IAS MCR User's Guide

Order Number: AA–H006C–TC

This manual gives an overview of the MCR terminal interface to the IAS operating system.

Operating System and Version: IAS Version 3.4

May 1990

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

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

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

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

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

This document was prepared using VAX DOCUMENT, Version 1.2

Contents

FIGURES

TABLES

Preface

Manual Objectives and Readership Assumptions

This manual provides the user with an introduction to the basic concepts of MCR, and describes its use as the terminal interface with the IAS operating system. The manual is intended for users familiar with RSX-11D who want to retain the MCR user/system interface under IAS.

The manual identifies two types of users: privileged and nonprivileged. Functions available to the privileged user provide the means for system control and modification. Functions available to the nonprivileged user are restricted to those required for program development and maintenance.

Structure of the Manual

Chapters 1 to 4 provide an introduction to MCR: accessing the system and its file structure, the concepts of tasks, and scheduling.

Chapter 5 details the differences between MCR and MCR mode.

Chapter 6 contains specifications of nonprivileged commands.

Chapter 7 contains specifications of commands available to privileged users only.

Chapter 8 describes the use of magnetic tape volumes.

Chapter 9 contains a summary of the most commonly used utilities.

Appendixes A and B contain MCR command summaries and error messages.

Appendix C contains a summary of TKB commands and error messages.

Appendix D contains system standard error codes.

Appendix E describes the node pool status program.

Associated Documents

All users should read the *IAS Utilities Manual.* Chapter 9 contains a summary of that manual.

The privileged user should become familiar with the *IAS System Management Guide*.

Other documents related to the contents of this manual are described in the *IAS Master Index and Documentation Directory*, which defines the intended readership of each document in the IAS documentation set and provides a brief summary of the contents of each manual.

1 Introduction to MCR

1.1 The Different Types of IAS Systems

IAS (Interactive Applications System) can be configured into three different types of systems:

1 Real-time system

2 Multiuser system

3 Timesharing system

You decide which system to use at system generation time (see the *IAS Installation and System Generation Guide*. The different types of systems are described in Sections 1.1.1 to 1.1.3.

1.1.1 Real-Time System

The real-time system is the simplest form of IAS system. You can configure a small number of terminals into this system, but there should be little or no program development activity. This type of system is suitable only for real-time and single-user applications.

1.1.2 Multiuser System

You can use the multiuser system in a mixed real-time and program development environment. The multiuser system includes the IAS scheduler, which controls tasks and provides optimum service to all users, but does not have the access, regulation, control, and protection facilities of timesharing control primitives (TCP).

1.1.3 Timesharing System

The timesharing system is used in a program development environment with a lower emphasis on real-time applications. The difference between the multiuser and the timesharing system is that timesharing includes TCP, providing a number of protection and privilege control features typically required in a timesharing environment.

1.2 The MCR User Interface

The monitor console routine (MCR) is one of the user/system interfaces provided by the IAS operating system. MCR is the normal user interface on real-time systems, and MCR can be the user interface on multiuser systems.

The other IAS user interface is the Digital command language (DCL), implemented by the program development system (PDS). PDS can be the user interface on multiuser systems, and is the only user interface on timesharing systems. MCR can, however, be used on timesharing systems by entering MCR mode (see Section 1.3. PDS is fully described in the *IAS PDS User's Guide*.

MCR and PDS are both examples of command language interpreters (CLIs). Users can write their own CLI if desired (see the *IAS Guide to Writing a Command Language Interpreter*).

1.3 MCR Features

The major features of MCR are listed below and described in the rest of this manual.

1 MCR performs operations in response to MCR commands.

Commands can be general user commands, available to all authorized MCR users, or privileged commands, available to users who log in with a group code less than 10 (octal). General user commands are described in Chapter 6. Privileged commands are described in Chapter 7.

2 MCR enables you to access IAS by logging in and to leave IAS by logging out.

3 MCR maintains system security by examining your access rights.

MCR maintains file security by examining a file's protection code before granting you access to the file. Files and protection are described in Chapter 3.

4 MCR (and PDS) enable you to assemble and compile source programs and to link object modules to produce programs known as tasks. See Chapter 4 for a description of MCR tasks.

5 MCR supports indirect command file processing. MCR indirect command files are described in the *IAS Indirect Command Processor Reference Manual.*

6 MCR provides an interface to the system utilities. Chapter 9 briefly describes the utilities and refers you to the appropriate manual.

7 IAS enables MCR commands to be issued at a PDS terminal by way of MCR mode. See Chapter 5 for a description of MCR mode.

2 User/System Communication

2.1 Monitor Console Routine

The monitor console routine (MCR) is an interface between you and the operating system. MCR commands enable you to perform the following functions:

1 Gain access to the system.

2 Initiate and terminate execution of user/system programs.

3 Adjust, modify, and control the system environment.

This chapter describes the conventions and procedures observed when using MCR to communicate with the operating system. General and privileged MCR commands are described in Chapters 6 and 7. Certain differences exist between MCR and MCR mode; these are described in Chapter 5. However, information contained in this chapter is relevant to MCR mode users and should be read in conjunction with Chapter 5.

2.1.1 MCR Organization

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

2.1.2 Command and Slave Terminals

The MCR user communicates with the system via a user terminal. User terminals are of two types:

1 Command terminals

2 Slave terminals

You use the command terminal to activate MCR and gain access to the system. Once system access is established, you use the terminal to initiate system commands or respond to system prompts.

Slave terminals are passive; they can only respond to, or interact with, user tasks. For example, you might use a slave terminal in a process control environment to display or print out data relevant to an operation.

The MCR dispatcher task (or PDS) must be active to receive requests (system commands) from the user terminal.

NOTE: The system ignores any attempt to use a designated slave terminal as a command terminal.

The classification of a terminal is the responsibility of the system manager or other privileged users. They can classify a terminal as being command or slave by using the SET command (see Chapter 7).

2.1.3 Activating MCR

You activate MCR by typing Ctrl/C on the terminal keyboard (see Section 3.4). When MCR becomes active, the system issues the following prompt:

```
MCR>
```

At this point, the link between you and MCR is established.

2.1.4 Logging In

To gain access to the system (that is to log in) issue the HEL (HELLO) command. The HEL command performs the following functions:

1 Identifies you to the system.

2 Establishes your privileges.

3 Grants you access to the system.

When access has been granted, you can issue MCR commands and access files (depending on your access privileges).

NOTE: Typing Ctrl/C does not affect the execution of any tasks currently running (except when you are running in MCR mode on a timesharing system—see Chapter 5).

2.1.5 User Privileges

The system determines your associated privileges by examining the user identification code (UIC) under which you log in. UICs are issued by the system manager.

Two categories of MCR user are granted access to IAS:

1 Privileged

2 Nonprivileged

The privileged user is identified by a UIC group code of less than 10 (octal) and can control the system through privileged commands. These commands are described in Chapter 7. The nonprivileged user is identified by a UIC group code of 10 (octal) or more, but is not able to control the system. A subset of MCR system commands, described in Chapter 6 is provided for the nonprivileged user.

2.1.6 Terminal Privileges

The two types of terminals are privileged and nonprivileged.

If you are a privileged user, a terminal where you log in automatically becomes a privileged terminal. Because you are a privileged user, any terminal you use automatically becomes privileged until you log out.

While you are logged in as a privileged user at a privileged terminal, you can set any other terminal privileged or nonprivileged by using the SET /PRV or SET /-PRV commands. You can, therefore, set a terminal privileged even if it is being used by a nonprivileged user. Any nonprivileged user currently using a terminal that has been set privileged can issue privileged commands until logging out.

2.1.7 Entering Commands

Enter MCR commands in response to the MCR> prompt. Once the prompt is issued, MCR waits for a command to be entered. The command is issued after the right angle bracket character (>). If the command is not entered within a preset period, MCR *times out* (see Section 2.1.8). To reactivate MCR, type Ctrl/C.

Enter commands in the following format:

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

where:

- command = Function to be performed.
- command string = Mandatory or optional command modifiers or switches.

The command line is stored by the MCR dispatcher, which then initiates the corresponding MCR system command task and exits. The MCR command task then interprets the command line that the MCR dispatcher stored and performs the desired function.

If the command line was terminated by carriage return RET, the MCR dispatcher is reactivated for that terminal when the command task exits. If the command line was terminated by ESC, the MCR dispatcher is not reactivated. In this case, you must type Ctrl/C to reactivate MCR.

2.1.8 Timeout

MCR waits for a specific length of time for a command to be entered. This period, known as timeout, is normally five minutes (set at system generation) unless changed by the system manager. If this period is exceeded, MCR times out. All information typed after the previous line terminator RET or ESC is lost. The timeout is reset whenever you type a character.

Commands issued after MCR has timed out are ignored. To reestablish communication with MCR, you must type Ctrl/C.

2.2 Initiating Tasks

Table 2–1 illustrates what the UIC of an executing task is, according to whether the task is initiated by a privileged or nonprivileged user and depending on the state of the task prior to initiation. See Chapters 6 and 7 for nonprivileged and privileged versions of the RUN command.

Table 2–1 UIC of an Executing Task

Initiation Method	Task Already Installed		Task Not Installed, but to Be Removed on Exit		Task Not Installed	
	Privileged User	Nonprivileged User	Privileged User	Nonprivileged User	Privileged User	Nonprivilege User
1	User UIC[1]	User UIC[1]	N/A	N/A	N/A	N/A
2	Task UIC[2]	Illegal	N/A	N/A	N/A	N/A
3	User UIC[1]	Illegal	User UIC[1]	User UIC[1]	N/A	N/A
4	User UIC[1,3]	Illegal	User UIC[1,3]	User UIC[1]	User UIC[1]	User UIC[1]

[1] This is the UIC specified in the HEL command when the user logs into the system.

[2] This is the UIC specified during taskbuilding, when the task is installed, or when an explicit UIC is given in a RUN command. If no UIC is specified, the user UIC becomes the task UIC when the task is actually activated.

[3] This user UIC can be overridden at runtime if you use the /UIC option with the run command.

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

1 MCR>XYZ (for example, MCR>PIP)

2 MCR>RUN ...XYZ (for example MCR>RUN ...PIP)

3 MCR>RUN $file-specification (for example, MCR>RUN $EDI)

4 MCR>RUN file-specification (for example, MCR>RUN [200,200]MYTASK)

2.2.1 Error Reporting

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

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

The error messages are listed and described alphabetically in Appendix B. The only messages that do not have the three-letter prefix are those issued by the MCRERR task. Error messages from the Task Builder (TKB) are listed and described in Appendix C.

3 MCR Files and Devices

3.1 Introduction

This chapter describes MCR files and devices. The following topics are covered:

1. Files (Section 3.2).
2. File name strings (Section 3.3).
3. Terminal control (Section 3.4).
4. Peripheral devices (Section 3.5).

3.2 Supported File Structures

IAS supports the following two file structures:

1. Files-11 for disk and DECtape volumes.
2. ANSI Standard Level 3 for single-volume or multi-volume magnetic tape.

Files-11 is described briefly in Section 3.2.1. See the *IAS I/O Operations Reference Manual* for detailed information.

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

> **Magnetic Tape Labels And File Structure For Information Interchange Order No. ANSI X3.27-1969**

ANSI Standard Level 3 is also described in the *IAS I/O Operations Reference Manual.*

3.2.1 Files-11

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

Files-11 has volume and file protection, which allows the owner of a volume or file to deny or enable access to other users in the system. Volume and file protection provides the key to system security (see Section 3.2.7 for a full description).

3.2.2 Files-11 Volumes

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

Each Files-11 volume has two kinds of directory files that are used for file management:

1 The master file directory (MFD) file is automatically generated by the file system when a volume is initialized as a Files-11 volume. The MFD is used to store pointers to all the user file directory (UFD) files on the volume.

2 The user file directory (UFD) files are created as needed. They are used to store pointers to all the files belonging to, or associated with, the user whose UIC corresponds to the UFD file name. UICs are described in Section 3.2.5.

3.2.2.1 File Format

All Files-11 files, whether MFD, UFD, or user files, have the same basic format. Figure 3–1 shows a sample Files-11 file structure. All files have a header area and one or more data areas.

The file header area contains all the information required by the file system to process the file. For the purpose of this introduction, you should be familiar with only the following fields:

1 File owner field—Contains the UIC of the user who created the file.

2 File name field—Contains the name assigned to the file when it was created. File names can be a maximum of nine alphanumeric characters in length, and are described in Section 3.3.

3 File type field—Contains the mnemonic that identifies the file by some aspect of its contents (for example, FTN defines a FORTRAN source file). File types are described in Section 3.3.

4 Version number field—Identifies the particular version or generation of the file. Version numbers are described in Section 3.3.

5 File protection field—Contains a 16-bit code that describes the access to the file. The field also describes the type of access allowed; that is, Read, Write, Extend, or Delete. Volume and file protection is described in Section 3.2.7.

6 Data pointer field—Describes the physical allocation of the file on the volume. Each data area pointer describes a physically contiguous portion of the file. Figure 3–2 illustrates this physical allocation of a file.

