eliminate a partition containing installed tasks.

XXX -- TASK MARKED FOR ABORT

Explanation: A previous request has already been made to abort the specified task, but the abort has not yet occurred. The task marked for abort is probably not currently resident in memory.

XXX -- TASK NAME ALREADY IN USE

Explanation: An attempt was made to install a task with the same name as one already in the system.

XXX -- TASK NOT ACP

Explanation: The task specified as an ACP did not have the characteristics of an ACP.

XXX -- TASK NOT ACTIVE

Explanation: The specified task was not currently active.

XXX -- TASK NOT FIXED

Explanation: An attempt was made to unfix a task that was not fixed, or to open a memory location of a task that was not fixed.

XXX -- TASK NOT IN SYSTEM

Explanation: The specified task had not been installed.

XXX -- TASK NOT IN SYSTEM tttttt

Explanation: An .IFACT, .IFNACT, or .WAIT directive referred to a task that was not installed. These three directives can only specify an installed task.

XXX -- TASK NOT SUSPENDED

Explanation: The task used as the argument of the Resume command was not suspended.

XXX -- TASK taskname HAS OUTSTANDING I/O

Explanation: A system cannot be saved if a task is waiting for I/O to complete.

XXX -- TASK taskname IS ACTIVE AND CHECKPOINTED

Explanation: You cannot save a system if it contains an active and checkpointed task. This restriction prevents a possible system crash; if the task was installed in another system image file, the task image could be corrupted when the task was run in the second system.

XXX -- TASK taskname IS CONNECTED TO AN INTERRUPT VECTOR

Explanation: Disconnect the task from the interrupt vector or abort it and retry the Save command.

XXX -- TASK taskname NOT INSTALLED FROM AN LB:

Explanation: All tasks must be installed from an LB:. Use the TAS command and determine which tasks are not. Retry the Save command after removing the tasks or installing them from an LB:.

TBIT TRAP OR BPT EXECUTION

Explanation: This is a TKTN message. The task has either set the T bit in the Processor Status Word or executed a Breakpoint Trap instruction, and no SST routine was specified to process the trap.

XXX -- TERMINAL ALLOCATED TO OTHER USER

Explanation: The issuing terminal has been allocated to another user. A user cannot log onto a terminal allocated to someone else.

XXX -- TERMINAL IS BUSY -- ttn:

Explanation: The broadcast message could not be displayed within 10 seconds at the terminal specified by ttn:.

TRAP EXECUTION

Explanation: This is a TKTN message. The task executed a TRAP instruction, and no SST routine was specified to process the trap.

XXX -- TI REDIRECT ERROR

Explanation: An attempt was made to redirect the pseudo device TI:. This device cannot be redirected.

TI VIRTUAL TERMINAL ELIMINATED

Explanation: This is an RSX-11M-PLUS TKTN message. The message indicates that a virtual terminal was eliminated (possibly because the parent task aborted). Nonprivileged tasks (offsprings) with the eliminated virtual terminal as TI: are aborted with this message.

XXX -- TOO MANY COMMON BLOCK REQUESTS

Explanation: A task is limited to seven common block references.

XXX -- TOO MANY LUNS

Explanation: A task attempted to assign more than 255(10) LUNs.

XXX -- TOO MANY SUBPARTITIONS

Explanation: An attempt was made to assign more than seven subpartitions in a user-controlled main partition. A main user-controlled partition is limited to a maximum of seven subpartitions.

XXX -- TOO MANY SYMBOLS OF THE FORM $xxTMO IN FILE filename

Explanation: Load is attempting to extract symbols of the form $xxTMO from the drivers symbol definition file. There are more symbols of this form than LOAD can process.

XXX -- UNDEFINED COMMON BLOCK commonname

Explanation: A task referenced a common block that was not defined in the system. Usually, this message indicates that the task was built to run in another system.

XXX -- UNDEFINED DENSITY SELECTION

Explanation: The command specified an illegal density value.

XXX -- UNDEFINED LABEL llllll

Explanation: The label llllll specified in a .GOTO or .ONERR directive could not be found.

XXX -- UNDEFINED SYMBOL ssssss

Explanation: The symbol ssssss was being tested, but it had not been defined by the indirect command file.

XXX -- UNIT DOES NOT SUPPORT 800 BPI

Explanation: The command specified a drive that does not support 800 BPI.

XXX -- UNIT DOES NOT SUPPORT 1600 BPI

Explanation: the INITVOL command specified a tape drive that does not support 1600 bpi.

XXX -- UNIT DOES NOT SUPPORT HIGH/LOW DENSITY SELECTION

Explanation: The command specified a device for which the High and Low keywords are undefined.

XXX -- UNIT DOES NOT SUPPORT DENSITY SWITCH

Explanation: The command specified a device which does have multiple density capability.

XXX -- UNIT IS NOT AT LOW DENSITY

Explanation: The command line specified that a floppy disk in an RX02 drive be initialized at low density, but the floppy was formatted at high density.

XXX -- UNIT IS NOT AT HIGH DENSITY

Explanation: The command line specified that a floppy disk in an RX02 drive be initialized at high density, but the floppy was formatted at low density.