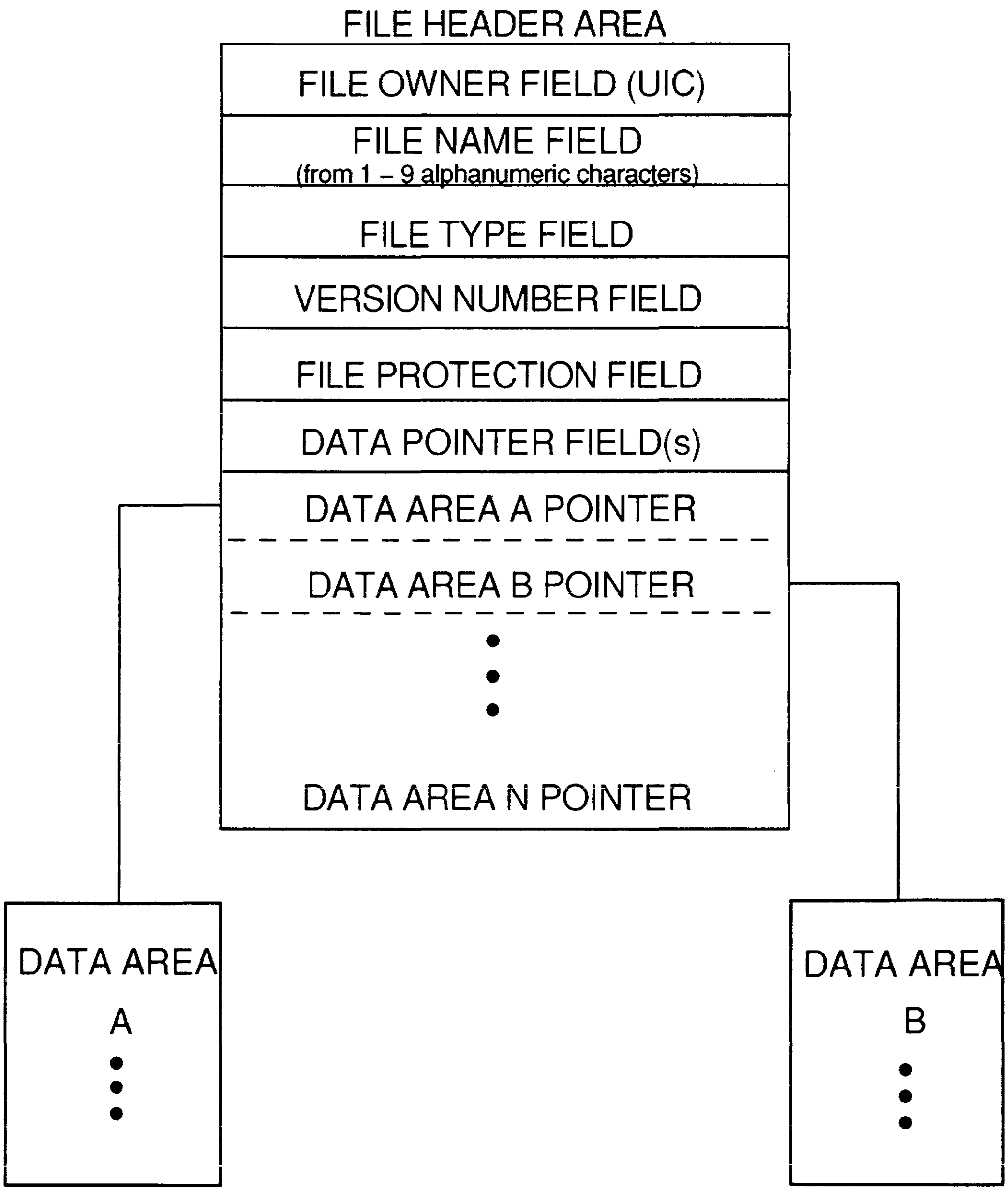
By establishing pointers to blocked data in the file header area, as opposed to storing the data immediately following the file header, you can accomplish the following results:

1 All files on the volume have the same structural format, regardless of their use.

2 All fragmented or noncontiguous areas of the volume can be put to optimum use; that is, a you can expand a file by attaching another pointer to a blocked data area to its file header.

You always address data in a file-relative manner. The file system translates file-relative addresses into physical addresses.

Figure 3–1 Sample Files-11 File Structure



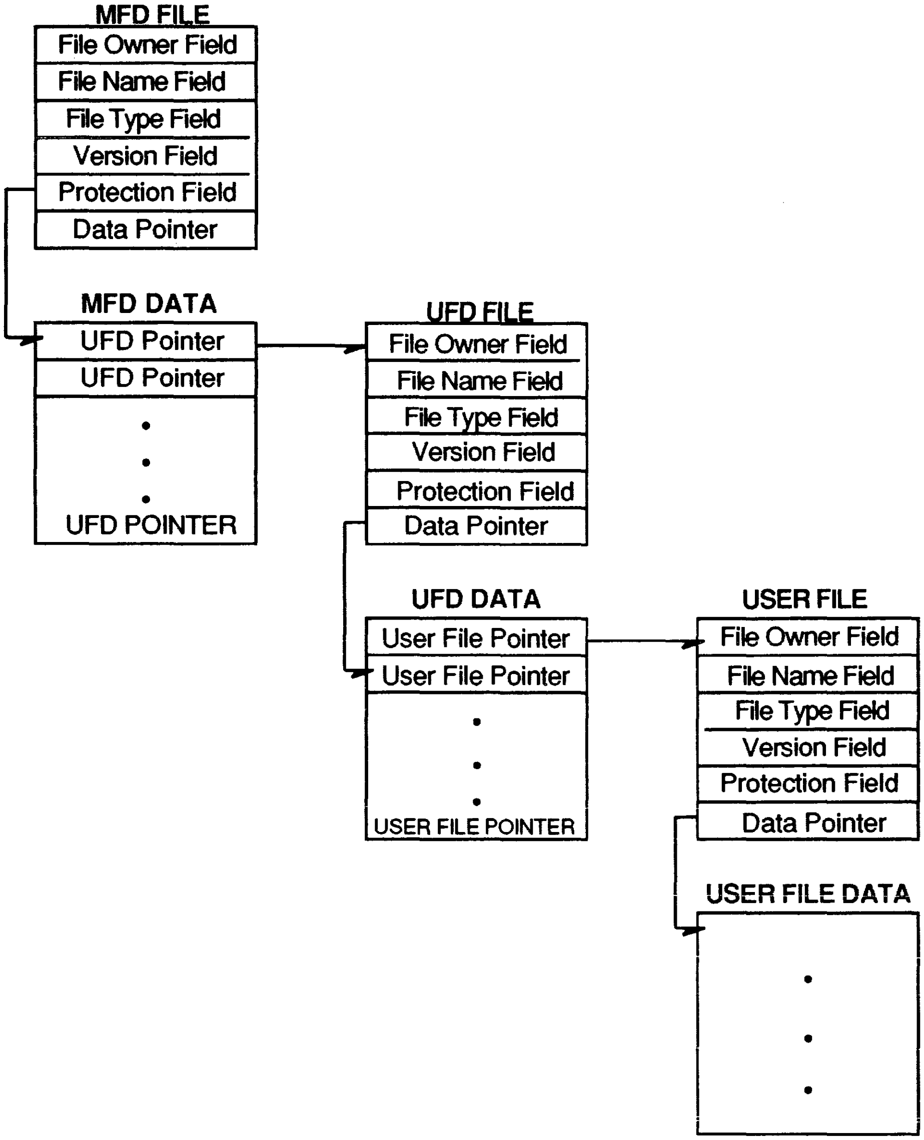
3.2.3 Master File Directory (MFD)

The file system uses the master file directory (MFD) to locate requested UFDs. The MFD of the system device serves a dual purpose:

1 It aids the file system to locate UFDs on that volume.

2 It is used by the operating system to identify users as they log in.

Figure 3–2 Files-11 File Access



The MFD is automatically generated when the volume is initialized as a Files-11 volume. Files-11 volumes can be initialized only by the system manager or another privileged user.

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

3.2.4 User File Directory (UFD)

The UFD is used by the file system to locate user files that are associated with or owned by the UFD owner.

UFD files are generated, as needed, by the system manager or another privileged user. When a user file is created, a pointer to the file header area is established in the owner's UFD. The protection attributes assigned to the UFD establish the access rights to the files to which it points. A user whose access rights are not consistent with the protection attributes assigned to the UFD cannot access any of the files pointed to by the UFD. See Section 3.2.7 for a description of file protection.

3.2.5 User Identification Code (UIC)

User Identification Codes (UICs) are unique account numbers assigned to each user by the system manager.

UICs perform the following functions:

1 Identify you as someone with authorized access to the system.

2 Identify you as a privileged or non-privileged user. A privileged user is assigned a UIC with a group code of less than 10 (octal).

3 Identify your access privileges (see Section 3.2.7).

4 Establish your default UFD (see Section 3.3.1).

NOTE: Under MCR, tasks are normally run using the UIC under which you have logged into the system. See the description of the RUN command in Chapter 7 for exceptions to this rule.

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

```
[ggg,mmm]
```

where:

- ggg = Octal number ranging from 1 to 377 representing the user's group code.
- mmm = Octal number ranging from 1 to 377 representing the user's unique identification within the group or department (that is, programmer code).

For example:

```
[200,20]
```

identifies the user's group as 200 and the user's individual number within that group as 20.

3.2.6 File Ownership

File ownership is determined by the UIC stored in the file owner field of the file header. The system considers the user whose UIC exactly matches the UIC stored in the file owner field to be the file's owner.

3.2.7 Volume and File Protection

Files-11 provides the user with a facility to protect volumes and files against unauthorized access. To ensure such protection, a user can specify protection attributes for the entire volume, as well as for each file within the volume, regardless of the file's hierarchy or contents.

Four types of access are available for a file:

1 Read—Read access.

2 Write—Write access.

3 Extend—Extend access (necessary to install or run a task).

4 Delete—Delete access.

Tasks that can perform combinations of these operations are also divided into four categories:

1 System—Comprises all tasks that run under a system UIC. System UICs have a group number of less than 10 octal (for example, [2,200]).

2 Owner—Comprises all tasks that run under a UIC that matches the UIC in the file owner field.

3 Group—Comprises all tasks that run under a UIC whose group number matches the group number of the UIC in the file owner field.

4 World—Includes any task not included in the three categories described above.

Every file has a file protection field associated with its file header. This field contains a 16-bit code that describes the file's protection attributes. Figure 3–3 shows the format of the file protection word. Before file access is allowed, the file protection code is interrogated by the file system to determine if the requesting task is allowed to perform a requested action.

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

1 The task UIC is compared to the file owner UIC to determine the task's category (see Figure 3–4).

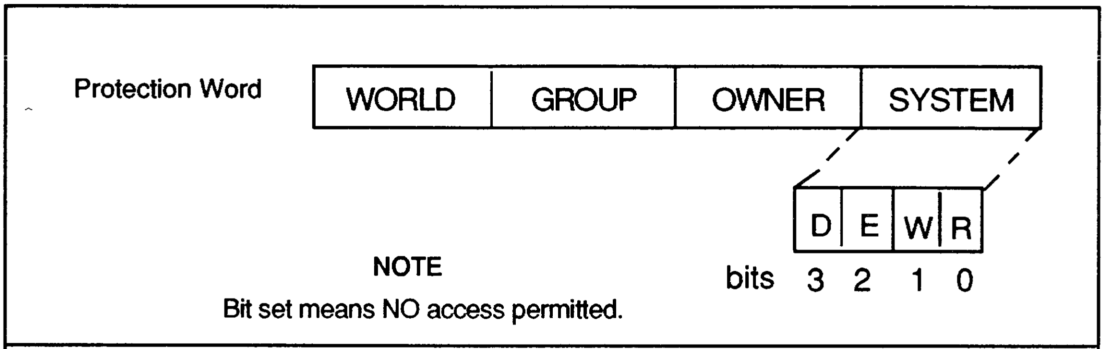
2 When the task category is determined, the file system inspects the file protection word to determine if the task's category is allowed access to the file.

3 The file system then analyzes the file protection word to determine if the function is allowed for this category of task.

If all these checks determine that the task is authorized to access the file, access is permitted.

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

Figure 3–3 Format of the File Protection Word



NOTE

Bit set means NO access permitted.

Example

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

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

3.3 MCR Command Format

MCR commands have the following format:

```
outfile1/sw1,outfile2/sw2...=infile1/sw1,infile2/sw2..., infilen/swn
```

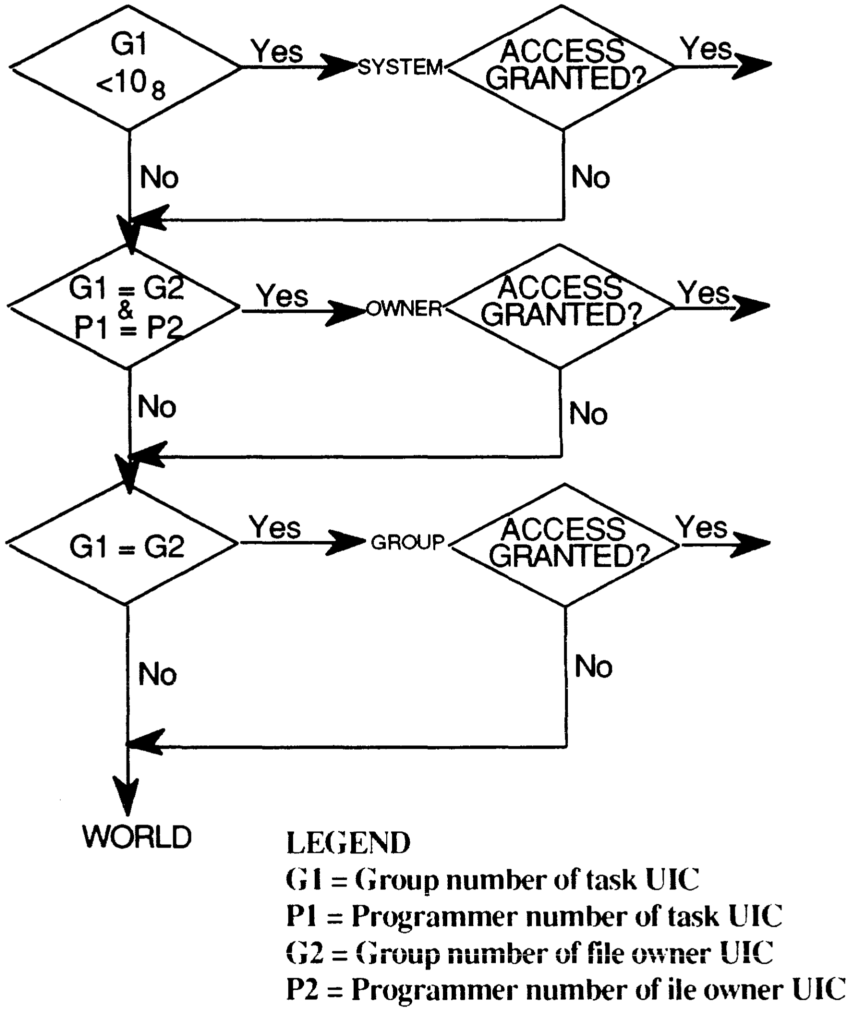
Input and output files are specified in the following format:

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

where:

- dev: = Physical device where the volume containing the desired file is mounted (for example, DK0: or DT1:). The colon is required as part of the device specification.
- [ufd] = UFD containing the desired file.
- filename = Name of the file. In MCR, file names can be up to nine alphanumeric characters long. File name and type are always separated by a period.
- .type = Means by which various forms of the same file are distinguished. For example, a source FORTRAN file might be named COMP.FTN, while the object file associated with that file might be called COMP.OBJ. The file type and version are always separated by a semicolon. File types are described in Section 3.3.2.

Figure 3–4 Formula for Determining Task File Access Category



LEGEND
G1 = Group number of task UIC
P1 = Programmer number of task UIC
G2 = Group number of file owner UIC
P2 = Programmer number of ile owner UIC

- ;version = Octal number used to distinguish versions of a file. For example, when a file is first created with the Editor, it is assigned a version number of 1. If the file is subsequently opened for editing, the Editor keeps the original file for backup and creates a new file with the same filename and type, but with a version number of 2. Version numbers can range from 1 to 77777 octal.
- /sw1.../swn = Usually optional qualifiers. Switches are normally used either to direct the execution of a task, or to qualify an input parameter.

Input and output file specifications are separated by an equal sign (=). Optional switches are used to indicate desired actions.

3.3.1 File Specification Defaults

If any of the fields of the file specification except the file name is omitted, the system uses a default value. A task can establish defaults for a file. When a task does not specify defaults, the defaults listed in Table 3–1 apply.

Table 3–1 File Specification Defaults

Field	Default
dev:	If omitted in the first or only file specification, SY: (system device) is assumed.
	If the unit number is omitted from the device specification, unit 0 is assumed.
	If omitted from subsequent file specifications, the device (specified or defaulted) for the previous file specification is used.
[ufd]	If omitted from the first or only file specification, the UFD that corresponds to the UIC under which the task is running is used.
	If omitted from subsequent file specifications, the UFD (specified or defaulted), for the previous file specification is used.
filename	No default—you must specify the file name either explicitly or implicitly (using a wildcard as described in Section 3.3.3), except in certain PIP and FLX strings.
.type	If, for example, a list file is created, the system assigns it a file type of .LST. Table 3–2 lists default file types.
;version	For input files, the most recent version number. For output files, the next highest version number. An exception is the PIP file delete function which requires an explicit version number; this feature prevents the user from inadvertently deleting the latest version of a file.

NOTE: The carrying forward of a specified device and/or UFD from one file specification to the next does not apply across the equal (=) sign.

3.3.2 MCR File Types

MCR has a defined set of file types that the system uses when the type is omitted from a file specification. You can assign file types, using the abbreviations listed in Table 3–2. File types can be defaulted in most commands. For example, if the FORTRAN compiler is given the source file XYZ to compile, it automatically searches for XYZ.FTN.

Table 3–2 MCR File Types

File Type	Meaning
BAS	A BASIC language source file
BIS	A Batch input stream
CBL	A COBOL language source file
CMD	A file containing a list of commands to a system program (that is, an indirect file)
COR	A CORAL language source file
DAT	A data file (as opposed to a program file)
DIR	A directory file, for example, a UFD directory
EML	An EDITOR macro library file

Table 3–2 (Cont.) MCR File Types

File Type	Meaning
FLB	A form library file (for use with FMS-11)
FMD	A form description file (for use with FMS-11)
FRM	A form file (for use with FMS-11)
FTN	A FORTRAN language source program
LST	A file in print-image format
MAC	A MACRO assembly-language source program
MAP	A file associated with Task Builder output
MLB	A macro library file
OBJ	An object program (output from MACRO or FORTRAN)
ODL	An overlay descriptor file
OLB	An object library file
SAV	A system image file
SML	A System Macro Library file
SPR	A spooled output file
SRT	A sort input, output, or specification file
STB	A symbol table file
SYS	A file reserved for system use
TMP	A temporary file
TSK	A file containing a runnable task image

3.3.3 The Wildcard Convention

Asterisks (*) placed in the file specification indicate that the corresonding fields in a command string are to be ignored. This is called the wildcard convention. By ignoring the content of elements within a command string, you can operate on more than one file at a time. For example, to delete three files named PROG.MAC;1, PROG.OBJ;1, and PROG.TSK;1, the following wildcard specification can be used rather than three explicit delete requests:

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

The PIP> prompt indicates that the PIP utility is running. PIP is the utility used to delete the files.

If the file PROG exists with other version numbers, for example, PROG.MAC;2, the other versions are not affected by the delete request.

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

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

3.3.4 Examples of Command Strings

The following are examples of command strings.

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

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

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

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

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

A request to delete file SBG.OBJ;5 from device SY: in the default UFD. The output file name string is omitted because it is not applicable.

```
TKB>CRGPT=CRGPT
```

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

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

A request to delete all files under the default UFD, on SY:, with the name CRGPT, regardless of file type or version.

3.4 Terminal Conventions

Several special keyboard characters that cause specific functions to be performed are recognized by the system. These functions are described in Table 3–3.

Perform control functions—for example, Ctrl/C, by holding down the Ctrl key while you type the letter C.

Table 3–3 Terminal Control Conventions

Keys	Function
Ctrl/C	Causes MCR to be activated. The system prints the prompt MCR>. The functions of Ctrl/C can be altered for specific applications (see the *IAS Device Handlers Reference Manual*). **NOTE**: Typing Ctrl/C does not affect the execution of any tasks currently running (except on a full timesharing system using MCR mode).
Ctrl/Z	Logical end-of-file. Causes a system program to exit when typed in response to a prompt.
RET	Terminates the current line and causes the system to print the prompt for the next command. All lines are terminated using RET unless otherwise stated.
DEL	Erases the last character typed. You can use this key repeatedly to erase a number of characters. On a VDU the current printing position moves to the left and erases the deleted characters. On other terminals the string of deleted characters is echoed between an initial backslash (\) and a final backslash (\).
ESC	Terminates MCR. Normally used when requesting a program (user or system) that is to interact with the operator after the command is executed, for example, RUN ...MAC ESC.

Table 3–3 (Cont.) Terminal Control Conventions

Keys	Function
Ctrl/B	For a terminal set in TAPE mode and with a low speed paper-tape reader attachment switched on, this signals the computer to start reading the tape.
Ctrl/I	Causes a horizontal tab. Tab stops are set by the software at every eighth character position (for example, 9, 17, 25, 33).
Ctrl/K	Causes a vertical tab.
Ctrl/L	Causes a formfeed to the top of the next page.
Ctrl/O	Interrupts system output to the terminal. Successively pressing Ctrl/O causes output to start and stop. For example, if a directory listing on the terminal is requested and the first few lines present the desired information, CT/O can suppress the printing of the rest of the directory.
Ctrl/R	Causes the system to print the current terminal line. When errors make the line difficult to read, pressing Ctrl/R produces a clean copy. Input can continue on the newly printed line.
Crl/T	On a terminal with a low-speed paper tape reader, this stops reading from the tape. This can be present on the tape, or the reader can be switched off and Ctrl/T typed on the keyboard.
Ctrl/U	Cancels the current input line. The prompt, (for example, MCR> or PIP>) is not printed on the next line, but the system is ready to type a new command.
Ctrl/V	Flushes all characters typed ahead of a read. If a read is in progress, this has no effect. For type-ahead modes see the *IAS Device Handlers Reference Manual.*
Ctrl/X	Causes a user-written program named TTYNnn to execute; nn is the two digit unit number of the terminal with which the program is to interact during execution. This facility is not available in MCR mode.
Ctrl/Q and Ctrl/S	These two keys correspond to XON and XOFF respectively. Pressing Ctrl/S (XOFF) stops output to the terminal Ctrl/S until Ctrl/Q (XON) is pressed. Unlike Ctrl/O, the XON/XOFF function stops and starts output without any loss of characters.

3.5 Peripheral Devices

You refer to all peripheral devices by using a two-letter name and a one- or two-digit unit number followed by a colon; for example, DB0: and TT12:. If you omit the unit number, the system uses unit 0 by default (for example, LP: indicates line printer 0). You can refer to peripheral devices by using mnemonics or pseudo-device names. The system manager links pseudo-devices to device mnemonics. The logical unit numbers (see Section 3.5.7) for each task are in turn associated with peripheral devices to provide another level of device independence. Table 3–4 lists the mnemonics of devices supported by IAS.

3.5.1 Pseudo-Devices

IAS supports pseudo-devices and physical devices, both of which are identified by mnemonics. The physical and pseudo-device mnemonics are listed in Table 3–4. Linking pseudo-devices to physical devices permits a system manager to determine dynamically the physical devices that send or receive system information.

Table 3–4 Device Mnemonics

Mnemonic	Device Type
AD	AD01 A/D converter
AF	AFC11 Analog input
CI	Console Input
CL	Console Log
CO	Console Output
CR	Card reader
CT	Cassette
DB	RP04/5/6 disk
DD	TU58 cartridge
DF	RF11 disk
DK	RK05 disk
DL	RL01 or RL02
DM	RK06 or RK07 disk
DP	RP02 or RP03 disk
DR	RM02/RM03/RM05 disk
DS	RS03 or RS04 disk
DT	DECtape
DU	RA60, RA80, RA81, RA82, RA90 RD31, RD32, RD51, RD53, RD54 RX50, RX33, RC25
DX	RX01 floppy disk
DY	RX02 floppy disk
LB	Device holding system library files
LP	Line printer
LS	LPS A/D converter
MM	TU16, TE16, TU45, or TU77 magnetic tape
MO	Message output
MS	TS11, TS05
MT	TU10, TE10, or TS03 magnetic tape
MU	TU81, TK50, TU81-PLUS
NL	Null device
PP	Paper tape punch
PR	Paper tape reader
SP	Device holding spooled I/O files
SY	User's system disk
TI	User's data input stream
TO	User's data output stream
UD	UDC11 Universal Digital Control

Table 3–4 (Cont.) Device Mnemonics

Mnemonic	Device Type
WK	Fast-access device for work files

The mnemonics CI, CL, CO, LB, MO, NL, SP, SY, TI, TO, and WK, are logical device names (pseudo-devices) that can be made to refer to particular physical devices according to the needs of the installation.

3.5.2 System Device (SY)

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

3.5.3 Terminal Interface (TI)

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

When TI: is used as a device specification, it refers to the terminal at which the command is issued

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

The system manager uses the psuedo-device CI to communicate with the system. The system uses the CO pseudo-device to communicate with the system manager. The pseudo-device CL is used to list the files; output is normally directed to the printer.

3.5.5 Message Output (MO)

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

3.5.6 Spooling Output (SP)

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

3.5.7 Logical Unit Numbers (LUNs)

Logical Unit Numbers (LUNs) provide the mechanism for programs to maintain device independence. The LUNs used in a program can be assigned by means of device mnemonics to any available peripheral device that performs the desired function. The programmer can assign LUNs at assembly time or at task build time, or by the task at run time. Because the system provides default LUN assignments, it is not always necessary to assign a LUN for a task. Default LUN assignments are listed in Table 3–5.

Table 3–5 Default LUN Assignments

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

3.6 Card Reader Control Conventions

The card reader handler recognizes three control characters. The control character code identifies the alphanumeric format for either 029 card codes or 026 card codes, or it indicates end-of-file. Table 3–6 defines the control characters. See the *IAS Device Handlers Reference Manual* for information on default card codes.

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

Table 3–6 Card Reader Control Conventions

Card Column 1 Multipunch	Meaning
12-11-0-1-6-7-8-9	End-of-file ASCII mode[1]
12-0-2-4-6-8	029 mode
12-2-4-8	026 mode

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

NOTE: Control cards must enclose input data streams. The first card must specify the card code and the last card must specify end of file.

4 MCR Tasks

4.1 Tasks

A task is a source program, written, for example, in MACRO-11 or FORTRAN with the following characteristics:

1 It is assembled or compiled.

2 It is task built to make it relocatable.

3 It is installed into the system.

You can combine one or more source modules during compilation (assembly) or task building to form one task.

When a task is installed in IAS, the system establishes an entry in an internal table called the system task directory (STD). The STD holds information about the location of the task disk image. When a request for task execution is made, the system uses this information to load the task image into memory and start its execution. Tasks can also be fixed in memory, to allow faster task initiation and rescheduling by the system. Each task can execute independently or in conjunction with other tasks in its own or other partitions (see Section 4.1.3. A request for task execution can be made in one of the following ways:

- By a user at a terminal.
- By an executing task.
- By a request in a batch stream.
- By the IAS scheduler. (See the *IAS System Management Guide* for a description of the IAS scheduler.)

4.1.1 Overlays

IAS provides a task overlay facility that can be used to reduce the amount of memory a program requires. A task can be divided into several segments: a main segment, that must be resident throughout task execution, and other segments that are used less frequently and which are loaded into memory only when needed.

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

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

Overlaying is described in the *IAS Task Builder Reference Manual*.

4.1.2 Multiuser Tasks

More than one user can request a multiuser task for execution. Both system tasks and user-written tasks can be built as multiuser tasks.

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

When a task that consists of a pure and an impure area is run by more than one user, only one copy of the pure area resides in memory. One copy of the impure area is required for each user.

4.1.3 Task Interaction

Because a number of tasks can execute at the same time, four techniques for task interaction are provided:

1 Global Event Flags.

2 Shareable global areas (resident libraries and common blocks) and shared regions. See the *IAS Task Builder Reference Manual for further details.*

3 Shared access to data files.

4 SEND/RECEIVE and RECEIVE BY REFERENCE Executive directives.

These techniques are fully described in the *IAS System Directives Reference Manual.*

4.1.3.1 Global Event Flags

IAS provides a set of 32 global event flags that can be set and tested by all tasks in the system. Using the event flags, a task can wait for another task to perform a designated function before resuming execution. The waiting task tests a specified event flag; when the executing task has performed the designated function, it sets the flag, thus allowing the waiting task to resume. The system manager coordinates the use of global event flags. Global event flags 57 through 64 are reserved for system use.

Global event flags are described in the *IAS Executive Facilities Reference Manual.*

4.1.3.2 Shareable Global Areas (SGAs) and Shared Regions

Shareable global areas and shared regions provide a means for two or more tasks to share a single data area. All tasks access the data area at the same time, and one task can pass information to another by placing it there.

4.1.3.3 Shared Data Files

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

4.1.3.4 SEND/RECEIVE Directives

SEND/RECEIVE directives enable tasks to transmit data using either the system node pool or real memory as the data storage medium.

4.2 Memory Protection

Memory management hardware accesses the memory area allocated to a task. A task's memory area is protected from other tasks executing in the system; in addition the task can access only those areas allocated to it. The hardware also controls read and write access rights. The areas of memory that can be used concurrently by more than one task, such as resident library and common area (both SGAs), are also protected. Resident library SGAs contain reentrant (pure) code and have read-only protection. Common area SGAs can be either pure or impure; the memory management hardware controls access. Care must be taken when a common area SGA is shared between tasks with read/write access.

4.3 Task Scheduling

IAS schedules tasks for execution in two ways:

1 On a real-time system, each real-time task has an associated priority in the range 1 to 250. 250 is the highest, or most urgent, priority. The Executive constantly tries to allocate the processor to the runnable task with the highest priority.

2 On a multiuser or timesharing system, the IAS scheduler, in addition to the Executive, controls task scheduling. The IAS scheduler uses a heuristic round-robin algorithm to give optimum service to all users. Timesharing tasks, multiuser tasks, and real-time tasks that run in a timesharing partition at a priority less than or equal to that of the IAS scheduler, all run under scheduler control. Real-time tasks with a higher priority than the scheduler run under Executive control.

The IAS scheduler is described in the *IAS System Management Guide.*

The system might not be able to fit all currently active tasks into available memory at the same time. In this case, an active task might have to be temporarily removed from memory to make room for other tasks. This process is called checkpointing for a real-time task and swapping for a timesharing or multiuser task. Different names are used because of the different ways the system removes tasks from memory.

Checkpointing and swapping are described in Sections 4.4 and 4.5.

4.4 Checkpointing

Checkpointing is a means of suspending the operation of a real-time task when there is insufficient memory available to run a higher priority task. The system checkpoints out of memory-sufficient resident task(s) of lower priority than the higher priority task to make the desired space available. The checkpointed task is recorded on disk for subsequent retrieval.

Only real-time tasks that you build checkpointable can be checkpointed. For further information, see the *IAS Executive Facilities Reference Manual.*

4.5 Swapping

Swapping is a means of enbling more IAS scheduler-controlled tasks to execute than would otherwise fit into available memory. Swapping is under the control of the IAS scheduler. The whole process consists of first moving resident tasks in memory to create the required contiguous space; this is known as shuffling. Then, if necessary, tasks are moved out of memory to a disk; this is known as swapping.