XXX -- UNITS ATTACHED, BUSY, AND/OR MOUNTED

Explanation: One or more device units handled by the driver are attached, mounted, and/or have outstanding I/O. Therefore, the driver cannot be unloaded.

XXX -- UNKNOWN MAIN PARTITION

Explanation: An attempt was made to define a subpartition of a nonexistent main partition.

XXX -- UNKNOWN QUALIFIER

Explanation: The Help file does not contain the last qualifier printed in the display of the command line below the error message.

*** ddn: UNRECOVERABLE HARDWARE DEVICE ERROR

Explanation: This is a TKTN message. The error logging subsystem recorded an unrecoverable hardware error. TKTN prints this message as notification of such.

XXX -- USER LOGGED ON TERMINAL

Explanation: The command attempted to allocate a terminal that has been logged by another user. Logged terminals cannot be allocated.

XXX -- USER PARTITION >32K NOT PERMITTED

Explanation: The command attempted to create a partition greater than 32K. Partitions greater than 32K cannot be created in systems that do not support dynamic memory allocation, a system generation option.

XXX -- VOLUME MOUNTED FILES-11

Explanation: An attempt was made to initialize a mounted Files-11 volume. Mounted Files-11 volumes can not be initialized.

XXX -- VOLUME MOUNTED FOREIGN WITH ACP

Explanation: The command line specified a device that was mounted foreign, but with an ACP. To initialize a volume in RSX-11M-PLUS, you cannot include an ACP name when the volume is mounted as foreign.

XXX -- VOLUME NAME TOO LONG - (volume name)

Explanation: The command line specified a volume label that exceeded six characters for magnetic tape devices or twelve characters for disk devices.

XXX -- VOLUME NOT MOUNTED

Explanation: An attempt was made to create a UFD on a volume that was not mounted. A volume on which a UFD is to be created must be mounted.

XXX -- VOLUME NOT MOUNTED BY TI:

Explanation: The command specified a dismount operation for a volume that was not mounted from the issuing terminal.

XXX -- VOLUME NOT READY

Explanation: The command specified a volume that was not ready (not up to speed).

XXX -- VOLUME(S) STILL MOUNTED

Explanation: A system with mounted volumes cannot be saved.

User Action: To correct the problem, dismount the volumes and retry the command. The MCR Devices command can be used to display the mounted volumes.

XXX -- VOLUME STRUCTURE NOT SUPPORTED

Explanation: RSX-11M/M-PLUS does not support the Files-11 structure level of the volume being mounted.

In response to a Save command, while rebooting a saved system, the system determined from the disk's home block that the disk's structure is not supported by the current version of RSX-11M/M-PLUS. This error causes the system to halt.

XXX -- VOLUME WRITE LOCKED

Explanation: The command specified a volume that was write-locked and therefore could not be initialized as a Files-11 device.

XXX -- WARNING BLOCK 0 IS BAD

Explanation: Block 0 of the specified volume, the boot block, was bad. A bootable image cannot be placed on this volume.

XXX -- (WARNING) COULD NOT SIZE BOOTED DEVICE

Explanation: Save recognized the device name as a device that it should size, but it could not size it. The UCB words U.CW2 and U.CW3 for the booted device may incorrectly indicate device size. I/O errors may result.

XXX -- WARNING - KRB name INTERRUPT VECTOR nnn IN USE

Explanation: An attempt was made to load a driver that specifies an interrupt vector which is not pointing to a nonsense interrupt entry point address. This is usually caused by a previously loaded driver which has used this vector.

When you get this message, Load has loaded the driver correctly, but it has not overwritten the interrupt vector.

XXX -- WARNING - KRB name INTERRUPT VECTOR nnn TOO HIGH

Explanation: An attempt was made to load a driver that specifies an interrupt vector address that is higher than the highest permissible vector address in your system.

When you get this message, Load has loaded the driver correctly, but it has not overwritten the interrupt vector.

XXX -- WARNING - LOADABLE DRIVER LARGER THAN 4K

Explanation: Loadable drivers can be up to 8K words in length. The driver must explicitly map to any locations above 4K words.

XXX -- (WARNING) NONEXISTENT LUN ASSIGNMENT FOR TASK taskname

Explanation: The physical device to which Save was attempting to rebind a LUN does not exist in the booted system. This condition is resolved if the task performs a run-time assignment to the LUN, otherwise, the task will encounter an error condition when it attempts to use the LUN.

XXX -- WARNING - PRIVILEGED TASK OVERMAPS THE I/O PAGE

Explanation: A privileged task that is mapped into the Executive is usually also mapped into the I/O page to access the KT11 registers and other devices. Many tasks do not require this access and can use the additional 4K words of virtual address space for the task image. This message warns that a task that may require access to the I/O page might have grown too large.

XXX -- WRITE ATTRIBUTE FAILURE

Explanation: An error was encountered in writing the attributes of either the MFD or the newly created UFD.

XXX -- WRITE CHECK NOT SUPPORTED

Explanation: An attempt was made to support write check for a device other than a RK05 or RK06 disk. Write check is supported only for RK05 and RK06 disks.

XXX -- WRONG VOLUME LABEL

Explanation: The volume label and the label specified in the command did not match.

digital

digital equipment corporation

Printed in U.S.A.