For further information, see the *IAS Executive Facilities Manual* and the *IAS System Management Guide*.

4.6 Device Independence

One of the major factors in determining whether a program can run is resource availability (that is, whether the necessary peripheral devices are free for use). For the sake of flexibility, IAS provides device independence. This feature enables a programmer to code a program with input and output assigned to logical units rather than to specific physical devices. Before scheduling the task for execution, you can assign physical devices to the logical units specified in the task I/O. If the task uses the default device assignments, which are devices readily available for use, you need not assign devices.

4.7 Output Spooling

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

The system can temporarily reroute the output intended for the serial device to a disk file. The files rerouted to the disk are printed later on the original device. This process is known as automatic output spooling.

The automatic output spooler and the terminal user can both queue files for printing. Queued files are printed according to priority. To ensure that the spooled files are deleted after output, the terminal UIC must be the same as the UIC under which the requesting task is running. Therefore, do not change the UIC until the spool requests have been queued.

4.8 Input Spooling

If the system has a spooled card reader, you can input batch jobs to the batch queue via that card reader. The commands on the cards must conform to the PDS batch command format and can contain DCL and/or MCR format commands. (See Section 5.6.2 and the *IAS PDS User's Guide*.)

5 MCR Mode Under PDS

5.1 Introduction

Digital Command Language (DCL) is the default user interface on timesharing systems and is implemented by PDS (the default CLI—see the *IAS PDS User's Guide*). However, you can issue MCR commands by entering MCR mode at a PDS terminal.

If you are more familiar with the MCR command conventions, MCR mode allows you to continue using these commands when communicating with IAS. When you use this facility, you see little difference between MCR mode and MCR; these differences are detailed in Sections 5.5 and 5.6.

5.2 Entering MCR Mode

If you want to use MCR mode, you must have the appropriate privilege (PR.MCR) assigned by the system manager. If the default mode is MCR and you do not have PR.MCR privilege, DCL mode is set for the terminal when you log in.

To enter MCR mode (when it is not the default), issue the following MCR command:

```
PDS> MCR
PDS>>
```

The prompt PDS>> indicates that this terminal will now accept MCR commands (that is, it is in MCR mode and is awaiting input).

You can issue an MCR command string without entering MCR mode as follows:

```
PDS> MACRO/CROSS/LI:MYPROG MYPROG
PDS> MCR TKB @MYBUILD.CMD
PDS> RUN MYPROG
```

In this example, MYPROG is assembled, then built (linked) with a TKB command file (an MCR command), then run. MCR mode is not entered during this sequence of commands, but the command TKB @MYBUILD.CMD is passed as an MCR command.

If a command is entered as:

```
MCR <command>
```

in MCR mode, MCR at the beginning of the line is ignored.

5.3 Leaving MCR Mode

To leave MCR mode and return to DCL mode, issue the following command:

```
PDS>> DCL
PDS>
```

NOTE: A single DCL command cannot be issued in MCR mode. You must return to DCL mode (see Section 5.4).

5.4 Combined MCR and DCL Commands

When you issue commands at a terminal in MCR mode, most DCL commands are also accepted. When you type a command, PDS attempts to process the command as an MCR command. If this fails it is taken as a DCL command or $$$ command, (see the *IAS PDS User's Guide*). In MCR mode, you can use a combination of MCR commands and DCL commands. However, you must take care with any commands of the same name (for example, SET, MAC, and DIS (DISMOUNT and DISABLE)):

```
PDS> MCR
PDS>> MAC MYPROG,MYPROG/-SP/CR=MYPROG
PDS>> MOU DK0:MYDISK
PDS>> SHOW STATUS
PDS>> LIBR REP DK0:[200,20]BIGLIB
FILE? MYPROG
PDS>> DCL
PDS>
```

The above example uses a combination of MCR and DCL commands to assemble MYPROG (MCR), mount a disk (MCR), show users status (DCL), and replace a module in a library on the disk (DCL).

NOTE: The action to be taken on error, as specified in an ON command, is always interpreted in DCL mode.

5.5 Task Control

The MCR facility to run a number of tasks simultaneously from a terminal is not supported in MCR mode, except on a multiuser or real-time system. For information on the different systems available under IAS, see the *IAS System Management Guide*.

If you type Ctrl/C when a task is running on a timesharing system, the task is suspended. Normally only the ABORT, CONTINUE, and other noninterruptable commands are enabled, but PDS can be built to enable other tasks to run at this time. (See the *IAS PDS User's Guide*.)

NOTE: All MCR commands are treated as interruptable except the INS, MOU and DMO commands (see Section 5.6).

If you type Ctrl/C when a task is running on a multiuser system, the task continues to run. However, the system displays the prompts (PDS>>) so that you can enter further commands. The task is still active (that is, not suspended as it is on a timesharing system). If at this stage the task terminates for any reason, PDS does not recognize the termination until you enter CONTINUE or ABORT, which forces it to check the status of the task.

Therefore, when you use PDS to run a task on a multiuser system, then type Ctrl/C, you must eventually enter either ABORT or CONTINUE to enable PDS to deal with the task (even if the task has already terminated). For example:

```
PDS>> PIP     Run the task PIP
PIP> Ctrl/C       (PIP is running but Ctrl/C reinvokes PDS)

PDS>> RET     (PDS prompts again to allow further
              commands to be entered)

PIP> RET      (PIP is still running)
PIP> Ctrl/Z       (PIP has now finished executing,
PDS>>         but PDS does not recognize this until
              you enter either ABORT or CONTINUE.

PDS>> ABO RET or CONT RET
              (PDS checks the status of the task PIP
              and takes appropriate action)
```

5.6 Differences Between MCR and MCR Mode Commands

Although you see little difference between MCR and MCR mode, some differences do exist. These are described in the following sections.

5.6.1 Logging out of the System

The MCR command BYE and DCL command LOGO Both have the same effect. These commands log you out and perform all functions associated with the logout process. For example:

```
PDS>> LOGO

USER: LEN UIC [200,60] TT03: 17:02:25 6-AUG-78

CONNECT TIME 55 M SYSTEM UTILIZATION 11 MCTS

BYE
```

or:

```
PDS>> BYE

USER LEN UIC [200,60] TT03: 17:02:25 6-AUG-78

CONNECT TIME 55 M SYSTEM UTILIZATION 11 MCTS
```

On a timesharing system, logging out dismounts and deallocates any devices mounted or allocated to you, and deassigns any LUNs.

On a multiuser system, devices are not dismounted on logout.

5.6.2 Batch Execution

When you run batch jobs, you create a batch command stream file with $JOB and $EOJ commands. See the *IAS PDS User's Guide*.

A batch job normally runs in DCL mode, or MCR mode if this has been made the default. You can override this by using the qualifiers /MCR and /DCL to the $JOB command, which force the job to run in the corresponding mode. The $MCR and $DCL commands are also available and are used in the same way as in an interactive job.

- Example 1:

```
$JOB/MCR JONDOE JOBA 300      JONDOE must have
$MAC MYJOB=MYJOB              PR.MCR privileges.
$ON ERROR PRINT THISTSK.MAP
$TKB TASK,THISTSK/-SP=MYJOB
$MCR UTL/SH                   Note--MCR single command
$EOJ                          does not affect the mode. MCR
                              is simply ignored.
```

- Example 2:

```
$JOB/DCL RICROE JOBB 300
$MACRO MYJOB
$ON ERROR PRINT THISTSK.MAP
$LINK/TASK:TASK/MAP:THISTSK.MAP MYJOB
$MCR UTL/SH                  If user RICROE does not have
$EOJ                         PR.MCR privilege, the MCR UTL/SH
                             command fails and
                             THISTSK.MAP is printed.
```

5.6.3 Installing Tasks

The INS command in MCR mode is identical to the MCR INS command, except that it cannot be interrupted by Ctrl/C. However, you can issue the INS command when a task has been suspended (on a timesharing system) or interrupted (on a multiuser system).

```
PDS>> TKB @MYBUILD
Ctrl/C
TASK SUSPENDED
PDS>> INS [11,1]TASK1
PDS>> CONTINUE
```

5.6.4 Task Control

The MCR commands ABO, CON, and RUN (which abort, continue, and run a real-time task) conflict with the DCL commands ABORT, CONTINUE, and RUN. In MCR mode the commands ART, CRT, and RRT respectively replace these commands. The command syntax is identical in each case to the corresponding MCR command, see the command specifications in Chapters 6 and Chapter 7. For example:

```
PDS>> RRT DTASK/MEM
PDS>> ART DTASK
```

If memory is available, DTASK is run as a realtime task and aborts when you issue ART DTASK.

```
PDS>> RRT CTASK 12:00:00
PDS>> CRT CTASK
```

CTASK will be run at 12:00 and will then issue a suspend option to MO to suspend itself; it will be continued after CRT CTASK.

If you issue the command ABO, CON, and RUN in MCR mode, they are passed as DCL commands and have the appropriate effect. That is, they abort or continue the latest subtask activated on the terminal, or run a task whose task image is held in the specified file. For example:

```
PDS>> RUN [200,200]ATASK  (Run the task whose task
Ctrl/C          image is held in [200,000]ASTASK,
                     task name to be assigned as
                     JOBxxx)
TASK SUSPENDED

PDS>> RUN[200,200]BTC
TASK SUSPENDED
```

```
PDS>> ABO                 (Will abort BTASK)
TASK ABORTED
10:30:00 SIZE:4K CPU:008
PDS>> CONTINUE            (Will continue ATASK)
```

NOTE: The above example assumes that you are able to run two suspendable subtasks at the same time. This is a build option to PDS and also requires a privilege to be granted by the system manager.

5.6.5 Mounting and Dismounting Devices

MCR mode supports the MCR commands MOU and DMO. If the system has been generated as a timesharing system then IAS applies device management checks and restrictions. See the *IAS PDS User's Guide.* Normally, users need not intervene when mounting volumes in MCR mode. That is, users need not issue a load request at the system console. However, PDS can be built with an option such that users do need to intervene. For example:

```
PDS> MCR MOU DK0:/OVR
PDS> MCR
PDS>> PIP DK0:/TB
PDS>> DMO DK0
PDS>>
```

MCR mode does not support mounting multivolume magtapes. You must return to DCL mode if you require this function. For example:

```
PDS>> DCL
PDS> MOUNT/DEV:3 MT: (VOLA,VOLB,VOLC)
PDS> MCR
PDS>>
```

5.6.6 Mode Control

The MCR mode control commands are also valid commands in MCR mode.

- Example 1:

  ```
  PDS>> MCR <command line>
  ```

  This facility is most useful when running in batch; that is, MCR batch jobs need minimal change (a new $JOB and $EOJ).

- Example 2:

  ```
  PDS>> MCR
  ```

  This causes the message ALREADY IN MCR MODE to be printed and a WARNING STATUS to be returned, but otherwise it is ignored.

- Example 3:

  ```
  PDS>> DCL
  ```

  To return to DCL mode.

5.6.7 Privileged Users

In MCR mode, a privileged user has a UIC group code of less than 10 octal or has PR.RTC privilege.

5.6.8 Changing Terminal Defaults

Do not use the SET /UIC command in MCR mode, as the effect is unpredictable. See the SET command in Chapter 7.

6 General User Commands

6.1 Introduction

The general user commands listed below are available to all MCR users. In this chapter, commands are listed in alphabetical order.

ABO
ACT
BYE
CON
DCL
DEM
DMO
HEL
LOG
LUN
MCR
MOU
OPR
PWD
QUE
RES
RUN
SYS
TER
TIM
WHO

6.2 MCR Command Descriptions

The layout of each command description is as follows:

1. Function—Describes the function of the command.
2. Format—Supplies the correct format, together with a description of the command parameters and switches.
3. Command Variations—Details any variations in the use of a command between MCR and MCR mode.
4. Technical Notes—Lists any additional information needed to assist in implementing the command.
5. Examples—Supplies typical working examples and an explanation (if necessary).

ABO—Abort

FUNCTION

The ABO (ABORT) command enables you to terminate the execution of tasks that you initiate at your terminal.

FORMAT

MCR>ABO[RT] *taskname[,taskname,...]*

parameter descriptions

taskname

Name of the task being aborted.

COMMAND VARIATIONS

The format of the ABO command varies in MCR mode as follows:

```
PDS>> ART taskname [,taskname...]
```

or:

```
PDS> MCR ART taskname [,taskname...]
```

If you enter ABO in MCR mode, it aborts the latest task activated on the terminal (see the *IAS PDS User's Guide*).

EXAMPLE(S)

- Example 1:

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

  The above command termiantes the execution of the tasks MAC and RICK.

- Example 2:

  ```
  PDS>> ART ...TKB,MYPROG
  ```

  The above example aborts tasks ...TKB and MYPROG, which have been activated by RRT or RUN/REALTIME.

ACT—ACTIVE TASK LIST

FUNCTION

The ACT (ACTIVE TASK LIST) command enables you to obtain a list of the tasks active within the system, with information about their status.

FORMAT

MCR> ACT *[taskname][/switch(es)]*

parameter descriptions

taskname

Name of the task whose status information is to be listed. If you do not specify taskname, all tasks with the same TI as the terminal from which the command is issued are listed.

/switch(es)

One or more of the following switch options:

NOTE: If you do not specify a switch, the following information is listed for all active tasks with the same TI as the terminal that issued the command:

- **Task name**
- **Task scheduling type (RT=realtime,TS=timesharing)**
- **Task status**
- **TI device**

Option	Description
/FU	Specifies that a full listing of the named task's status information is to be generated (see Example 7 for an explanation of what is displayed.) Use of this switch has the following restrictions: 1 It can be specified only when taskname is specified. 2 It cannot be specified in conjunction with the /ALL switch.
/SH	Specifies that a shorter version of the /FU listing is required. The following information is listed:

Option	Description
	1 Task name 2 Task status (see Table 6–1) 3 Task scheduling type (RT=realtime, TS=timesharing) 4 TI device 5 Run priority 6 Partition 7 Real memory address 8 Current size of task read/write segment
/ALL	Specifies that every active task in the system is to be listed. Use of this switch has the following restrictions: 1 It cannot be specified in conjunction with the /FU switch. 2 It cannot be specified in conjunction with the /TI=dev switch. The following information is listed about each task: 1 Task name 2 Task status (see Table 6–1) 3 Task scheduling type (RT=realtime, TS=timesharing) 4 TI device
/TI=dev	Specifies that all the tasks of a specific TI are to be listed. dev is the terminal identifier for the specified TI. The following information is listed about each task: 1 Task name 2 Task status (see Table 6–1) 3 Task scheduling type (RT=realtime, TS=timesharing) 4 TI device

Table 6–1 ACT Task Status Characters

Status	Description
LRP	Load request pending
LRQ	Load request queued
LRS	Load request succeeded
LRF	Load request failed
LRG	Task waiting for regions to load
RUN	Task is runnable
AST	AST queued to task
RLA	Reloaded for AST checking
SUS	Task suspended
WND	Suspended waiting for nodes

Table 6–1 (Cont.) ACT Task Status Characters

Status	Description
WSM	Waiting for privileged task semaphore
STP	Stopped by STOP$
ST0	Stopped for event flags 1-16
ST1	Stopped for event flags 17-32
ST2	Stopped for event flags 33-48
ST3	Stopped for event flags 49-64
ST4	Stopped for event flags 1-64
WF0	Waiting for event flags 1-16
WF1	Waiting for event flags 17-32
WF2	Waiting for event flags 33-48
WF3	Waiting for event flags 49-64
WF4	Waiting for event flags 1-64
EXT	Task exited
IR1	I/O Rundown required
IR2	I/O Rundown in progress
IR3	I/O Rundown finished
IR4	I/O Rundown failure
TFF	Execution fault/termination
TNR	Termination notice requested
STN	Suspended until termination notice output
SFC	Suspended for checkpoint (I/O quiescent)
RRQ	Record request queued
RRS	Record request successful
RRF	Record request failed
PAR	Parity error (task suspended)
TSE	Timesharing task exited
TS1	Special state used by scheduler
TS2	Special state used by scheduler
MRL	Checkpointed or swapped. Realtime task waiting for memory, or scheduler-controlled task swapped out.
MRE	Memory required. Task is being fixed or was requested using EXEC$, and the system is trying to find memory for the task.
MRR	Waiting for memory for regions
WDI	Task waiting for directive to complete
DIF	Directive completed
IDL	Special state for null task
WAC	Waiting for accounts write
EX1	Task exit entry after accounts written

Table 6–1 (Cont.) ACT Task Status Characters

Status	Description
MEX	Task marked for extension

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

- Example 1:

```
MCR>ACT
...ACT RUN TS TT10
...PIP WF0 TS TT10
```

- Example 2:

```
MCR>ACT ...PIP/SH
...PIP WF0 TS TT10 001 GEN 00423200 RW SIZE 024000
```

- Example 3:

```
MCR>ACT ...PIP/ALL/SH
...PIP WF0 TS TT06 001 GEN 00750300 CURRENT RW SIZE 024000
```

- Example 4:

```
MCR>ACT /ALL
DB.... WF0 RT TT00
TT.... WF0 RT TT00
LP.... WF0 RT LP00
DS.... WF0 RT TT00
DK.... WF0 RT TT00
F11ACP SUS RT TT00
MO.... WF0 RT TT00
...ACT RUN TS TT10
...CMP WF1 TS TT00
...PIP SUS TS TT10
```

- Example 5:

```
MCR>ACT /TI=TT10
...ACT RUN TS TT10
...PIP WF0 TS TT10
```

- **Example 6:**

```
MCR>ACT /TI=TT00
DB.... WFO RT TT00
TT.... WFO RT TT00
DS.... WFO RT TT00
DK.... WFO RT TT00
F11ACP SUS RT TT00
MO.... WFO RT TT00
```

- **Example 7:**

```
MCR> ACT DK..../FU
DK.... WFO RT TT00 248 GEN 00435600 CURRENT RW SIZE 004000
REGS 174000 000350 003146 111740 000000 000000 000334
EV 1-16 000000 EV 17-32 000000 ATL FLGS 000000 STD FLGS 064200
EV MASKS 000003 000000 000000 101560 MKTM CNT 000 ACT VERS 001
ATL ADDR 111600 STD ADDR 111540 TSK SIZE INITIAL RW 004000
I/O PEND 000 I/O PROG 000 POOL LIM 040 POOL USE 004 REQ TASK ...LOA
REGIONS HNDLIB
HW PARS 004362 000000 002143 000000 000722 001122 001322 177600
HW PDRS 015406 000000 047002 000000 000016 077406 077406 077406
```

BYE

FUNCTION

The BYE command provides you with a facility for logging out. The only command MCR recognizes after the BYE command is the HEL command. The BYE command has no effect on active tasks previously initiated from a terminal.

FORMAT

MCR>BYE

COMMAND VARIATIONS

In MCR mode, the BYE command logs out the PDS terminal. This is accepted only if no tasks are currently suspended (timesharing system) or active (multiuser system) on the terminal. All devices allocated or mounted are dismounted on a full timesharing system.

The function of the BYE command is identical to the PDS LOGOUT command. See the LOGOUT command in the *IAS PDS User's Guide* for further information.

EXAMPLE(S)

- Example 1:

```
PDS>> BYE

*** DK0 DISMOUNT COMPLETE * * *
USER HANK UIC [200,60] TT10: 13:59:02 15-AUG-78
CONNECT TIME 08M SYSTEM UTILIZATION 23 MCTS
BYE
```

- Example 2:

```
MCR>BYE
```

CON—CONTINUE

FUNCTION

The CON(CONTINUE) command continues the execution of a task that has selected the SUSPEND option when using the message output handler task.

FORMAT

MCR>CON[TINUE] *taskname[,taskname,...]*

COMMAND VARIATIONS

In MCR mode the command format is:

```
PDS>> CRT taskname[,taskname,...]
```

If you use CON in MCR mode, it causes the last task interrupted to continue. See the *IAS PDS User's Guide*.

EXAMPLE(S)

```
MCR>CON XKE
MCR>CONTINUE XKE,NK111
```

In the examples above, the respective requests are to:

1 Continue task XKE.

2 Continue task XKE and task NK111.

DCL

FUNCTION

The DCL command enables you to return from MCR mode to DCL mode when running on a PDS terminal. See Chapter 5 for a description of MCR mode.

FORMAT

PDS> MCR

PDS>> DCL

PDS>

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

```
PDS> MCR
PDS>> PIP/FR
DB0: HAS 28931. BLOCKS FREE, 142867 USED OUT OF 171798
LARGEST CONTIGUOUS SPACE = 2771. BLOCKS
PDS>> DCL
PDS>
```

DEM—DISPLAY MEMORY

FUNCTION

The DEM (DISPLAY MEMORY) command displays memory and task activity.

FORMAT

MCR prompt:

MCR>DEM[O] *[start]*

No MCR prompt:

B[ASE] *base*

parameter descriptions

start

Beginning of the area of memory whose activity is to be displayed. Start is entered either in the form:

```
mK-words
```

or in the form:

```
n (that is, n(octal) blocks of 32 words or 100
(octal) bytes).
```

If start is omitted, zero is used as the starting address, and all of the memory is displayed.

Each column of the display refers to a 1K-word area.

base

Beginning of the area of memory whose activity is to be displayed. base is entered either in the form:

```
mK (that is, mK words (m decimal))
```

or in the form:

```
n (that is, n(octal) blocks of 32 words or 100
(octal) bytes)
```

The following commands can then be entered:

Command	Description
G[RAIN] grain	Reset the amount of memory referred to by a single column of the display. Grain is entered as mK words (m is decimal), or as n (that is, n(octal) blocks of 32 words or 100(octal) bytes).
C[LEAR]	Clear the VDU screen and redisplay. Used, for example, to clear an external message from the screen.
E[XTENT] extent	Change size of memory area displayed. Enter extent either as mk words (m is decimal) or as n (that is, n(octal) blocks of 32 words or 100(octal) bytes).
E[XTENT] ALL	Display all memory (initial state), started at address 0.
I[NTERVAL] n	Update display every n seconds (initially n = 1). n = 0 gives continuous update.
X	Exit to MCR (or PDS).
<	Exit to MCR (or PDS).

When you next use MCR to run the DEM command, the areas displayed return to the values determined by Format 1.

Commands in Format 2 are identified by the first letter. An illegal command or a command containing an error is displayed on the third line of the screen but is otherwise ignored.

COMMAND VARIATIONS

Not applicable.

TECHNICAL NOTES

1 The DEM command displays on a VDU terminal (VT05, VT50, VT52, VT55, VT61, VT100) the memory and task activity of the system.

The display appears in two rows of columns (one row only on a VT50). Each column refers to a portion of memory.

All types of task area within the occupied memory are displayed by task name. Shareable global areas are displayed by name.

Tasks listed down the right hand side of the screen are real-time tasks waiting for memory to become available. The number of nodes available and the largest hole are included in the heading information at the top of the screen.

On the display, at the bottom or top of the columns, the following symbols can appear:

<–>	=	Task's read/write (impure) area
[–]	=	Inactive fixed task
<=>	=	Task's read-only (pure) area

[=] = Shareable global area (SGA) or dynamic region

<+> = Fixed or noncheckpointable task

Once the memory diagram is displayed, the portion of memory being displayed can be altered dynamically by one of the commands in Format 2.

NOTE: Do not type Ctrl/C or use the control key with these commands.

2 DEM is a multiuser task, so more than one copy can be run.

3 Output is on LUN1 that is assigned to TI.

4 DEM can be rebuilt to expand the internal buffer to accommodate more tasks. Information about rebuilding is in the build file. The build file is in [11,13]DEMOBLD.CMD.

5 DEM can be rebuilt to change the default time interval between snapshots of task activity. Information for rebuilding is in the build file.

6 DEM normally uses the "get characteristics" facility of the terminal handler to determine the type of terminal where it is running. See the *IAS Device Handlers Reference Manual*. If the "get characteristics" facility was not specified when the handler was built, DEM must be built for a particular type of terminal. See the DEM build file for details.

7 The amount of memory displayed by default can be changed by editing.

8 If DEM is run with I 0, especially on a 9600 baud terminal, the speed of system performance will be reduced. This occurs to a greater degree if DEM is run as a real-time task; the more terminals running, the greater the speed reduction.

EXAMPLE(S)

- Example 1:

```
MCR>DEM
```

Display all memory activity.

- Example 2:

```
MCR>DEM 32K
```

or:

```
MCR>DEM 2000
```

Display activity in memory area 32K words to top of memory.

DMO—DISMOUNT VOLUME

FUNCTION

The DMO (DISMOUNT VOLUME) command enables you to dismount a previously mounted volume. The DMO command declares the volume to be off-line, thereby rendering it inaccessible.

Before shutting down the system, dismount all volumes to ensure that no files are currently being accessed. Any files currently being accessed when the system is shut down might contain inconsistent data.

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

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

FORMAT

MCR>DMO *dev:[volumelabel][/UIC=[uic]][/LOCK][/NOUNL]*

parameter descriptions

dev:
Device specification for the device where the volume is mounted.

volumelabel
Label of the volume being dismounted. If specified, the label entered is compared with the corresponding label in the Volume Control Block (VCB). If the labels are different, the volume is not dismounted.

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

/LOCK
Option that disables further access and actually dismounts the volume when all current file accesses end. If you specify this option, the volume is marked for dismount.

/NOUNL
For tape: prevents ANSI-mounted tapes from being unloaded after dismount.

For disk: leaves MSCP disks online to controller after dismount.

COMMAND VARIATIONS

1 The command format in MCR mode is identical to that of the MCR DMO command. However, when you issue DMO on a full timesharing system all relevant IAS device management functions are performed.

See the *IAS PDS User's Guide*.

2 The /LOCK qualifier is not valid on a timesharing system.

3 Multivolume magnetic tapes should not be dismounted in MCR mode. Although the volumes will be dismounted, the devices will not be freed for further use by timesharing users. To dismount a multivolume magnetic tape file on a timesharing system, you should first reenter DCL mode as follows:

```
PDS>> DCL
```

Then perform the dismount via the PDS DISMOUNT command.

TECHNICAL NOTES

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

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

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

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

EXAMPLE(S)

- Example 1:

```
MCR>DMO DK1:
```

In this example, the volume mounted on DK1: becomes inaccessible.

- Example 2:

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

In this example, the file system checks to see whether the volume mounted on DK1: has the volume label RICKSVOL and a UIC of [300,300]. If so, the volume is dismounted. Otherwise, the command has no effect.

- Example 3:

```
MCR>DMO MT1:VOLUM1
```

In this example, the volume mounted on MT1: has been interrogated. If it contains a volume label of VOLUM1 it is dismounted otherwise the volume remains mounted.

HEL—HELLO

FUNCTION

The HEL (HELLO) command enables you to log into a terminal and be identified as a valid user by the system. The command also identifies you as a general or privileged user.

FORMAT

MCR>HEL[LO] *[uic]*

COMMAND VARIATIONS

In MCR mode on a timesharing system, the HEL command is taken to be the DCL command HEL[P] (see the *IAS PDS User's Guide* for details of the HELP command). You should issue the DCL LOGIN command, described in the *IAS PDS User's Guide*.

TECHNICAL NOTES

1 If your [uic] does not have a corresponding UFD on the system device (SY:), system access is denied and the following message is issued:

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

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

```
PASSWORD>
```

You must enter the correct password immediately following the prompt. For security, your user-entered password is not echoed when typed in.

3 If there is an error in your password system access is denied and the following message is issued:

```
HEL -- ILLEGAL PASSWORD
```

EXAMPLE(S)

- Example 1:

  The log-in procedure for a non-password-protected UIC is:

  ```
  MCR>HEL [200,200]
  MCR>
  ```

- Example 2:

  The log-in procedure for a password-protected UIC is:

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

- Example 3:

  The log-in procedure in MCR mode is:

  ```
  PDS>> LOGIN username password
  PDS>>
  ```

  Your user name and password are assigned by the system manager.

LOG

FUNCTION

The LOG command enables you to type a comment on the terminal. You can use this feature to note information about the current system. Data typed following this command has no effect on the system.

FORMAT

MCR>LOG *comment-line*

or:

MCR>; *comment-line*

or:

MCR>! *comment-line*

A log message can fill an entire line.

COMMAND VARIATIONS

The LOG command is illegal in MCR mode. Instead, you should use the (!) character.

EXAMPLE(S)

- Example 1:

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

- Example 2:

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

- Example 3:

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

LUN—LOGICAL UNIT NUMBERS

FUNCTION

The LUN (LOGICAL UNIT NUMBERS) command lists on your terminal the physical device units and corresponding LUNs for an indicated task or tasks. You use the LUN command to determine which physical devices a task requires.

FORMAT

MCR>LUN[S] *taskname[,taskname,...]*

COMMAND VARIATIONS

Not applicable.

TECHNICAL NOTES

The LUN command lists the following information:

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

2 Only the devices to which assignments have been made.

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

EXAMPLE(S)

```
MCR>LUN ...SYS
**** ...SYS
SY0 1,4
TI0 2,3,5
CL0 6
OV0 7
```

MCR

FUNCTION

The MCR command enables you to enter MCR mode when running on a PDS terminal. See Chapter 5 for a description of MCR mode.

FORMAT

<u>**Format 1:**</u>

PDS>> *[MCR command string]*

Remains in MCR mode until the DCL command is issued.

<u>**Format 2:**</u>

PDS> [$] *[MCR command string]*

parameter descriptions

MCR command string

Valid MCR command of 79 characters or less. This format issues a single MCR command and returns you to PDS mode.

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

- Example 1:

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

- Example 2:

```
PDS> MCR CRT MYTASK
PDS>
```

MOU—MOUNT

FUNCTION

The MOU (MOUNT) command enables you to make a selected volume accessible by the system. The MOU command declares that the volume is on-line for access by the file control primitives.

FORMAT

General Format:

```
MCR>MOU[NT] \ dev:volumelabel[/keyword(s)])
```

Multivolume magnetic tape format

MCR>MOU[NT] *dd(n1[,n2...,nn]):(label1[,label2,...,labeln]) [/keyword(s)]*

parameter descriptions

dev:

Device specification for the device that contains the volume being mounted.

volumelabel

Volume label of the volume being mounted. The label specified is compared to the label field in the volume's home block, to ensure that the physically mounted volume is the one to be mounted by the system.

If a foreign volume is to be mounted (/CHA=[FOR]), the volume label is not checked and need not be specified.

/keyword(s)

Optional modifiers that enable you to override the volume characteristics that were assigned to the volume when it was initialized. The following keyword options are provided.

The notations listed to the left of the keyword options are as follows:

- * Applies to disk and DECtape only
- = Overrides the corresponding option specified when the volume was initialized (the INI command)

- + Applies to magnetic tape only

Keyword Option	Description
/CHA=[characteristic word]	Changes the device characteristic word. Possible parameters for this option are: *+ FOR A foreign volume (when this parameter is specified, DCF is automatically assumed). ATCH Device can be attached for exclusive use by one task. DCF Device control functions permitted. This qualifier enables tasks to perform read/write logical and positioning operations on a file structured volume. ATCH and DCF are legal for magnetic tape only when it is mounted as Foreign. For example: `/CHA=[FOR,ATCH]`
* /UNL	Specifies that the volume's index file is to be left unlocked, thus giving tasks write access to the index file.
+ /DENS=magnetic tape density	Possible parameters are: 800 for 800 bpi, 1600 for 1600 bpi. The default is 800 bpi. For example: `/DENS=800`
/ACP=taskname	Enables you to designate the task that is to be the file processor for the volume. For example: `/ACP=MTAACP`
*= /EXT=default file extension size in blocks	For further information see the *IAS Performance and Tuning Guide*. For example: `/EXT=1` Default: volume default set when volume is initialized.
*= /FPRO=[default file protection]	Enables you to change the default file protection assigned to new files created on the volume. Each entry consists of from one to four letters with the following meaning: • R—Read access • W—Write access • E—Extend access • D—Delete access The absence of one of these letters in an entry signifies that you are denied that access right. Protection code subparameters (system, owner, group, world) are positional; therefore, the location of the entry in the parameter string defines the user to whom the code applies. For example: `/FPRO=[RWED,RWED,RW,RW]` In this example, group and world are denied Extend and Delete access.

Keyword Option	Description
*= /LRU=number of directories to be kept pre-accessed	The number of pre-accessed directories to be kept is the number of entries in the LRU (least recently used) list for each volume. Pre-accessing directories, that is, entering them in the LRU list, will normally save three disk accesses for each directory operation. Ideally the number of pre-accessed directories should be equal to the number of concurrent users of the volume; however, this will be at the expense of system dynamic memory. For further information see the *IAS Performance and Tuning Guide.* Default equals volume default set when volume is initialized.
+ /OVRFSID	Overrides the file set identifier check. You need this option when processing magnetic tape with inconsistent file set identifiers.
+ /OVREXP	Overrides the expiration date check. The option enables you to overwrite unexpired magnetic tape files.
*= /WIN=n	n is the number of retrieval pointers to be kept in each window block for each open file on this volume. Increasing this number speeds access, particularly to randomly accessed files, at the expense of system dynamic memory. For further information, see the *IAS Performance and Tuning Guide.* Default equals volume default set when volume is initialized.

dd

Device name (for example MT).

n

Logical tape drive number(s) (in the order of selection) representing the drive(s) to be dedicated to the processing of the multi-volume set. If only one drive is being dedicated, the parentheses can be omitted.

label

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

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

/keyword(s)

See the keywords listed in format 1.

COMMAND VARIATIONS

1 In MCR mode the command format is identical. However, mounting volumes in MCR mode on a full timesharing system implies that all IAS device management functions will be applied. Normally this will not include a request to the operator to load the required volume (implied /NOOP), see the IAS PDS User's Guide. However, PDS can be built to include this option.

In this case, on issuing a MOU command a message will be displayed on the system console. The request must be satisfied by the System Operator issuing a SCI>LOAD command after physically loading the correct volume.

2 The facility to mount multivolume magtapes is not available in MCR mode. To perform this function you must first reenter DCL mode:

```
PDS>> DCL
```

Then issue the PDS MOUNT command (see the *IAS PDS User's Guide*).

3 In MCR mode on continuation, the prompt is > not MOU> and continuation (-) can be specified anywhere in the command (see Example 4, below).

TECHNICAL NOTES

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

1 Issue MOU as follows:

```
MCR>MOU
```

This results in the prompt MOU> being displayed (except in MCR mode, see Example 4).

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

EXAMPLE(S)

- Example 1:

```
MCR>MOUNT DK1:SYS004
```

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

- Example 2:

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

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

- **Example 3:**

```
MCR>MOU
MOU>DK1:ACCTSYS/WIN=10.-
MOU>/LRU=4/FPRO=[R,RWED,R,]
```

- **Example 4:**

```
PDS>>MOU DK0:MYDISK/WIN=10.-
>/FPRO=[RWED,RWED,R,]
```

OPR—OPERATE

FUNCTION

The OPR (OPERATE) command enables you to control output spooling.

FORMAT

MCR>OPR[ATE] *dev:/switch[,dev:/switch,...]*

parameter descriptions

dev:

Device specification for the output spooling device being controlled.

/switch

Command modifier that directs OPR operation. OPR switches are described in Table 6–2.

COMMAND VARIATIONS

Not applicable.

TECHNICAL NOTES

Table 6–2 OPR Switches

Switch	Description
/SP	Start printing, on the specified device, all files whose forms type match the forms type currently specified for this device.
/AB	Abort printing of the current file on the specified device. Select the next eligible file for processing. Although the aborted file is removed from the queue, it is not deleted. Therefore it is still available and can be requeued for printing at some later time.
/ST	Stop printing of the current file on the specified device. Printing of the file is stopped in place. Printing can be resumed by specifying one of the /RS switches; or it can be aborted by specifying the /AB switch.
/RS	Resume printing of the current file on the specified device.

Table 6–2 (Cont.) OPR Switches

Switch	Description
/RS:T	Resume printing of the current file from the last encountered form feed. If no previous form feed was encountered, printing begins at the top of the file.
/RS:TOF	Restart printing of the current file from the beginning.

EXAMPLE(S)

1 Example 1:

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

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

2 Example 2:

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

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

PWD—PASSWORD

FUNCTION

The PWD (PASSWORD) command enables you to change or create a password. You must be logged in and can change only your own password.

FORMAT

MCR>PWD *[ufd]*

PASSWORD>

parameter descriptions

[ufd]

PASSWORD> is followed by the new password (not echoed on the terminal).

NOTE: Only the UFD that corresponds to the UIC under which you logged in can be changed.

COMMAND VARIATIONS

This command is not valid in MCR mode. To change the password you should return to DCL mode and issue the SET PASSWORD command. See the *IAS PDS User's Guide* for further details.

EXAMPLE(S)

```
MCR>PWD [200,200]
PASSWORD> SECRET (not echoed)
MCR>
```

This creates (or changes) the password associated with the UFD [200,200] to be “SECRET”.

QUE—QUEUE

FUNCTION

The QUE (QUEUE) command enables you to perform the following functions:

1 Print files on a line printer or terminal (see Format 1).

2 List user-related entries in the queue (see Format 2).

3 List all entries in the queue (see Format 3).

4 Kill (delete) user-related entries in the queue (see Format 4).

5 Modify user-related entries in the queue (see Format 5).

6 Submit a file to a batch queue (see Format 6).

FORMAT

Format 1:

MCR>QUE *[dev:=]infile-1[,infile-2,...,infile-n][/switch(es)]*

MCR>QUE *@indirect*

Queue A File For Printing

This function of the QUE command enables you to queue a file or files for printing on a line printei or terminal.

Format 2:

MCR> QUE *[dev:]/LI*

List Elements In Queue (/LI)

The list function (/LI) of the QUE command enables you to obtain a listing of all entries in the queue, queued by the logged in user. The output from this function provides you with information concerning the status of each entry in the queue, as well as the options specified for each entry.

Format 3:

MCR>QUE /AL

List All Elements in Queue (/AL)

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

Format 4:

MCR>QUE /KI *:seq[:seq:...:seq]*

Kill (Delete) Elements in Queue (/KI)

The kill function (/KI) enables you to delete elements from the queue. You do this by specifying the entry's queue sequence number as a parameter to the /KI function. Queue entry sequence numbers can be obtained by executing the QUE command with the /LI function specified.

Any attempt to delete a queued entry with an account number that is not yours results in a privilege violation error message.

Format 5:

MCR>QUE*[dev:=] / MO:seq[:seq...:seq][/switch(es)]*

Modify A Queue Entry (/MO)

The modify function (/MO) enables you to change the option(s) originally specified for a queued entry or entries. You can modify the following queued entry fields by the /MO function: listing device, priority, forms type, number of copies, delete option.

Format 6:

MCR>QUE BA0:=MY JOB.JOB/BA

Submit a File to a Batch Queue (/BA)

The batch function (/BA) enables you to submit a file to a batch queue.

parameter descriptions

Format 1:

dev:

Optional listing device specification; the default is LP0.

infile

File specification for the file(s) being queued for printing. Wildcards can be used. See Table 6–3 for a list of defaults in the QUE command file specifications.

/switch

One or more of the QUE command option switches (see Table 6–4). The QUE command option switches are global; that is, if specified on any file specification they are applied to the entire list of input file specifications. They can be appended to any of the file specifications in a list. However, if a switch is specified repeatedly, the last value specified is used.

@indirect

Indirect command file specification that contains valid QUE commands. The QUE function supports three levels of indirect file.

Format 2:

dev:

Listing device specification in the form ddnn: (that is, device name and number).

If a device is not specified, all files queued by the logged in user for all devices are listed.

If the device name only is specified (for example, TT:) all files queued by the logged in user for that device name are listed.

If both device name and number are specified (for example, TT5:) only the files queued by the logged in user for that particular device are listed.

Format 4:

:seq

Sequence number for the queued entry being deleted. Two formats for specifying sequence numbers are supported: a single defined sequence number (:seq) or a range of sequence numbers (:seql-seqn).

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

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

Any combination of formats can be used in the same /KI command line; however, the maximum number of individual combinations is ten.

Format 5:

dev:

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

:seq

Sequence number of the queue entry being modified. You can enter sequence numbers individuall

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

/switch(es)

Option modification switch(es). Modification switches are described in Table 6–5.

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

If you queue a file for deletion and later modify that file in the queue, the file is automatically preserved unless you specify /DEL when you modify the queue entry.

COMMAND VARIATIONS

Not applicable.

TECHNICAL NOTES

Table 6–3 Defaults in QUE File Specifications

Specification	Default
dev:	SY:
[ufd]	For the first or only file specification, the UFD that corresponds to the UIC where you log in. For further file specifications, the UFD specified or defaulted for the previous file specification.
file name	Must be specified.
.type	.LST
;ver	The latest version for the file.

Table 6–4 QUE Command Optional Switches

Switch	Description	Default
/PR:nnn	Set the priority of output selection to nnn (nnn is a decimal number ranging from 1 to 250).	/PR:50
/FO:n	Set the output forms type to n (n is a decimal number ranging from 0 to 6). Specific form types are defined by the System Manager; for example, form type 0 should correspond to standard line printer forms; form type 2 could correspond to payroll check forms; form type 3 could correspond to standard billing forms.	/FO:0
/CO:nn	Set the number of copies to be printed to nn (nn is a decimal number ranging from 1 to 31).	/CO:1
/CP	Copy onto a spooled device.	/CP
/DE	Delete the input file after it has been spooled.	/-DE
/-HD	Print the file with no banners.	/HD
/AF:[date]time	Specify a time, or a date and time, after which a file in the spooler queue is processed. If this time has passed, the file is queued for immediate processing. The format for time is hh:mm. The format for the date is dd-mmm-yy.	/AF:00:00
/TE	Test for forms alignments. This switch is used to ensure that the desired forms are properly aligned in the receiving device. The priority entry is automatically set to 250, and the forms type entry is set to 7.	None

Table 6–4 (Cont.) QUE Command Optional Switches

Switch	Description	Default
	This option, although documented in this chapter, is normally used by the system manager.	

Table 6–5 /MO Option Switches

Switch	Description	Default
/PR:nnn	Change the priority of the queue entry to nnn (where nnn is a decimal number ranging from 1 to 250).	None
/FO:n	Change the form type to n (where n is a decimal number ranging from 0 to 6).	None
/CO:nn	Change the number of copies to nn (where nn is a decimal number ranging from 1 to 31).	None
/DE	Change the delete/preserve indicator to delete.	/-DE
/AF:[date]time	Change the time after which the file can be processed.	None

EXAMPLE(S)

Format 1, Example 1:

```
MCR>QUE TEST
```

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

Printing device	LP0: (default)
Priority	50 (default)
Forms type	0 (default)
Number of copies	1 (default)
Delete	No (default)
After (time)	00:00 (default)

Format 1, Example 2:

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

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

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

Delete	Yes
After (time)	00:00 (default)

Format 1, Example 3:

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

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

Printing device LP0:	(default)
Priority	250
Forms type	0 (default)
Number of copies	1 (default)
Delete	No (default)
After (time)	00:00 (default)

Format 1, Example 4:

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

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

Printing device	LP0: (default)
Priority	50 (default)
Forms type	0 (default)
Number of copies	1 (default)
Delete	No (default)
After (time)	00:00 (default)

Format 2, Example 1:

```
    MCR>QUE /LI

DEV ACT ACCOUNT   FILE SPECIFICATION        SEQ PRI FO CO  PBCA

LP0  *  [200,200] DP0:[200,200]TEST.LST;6       100  0  4  *
TT5     [200,200] DK1:[200,200]RSXMAC.SML;17  4 240  0  1     *
   AFTER 17:00
TT1     [200,200] DK0:[200,200]TEST.MAC;40    2  20  0  1
TT5     [200,200] DP0:[200,200]QUE.CMD;3      3  50  0  1
```

Format 2, Example 2:

```
    MCR>QUE TT:/LI

DEV ACT ACCOUNT   FILE SPECIFICATION          SEQ PRI FO CO  PBCA

TT5     [200,200] DK1:[200,200]RSXMAC.SML;17   4 240  0  1     *
   AFTER 09:30
TT1     [200,200] DK0:[200,200]TEST.MAC;48     2  20  0  1
TT5     [200,200] DP0:[200,200]QUE.CMD;3       3  50  0  1
```

The headings shown in Format 2, Examples 1 and 2 have the following meaning:

Heading	Meaning
DEV	Device
ACT	Active. An asterisk (*) in this column indicates that the file is currently being printed.
ACCOUNT	Account
FILE SPECIFICATION	File specification
SEQ	Sequence number
PRI	Priority
FO	Forms type
CO	Copies
PBCA	Preserve Banners Concatenated After An asterisk (*) in the P column indicates that the file will be preserved after printing (tl is, it will not be deleted). A letter N in the B column indicates that no banners will be printed. A letter C in the C column indicates that output will be concatenated. An asterisk (*) in the A column indicates that the file will be held in queue until the giv time.

Format 4, Example 1:

```
MCR>QUE /KI:2:6:10:12
```

In this example, the queued entries whose sequence numbers are 2, 6, 10, and 12 are deleted fro the queue.

Format 4, Example 2:

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

In this example, the queued entry 2, the entries 6 through 12, and entry 14 are deleted from th queue.

Format 5, Example 1:

```
MCR>QUE GAYE.RNO;1/DE
MCR>QUE/MO:1/CO:3 GAYE.RNO;1
```

In this example, GAYE.RNO;1 is automatically preserved. To make sure the file is still deleted, type the following:

Format 5, Example 2:

```
MCR>QUE GAYE.RNO;1/DE
MCR>QUE /MO:1/CO:3/DE
```

RES—RESUME

FUNCTION

The RES (RESUME) command enables you to continue execution of a previously suspended task or tasks, suspended by SPND$ (a system directive; see the *IAS System Directives Reference Manual*) or the FORTRAN PAUSE statement (see the *PDP-11 FORTRAN Language Reference Manual*).

FORMAT

MCR>RES[UME] *taskname[,taskname,...]*

parameter descriptions

taskname
Name of the task being resumed.

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

1 Example 1:

```
MCR>RES XKE
```

Resume task XKE.

2 Example 2:

```
MCR>RES XKE,NK111,240Z
```

Resume task XKE, task NK111, and task 240Z.

RUN

FUNCTION

The RUN command enables the user to initiate tasks that have not been installed. The RUN command causes a selected task image to be installed, loaded, executed, and removed upon termination.

FORMAT

MCR>RUN *filename[/TIM=nx] ESC*

or:

MCR>RUN *$filename ESC*

parameter descriptions

filename

File specification for the task image. Table 6–6 contains a list of defaults in task image file specifications.

$filename

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

/TIM=nx

- n = Decimal number indicating the number of time units (x) of central processor time that the task is enableed.
- x = Type of time unit. This can be one of the following:

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

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

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

COMMAND VARIATIONS

1 In MCR mode use the RRT command name.

2 If you issue RUN in MCR mode, the task runs with a taskname generated by the system in the form:

JOBxx (timesharing system), or

TTnnx (multiuser system)

3 The DCL command SET DEFAULT (available on timesharing systems only) establishes a new default device and/or UFD used in file specifications included in subsequent commands.

The new defaults established by this command do not apply to the MCR mode RRT command.

TECHNICAL NOTES

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

The task runs with the UIC where the terminal is logged in.

Table 6–6 Defaults in Task Image File Specifications

Specification	Default
dev:	SY:
[ufd]	UFD that corresponds to the UIC under which you logged in.
filename	Must be specified.
.type	.TSK
;ver	The latest version for the file.

2 When you use the formats RUN $filename or RRT $filename, ensure that the task image file specified in the command line is owned by the UIC under which it resides; (that is, the UIC defined by the DUIC directive at system generation time, or the default of [11,1] if DUIC was not explicitly specified at system generation time).

Alternatively, if the task image file is owned by another UIC, the access rights should change accordingly.

EXAMPLE(S)

1 Example 1:

```
MCR>RUN DK1:RICK.TSK;3 ESC
```

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

2 Example 2:

```
MCR>RUN FRED/TIM=10M
```

This will run the task held in file FRED.TSK and will be terminated after 10 minutes of CPU time if it has not completed execution.

SYS—SYSTEM STATUS

FUNCTION

The SYS (SYSTEM STATUS) command enables you to examine the status of the following lists and queues:

1 Active Task List (ATL)

2 Clock Queue (CKQ)

3 Global Common Directory (GCD)

4 Checkpointable Task List (CTL)

5 I/O Request Queue (IRQ)

6 Fixed Task List (FTL)

7 System Task Directory (STD)

8 Physical Unit Directory (DEV)

9 Task Partition Directory (TPD)

10 SEND/RECEIVE Queue (SRQ)

11 Asynchronous System Trap Queue (ASQ)

For each list, the SYS command prints the list name and the names of all tasks that have entries or are entered in that list. In the case of SRQ and ASQ, the command prints the queue name and the names of all tasks that have entries in that queue. The function and content of system lists and queues are described in the *IAS Executive Facilities Reference Manual*. The command can also print the names of shareable global areas (SGAs) to which tasks in the System Task Directory (STD) are linked and provide the following information:

1 List all installed tasks.

2 List all devices known to the system.

3 List the partitions defined at System Generation.

FORMAT

MCR>SYS *[/switch[:opt]]*

parameter descriptions

switch

Any one of the following switches (/BRF is the default switch):

Switch	Function
/BRF	Lists the names of all tasks in the ATL, MRL and CKQ. /BRF is used by default if no switches are included in the SYS command.
/ATL	Lists the names of tasks in the Active Task List.
/CKQ	Lists the names of tasks in the Clock Queue.
/GCD	Lists the names of libraries and common areas in the Global Common Directory.
/IRQ[:opt]	Lists the names of tasks with entries in the I/O Request Queues. If the IRQ for a particular device is desired, you must include the opt parameter to designate the device. For example, /IRQ:DK0 causes the IRQ associated with the DK handler to be listed.
/CTL[:opt]	Lists the names of tasks in the checkpointable Task List. You use the opt parameter to restrict the output from SYS to a particular partition; for example, /CTL:GEN indicates that the names of tasks running in the GEN partition and contained in the CTL are to be printed.
/FTL	Lists the names of tasks in the Fixed Task List.
/SRQ	Lists the names of tasks that have SENDs queued for them.
/ASQ	Lists the names of tasks in the ATL that have ASTs queued for them.
/SGA	Lists the names of all tasks in the STD that are bound to one or more SGA, and the SGA to which each is bound.
/FUL	Lists all of the above except that which is produced by using /SGA.
/TAS	Lists the names of and the following information for installed tasks: version number, default partition and priority, task size, and fixed and multiuser indicators. SYS /TAS sometimes shows multiple entries for a task, because the System Task Directory (STD) can change while the list of tasks is being displayed.
/DEV	Lists the name and PUD address of each symbolic device known to the system. On the listing, devices for which a device handler is resident are indicated by two asterisks (**). A redirected device is followed by the mnemonic of the device where it was redirected. Spooled and mounted (MOU) devices are indicated by the words SPOOLED and MOUNTED, respectively. The number following a spooled device is its current forms type.
/PAR	Lists the following description of each memory partition in the system: • Name • Base address (octal) • Size (octal) • Partition type; can be: U (user-controlled), S (system-controlled, priority-oriented), T (system-controlled, time-scheduled).

Switch	Function
/COM	Lists the following description of each installed shareable global area: • Name • Base address (octal) • Size • UIC • Access • Position independent or blank • Creation date • Type Codes TPA—task pure area LIB—resident library SGA COM—common area SGA IRG—installed region SGA DRG—dynamic region or task read/write resident overlay region

NOTE: Each scan performed by SYS inhibits task switching for a significant period of time. Therefore, you should not issue SYS /FUL when realtime response is critical.

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

1 Example 1:

```
MCR>SYS /ATL

 ATL:

 DP.... TT.... DK.... LP.... DS.... DT.... ...MCR ...SYS
 MO.... SPR2.. DLLEX
```

2 Example 2:

```
MCR>SYS /CKQ

 CKQ:

 NO ENTRIES
```

3 Example 3:

```
MCR>SYS /IRQ
 IRQ:
 TT0:
 ...PIP ...FLX
 CI0:
 NO ENTRIES
 CO0:
 NO ENTRIES
 CL0:
 NO ENTRIES
 SY1:
 NO ENTRIES
 IP0:
 NO ENTRIES
```

4 Example 4:

```
MCR>SYS /CTL
 CTL:
 ...SYS ...PIP DLLEX
```

5 Example 5:

```
MCR>SYS /CTL:GEN
 CTL:
 ...SYS DLLEX
```

6 Example 6:

```
MCR>SYS /FTL
 FTL:
 F11ACP
```

7 Example 7:

```
MCR>SYS /SRQ
 SRQ:
 NO ENTRIES
```

8 Example 8:

```
MCR>SYS /ASQ
 ASQ:
 NO ENTRIES
```

9 Example 9:

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

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

10 Example 10:

```
MCR>SYS /BRF
  ATL:

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

  MRL:

  NO ENTRIES

  CKQ:

  NO ENTRIES
```

11 Example 11:

```
MCR>SYS /FUL
  ATL:

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

  MRL:

  NO ENTRIES

  CKQ:

  NO ENTRIES

  FTL:

  F11ACP

  GCD:
```

```
SYSRES

CTL:

...SYS ...MAC

SRQ:

NO ENTRIES

ASQ:

NO ENTRIES

IRQ:

TT:

NO ENTRIES

CI:

NO ENTRIES

DK:

NO ENTRIES

DP:

NO ENTRIES
```

12 Example 12:

```
MCR>SYS /DEV
  TT0 141500 **
  CI0 141562 TT0
  CO0 141644 TT0
  CL0 141726 TT0
  SY1 142010 DP0
  IP0 142072
  NT0 142154
  SP0 142236 DS0
  WK0 142320 DS0
  LP1 142402
  XX0 142464
  BP0 142546
  MO0 142630 **
  DT3 142712 **
  DT2 142774 **
  DT1 143056 ** MOUNTED
  DT0 143140 **
  CT1 143222
  CT0 143304
  CR0 14336
  LP0 143450 ** SPOOLED 0
  TT20 143432
  TT17 143614 **
  TT16 143676 **
  TT15 143760 **
  TT14 144042 **
  TT13 144124 **
  TT12 144206 **
  TT11 144270 **
  TT10 144352 **
  TT7 144434 **
  TT6 144516 **
  TT5 144600 **
  TT4 144662 **
```

```
TT3 144744 **
TT2 145026 **
TT1 145110 **
MT1 145172
MT0 145254
DS0 145336 ** MOUNTED
DP1 145420 **
DP0 145502 ** MOUNTED
DK3 145564 **
DK2 145646 **
DK1 145730 **
DK0 146012 **
SY0 146074 **
```

13 Example 13:

```
MCR>SYS /PAR

  SYDISK 113200 003500 U
  MCR 116799 015000 S
  GEN 133700 601100 S
  TTY 735000 014400 U
```

14 Example 14:

```
MCR>SYS /ATL

  ATL:

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

  MO....
```

TER—TERMINAL

FUNCTION

The TER (TERMINAL) command enables you to change the characteristics of your terminal. Terminal characteristics revert to the system defaults when you disconnect a dialup line or when you type BYE or LOGOUT (MCR mode).

For details of the software facilities associated with characteristics, see the *IAS Device Handlers Reference Manual*. For the setting of characteristics at system generation time, see the *IAS Installation and System Generation Guide*.

FORMAT

Format 1:

MCR> TER

Format 2:

MCR> TER */terminaltype [/DS]*

Format 3:

MCR> TER */options*

Format 4:

MCR> TER *@indirect*

parameter descriptions

terminaltype
One of ASR33, ASR35, KSR33, LA120, LA180S, LA30P, LA30S, LA36, VT05, VT50, VT52, VT55, VT61, VT100, VT200, or VT300.

TER /terminaltype
Sets the characteristics other than the speed(s) to the default values listed in the *IAS Device Handlers Reference Manual*. If /DS is appended, the speed also is set to the default value.

/options

One or more of:

/[NO]option
/option:value

Each option and any short form listed with it can be abbreviated as long as it remains unique within the list of TER options. Each acceptable form of an option without a value can be negated by the prefix NO (for example, /NOSCOPE).

An asterisk (*) marks the options that are likely to be most commonly used.

/ALTMODE

Terminal is a teletype that generates 175 or 176 (octal) when the ALT key is pressed. Either of these characters are treated in the same way as (ESCAPE).

/ANSISEQUENCE

Escape sequences for the terminals are to be in ANSI mode.

/BACKSPACE

Terminal responds to the backspace character.

/BINARY

Terminal is to operate in binary mode.

/BLOCKMODE

Terminal is a VT61 to be used in block mode.

**/CARRIAGERETURN or /CR*

Lines exceeding the terminal width as set are continued on the following line(s).

/COMPATIBLE

Terminal requires RSX-11M compatible escape sequence handling.

/CONTROLCFLUSH or /CCF

Flush type-ahead when Ctrl/C is typed.

/CONTROLS or CSQ

When the terminal is in deferred processing mode, the terminal is set so that only Ctrl/S and Ctrl/Q are processed immediately.

/ESCAPESEQUENCE

Terminal requires escape sequence recognition.

**/FORMSMODE*

Terminal is a VT61 and is to be used in forms mode.

/FULLDUPLEX

Invokes full duplex mode, in which input and output operate independently. For use with intelligent terminals. (For details, see the *IAS Device Handlers Reference Manual*).

/HANGUP

Hang up dialup line. This cannot be negated.

/HARDWAREFORMFEED or /HFF

The characters form feed and vertical tab are recognized and do not need software simulation.

/HARDWARETAB or /HTAB

The character horizontal tab is recognized.

*/HOLD

Enters auto-hold mode. Output from the computer is stopped automatically when the screen becomes full with output and can be resumed by pressing the SCROLL key to enable a further line to be output. Pressing the SHIFT and SCROLL keys simultaneously will enable a further page to be output. For this facility to work correctly the terminal must transmit and receive at the same speed (VT5x and VT61 terminals only).

/KEYBOARD

Terminal is capable of input.

/LOCALCOPY

Terminal echoes all characters as they are typed.

*/LOWERCASEKEYBOARD or /LCKEYBOARD

Lower case characters are accepted. If immediate echo or Ctrl/R type-ahead is used, characters are echoed as lower case, whether or not they are processed as lowercase.

*/LOWERCASEINPUT or/LCINPUT

Lower case characters are to be passed to a program performing input even if a program, for example, EDI, asks for case conversion.

*/LOWERCASEOUTPUT or /LCOUTPUT or /LOWERCASEPRINTER or /LCPRINTER

Terminal can print lower case characters.

/LVF

LA36-type vertical fill is required for form feed and vertical tab that is 66 nulls.

/NEWLINE

Terminal sends newline when the carriage return key is pressed.

/NONSTANDARDTAB or /NSTAB

Terminal on receiving tab character does not space to the next 8-character boundary.

/NOPARITY

Terminal handler does not generate parity bit on character output.

/PASSALLBITS

Terminal passes all eight bits of characters read to the user buffer for a READPASSALL request.

/PRINTER

Terminal is capable of output.

/PROCESSCONTROLC

When the terminal is in deferred processing mode the terminal is set so that only Ctrl/S, Ctrl/Q, Ctrl/C are processed immediately.

/SCOPE

Terminal is a visual display unit (VDU) and rubout physically erases characters from the screen.

*/SIMULATEFORMFEED or /SFF

Form feed and vertical tab are to be software simulated to start a new page and skip to next six-line boundary respectively.

*/TAPE

Terminal has a low speed paper tape reader and interprets Ctrl/B and Ctrl/T accordingly.

/TWOSTOPBITS or /TSB

Terminal requires two stop bits as normally required for mechanical printers; for example, ASR33.

/VERTICALFILL or /VFILL

Terminal requires VT05-type vertical fill.

/option:value

Can be any of the following:

Option	Explanation	
/FILL:n	n is fill required for carriage return. n = 7 supplies LA30S-type fill.	
/LENGTH:n	n is page length in lines.	
/NAME:name	name can be one of: ASR33, ASR35, KSR33, LA120, LA180S, LA30P, LA30S, LA36, VT05, VT52, VT55, VT61, VT100. TER/NAME:name is for use in "deceiving" a program as to the type of terminal under which it is running, for example, when mixed characteristics are required. The options sets only the location holding the name of the terminal type, see the *IAS Device Handlers Reference Manual.* **NOTE**: This option does not set the corresponding characteristics implicitly.	
/PARITY:type	type is EVEN or ODD. Set line to generate characters with parity. Note that parity is not checked on input.	
/READAHEAD:type	type is one of:	
	NONE	No read-ahead enableed.
	DEFERREDPROCESSING or DP	Read ahead accepted but not examined until a read that uses it is processed.
	IMMEDIATEPROCESSING or IP	Read ahead is processed as it is typed but not echoed till it is read.
/SPEED:(m:n)	Set split-speed line. m is the keyboard (lower) speed. n is the printer or display (higher) speed.	
/SPEED:n	Set line speed. n can be one of: speed in baud 134 (meaning 134.5 baud) EXTA (DH11 external speed A) EXTB (DH11 external speed B)	
/WIDTH:n	n is the desired width in characters.	
indirect	is the file specification of a file holding the remainder of a TER command.	

With all TER formats, you can reply TER to the MCR prompt. TER then prompts for further input.

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

1 Example 1:

```
MCR>TER /CR/WIDTH:50/LENGTH:30
```

The width is set to 50 characters, and the page length to 30 lines. Lines of more than 50 characters are continued on the following lines.

2 Example 2:

```
MCR>TER /SPEED:(150:9600)
```

Terminal is to send a 150 baud and receive at 9600 baud.

3 Example 3:

```
MCR>TER /VT05/DS
```

Terminal is a VT05 and is to run at the corresponding speed (2400 baud).

4 Example 4:

```
MCR>TER /NAME:VT61
```

The terminal type is recorded as being VT61 but no characteristics are thereby changed.

TIM—TIME

FUNCTION

The TIM (TIME) command enables you to list the time and date on your terminal.

FORMAT

MCR>TIM

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

```
MCR>TIM
14-NOV-78 13:45:00
```

WHO—TERMINAL STATUS

FUNCTION

The WHO (TERMINAL STATUS) command indicates which terminals are in use, provides your UIC, and indicates whether each terminal is privileged or slave. The WHO command also indicates those terminals that were defined during System Generation but that are not currently in use.

FORMAT

MCR>WHO

COMMAND VARIATIONS

The DCL SHOW command can also be used when in MCR mode to give more detailed information. See the *IAS PDS User's Guide* for further details.

EXAMPLE(S)

```
MCR>WHO

TT0 [1,1] PRIVILEGED
TT3 NOT LOGGED ON
TT2 NOT LOGGED ON
TT1 [101,30]
```

7 Privileged User Commands

7.1 Introduction

This chapter describes privileged user commands. Under MCR, a privileged user is anyone logged in with a UIC group code less than 10 (octal). In MCR mode, a privileged user is anyone who has PR.RTC privileges, and whose associated UIC has a group code less than 10 (octal).

Some privileged user commands appear to duplicate nonprivileged commands. These are actually nonprivileged commands with enhanced capabilities for the privileged user.

A list of the privileged user commands follows:

ABO
ALT
BOO
CAN
DIS
DMO
ENA
FIX
HOM
INI
INS
LOA
MEM
MOU
OPE
OPR
PWD
QUE
REA
RED
REM
RES
RUN
SAV
SET
SWA
TER
TIM
UFD
UNL
UNF
UTL

Commands are presented in alphabetical order.

7.2 MCR Privileged Command Descriptions

The layout of each command description is as follows:

1 Function—Describes the function of the command.

2 Format—Supplies the correct command format, together with a description of the command parameters and switches.

3 Command Variations—Details any variations in the use of a command between MCR and PDS mode.

4 Technical Notes—Lists any additional information needed to assist in using the command.

5 Examples—Supplies typical working examples and an explanation (if necessary).

ABO—ABORT

FUNCTION

The ABO (ABORT) command enables you to terminate the execution of a specified task.

FORMAT

MCR>ABO[RT] *taskname[/TI=dev]*

parameter descriptions

taskname

Name of the task being aborted.

/TI=dev

Optional switch (appended to the taskname) that enables you to abort any task in the system, by specifying its TI. dev is the device mnemonic for the terminal corresponding to the TI under which the specified task is running. If the requesting task was initiated at the requesting terminal, the /TI switch is not required.

COMMAND VARIATIONS

The format in MCR mode is:

```
PDS>>ART taskname[/TI=dev]
```

In MCR mode, you can only use the ABO command to abort the task running as a result of a previous command.

EXAMPLE(S)

- Example 1:

```
MCR>ABO SCAN
```

- Example 2:

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

ALT—ALTER PRIORITY

FUNCTION

The ALT (ALTER PRIORITY) command enables you to alter the priority of an active task or tasks.

FORMAT

MCR>ALT[ER] *task[/TI=dev]/PRI=nnn[,task[/TI=dev]/PRI=nnn,...]*

parameter descriptions

taskname
Name of the task whose priority is to be altered.

/TI=dev
Optional switch (appended to the taskname) that enables you to alter the priority of any task in the system by specifying its TI. dev is the device mnemonic for the terminal corresponding to the TI under which the task runs. If the requesting task was initiated at the requesting terminal, or if the designated task is not a multi-user task, you will not require the TI switch.

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

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

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

BOO—BOOT

FUNCTION

The BOO (BOOT) command enables you to perform either of the following operations:

1 Stop the present operating system and bootstrap the system specified by the file specification.

2 Copy a bootstrap block onto block 0 of the specified device from a specified system image file on the same device.

FORMAT

Format 1:

MCR>BOO[T]

When you use this format, BOO issues the following prompt:

BOO>

You now enter another RET.

When you have executed the command in this format, BOO causes the latest version of the file IAS.SAV (residing in the default directory file of the system disk) to be bootstrapped into memory.

Format 2:

BOO[T] *filespec*

When you use this format, the BOO command causes the bootstrap routine to be loaded from the file specified by filespec. This bootstrap routine then loads the remainder of the file into memory thus starting up the new system.

Format 3:

MCR>BOO[T] /WB

When you use this format, a bootstrap routine is written on block 0 of the system device. The bootstrap routine is copied from virtual block 1 of the latest version of file IAS.SAV residing in the default directory file on the system device.

Format 4:

MCR>BOO[T] *filespec/WB*

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

parameter descriptions

filespec

File specification for the file that contains, as its first record, the bootstrap routines to be used in re-bootstrapping the operating system. Table 7–1 contains a list of default values for BOO file specification fields.

/WB

Write bootstrap switch.

filespec

File specification for the file that contains, as its first virtual block, the bootstrap routine to be used. Table 7–1 contains a list of default values for BOO file specifications.

Table 7–1 Defaults in BOO File Specification Fields

Specification	Default
dev:	SY:
ufd	The UFD that corresponds to the UIC where the user logged in.
filename	IAS
.type	.SAV
;ver	Latest version

COMMAND VARIATIONS

Not applicable.

TECHNICAL NOTES

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

2 When a rebootstrap is to take place, the currently running system should be *quiescent* (that is, no user tasks should be running).

3 Only as much memory as was specified at system generation time is loaded during the bootstrap operation.

4 The file specified to BOO must have been created at system generation time.

EXAMPLE(S)

- Example 1:

  ```
  MCR>BOO
  ```

  In this example, the system is rebootstrapped with the latest version of the system image file IAS.SAV, which resides in the default directory file on device SY:

- Example 2:

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

  In this example, the system is rebootstrapped with the latest version of the system image file IAS.SAV, which resides in directory file [10,10] on device DK1:

- Example 3:

  ```
  MCR>BOO /WB
  ```

  In this example, a bootstrap routine is copied from virtual block 1 of the system image file IAS.SAV, which resides in the default directory file on SY:, to block 0 of device SY:.

- Example 4:

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

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

CAN—CANCEL

FUNCTION

The CAN (CANCEL) command enables you to cancel periodic rescheduling of tasks.

FORMAT

MCR>CAN[CEL] *taskname[/TI=dev] [,taskname[/TI=dev],...]*

parameter descriptions

taskname
Name of the task being cancelled.

/TI=dev
Optional switch (appended to the task name) that enables you to apply the cancel to a particular terminal. dev is the device mnemonic for the terminal corresponding to the TI under which the task runs. If the requesting task was initiated at the requesting terminal or if the designated task is not a multiuser task, you do not need the /TI switch.

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

- Example 1:

  ```
  MCR>CANCEL XKE
  ```

  Cancels all scheduled requests for task XKE.

- Example 2:

  ```
  MCR>CANCEL XKE,NK111
  ```

  Cancels all scheduled requests for tasks XKE and NK111.

- Exaple 3:

  ```
  MCR>CANCEL XKE/TI=TT1
  ```

  Cancels all scheduled requests for task XKE run from terminal TT1.

DIS—DISABLE

FUNCTION

The DIS (DISABLE) command enables you to inhibit task execution without removing the task from the system. Disabled tasks cannot be initiated until they are enabled through the ENA command issued by a privileged user.

FORMAT

MCR>DIS[ABLE] *taskname[,taskname,...]*

parameter descriptions

taskname

Name of the task being disabled.

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

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

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

DMO—DISMOUNT VOLUME

FUNCTION

The DMO (DISMOUNT VOLUME) command is functionally the same as for a non-privileged user. See the DMO command description in Chapter 6. The difference is that only a privileged user is enabled to dismount the system volume.

ENA—ENABLE

FUNCTION

The ENA (ENABLE) command reverses the effects of the DIS command.

FORMAT

MCR>ENA[BLE] *taskname[,taskname,...]*

parameter descriptions

taskname
Name of the task being enabled.

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

- Example 1:

```
MCR>ENA SCAN
```

- Example 2:

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

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

FIX—FIX IN MEMORY

FUNCTION

The FIX (FIX IN MEMORY) command enables you to fix a task in its default partition, that is, to dedicate memory to a task for faster response to requests for execution. You cannot fix a task unless it was built as a fixable, non-checkpointable task. The use of fixed tasks is described in the *IAS Executive Facilities Reference Manual.*

FORMAT

MCR>FIX *taskname[/TI=dev] [,taskname[/TI=dev],...]*

parameter descriptions

taskname

Name of the task being fixed in memory.

/TI=dev

Optional switch that specifies the TI under which the task is to be fixed. dev is the device mnemonic for the terminal corresponding to the TI under which the task is to be run.

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

- Example 1:

```
MCR>FIX XKE
```

- Example 2:

```
MCR>FIX JAG,24OZ
```

In the examples above, the tasks XKE, JAG, and 24OZ are fixed in their default partitions.

- Example 3:

An SGA can be fixed in memory by writing a dummy task that uses that SGA. The dummy task should be built with the following options, which are described in the *IAS Task Builder Reference Manual*.

```
/-CP/-FP/FX
STACK = 10
UNITS = 0
SGA=sga name
```

Example of dummy task:

```
  .MCALL EXIT$S
START:
  EXIT$S  ; IN CASE TASK IS EXECUTED
  .END START
```

The dummy task uses 128 words; this includes the task header and minimum stack size, plus one fixed tasked list entry.

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

HOM—HOME

FUNCTION

The HOM (HOME) command enables you to modify certain fields in the Home Block of an existing Files-11 volume. It performs selected functions of the INI command, without volume re-initialization.

FORMAT

General Format:

MCR>HOM *ddn:volume-label[/switches]*

parameter descriptions

ddn:

Device specification where the volume is loaded.

volume-label

Current volume label. Volume labels can be from 1 to 12 alphanumeric characters long.

/switches

Fields to be modified. The following switches are provided:

Switch	Description
/UIC=[m,n]	m is group code and n is programmer code. This switch specifies volume owner.
/PRO=[volume protection]	Volume protection is of the form [system, owner, group, world]. This switch specifies the volume protection. For further details see Technical Note 1.
/MXF=n	n is the maximum number of files enabled on this volume. The value of n cannot be specified as smaller than the current value or greater than the value corresponding to the last bit in the existing index file bitmap. See *Tab7_2* for details of the default number of files for each type of volume.
/EXT=n	n is the default file extension size in blocks. For further information see the *IAS Performance and Tuning Guide.*
/FPRO=[default file protection]	This switch specifies the default file protection for files created on the volume. To enter this parameter, use the same format as the /PRO switch. (See Technical Note 1.)

Switch	Description
/CHA=[param 1, param 2]	This switch specifies the volume characteristics. The possible parameters for this switch are as follows: • ATCH–Device can be attached for exclusive use by one task. • DCF–Device control functions permitted; for example, read/write logical. For example: `/CHA=[ATCH,DCF]`
/WIN=n	n is the default number of window pointers available for file access to this volume. This value represents the default number of retrieval pointers that are maintained in memory for each open file. For further information, see the *IAS Performance and Tuning Guide.*
/LRU=n	n is the number of preaccessed directories to be kept while the volume is in use; that is, the number of entries in the LRU (least recently used) list for each volume. Preaccessing directories by entering them in the LRU normally saves three disk accesses for every directory operation. Ideally, n should be equal to the expected number of concurrent users of the volume; however, this will be at the expense of system dynamic memory. For further information, see the *IAS Performance and Tuning Guide.*
/DENS=density	Density is HIGH or LOW. /DENS checks the density of a floppy disk in an RX02 drive. If /DENS is specified, the drive density is compared against the specified density for consistency. Single (or RX01-compatible) density is LOW. Double density is HIGH.
/VI	Shows the current values of all the switches.
/NAME=volume-label	volume-label specifies a new volume label. This can be 1 to 12 alphanumeric characters.

COMMAND VARIATIONS

Not applicable.

TECHNICAL NOTES

1 Each entry in the /PRO and /FPRO switch consists of from one to four letters with the following meaning:

R—Read access
W—Write access
E—Extend access
D—Delete access

The absence of any of these code letters means that users are denied that right.

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

2 Extensive checks are performed to ensure that a valid home block is written back to the volume. If any of these checks fail, HOM attempts to place an alternate home block as the first free block on the MOUNT search path. If this is successfully completed a warning message will be issued that states that the volume has a replacement home block which is not protected. The volume should be backed up immediately. You should then run the badblocks utility on the volume, and then reinitialize the volume.

3 The volume for which the home block is to be modified should not be MOUNTED. However, on a timesharing system the device on which the volume is loaded should be allocated.

EXAMPLE(S)

```
MCR> HOM DB1:CAROL/VI

** VOLUME INFORMATION FOR DB1: **

LABEL = CAROL
/NAME = CAROL
/CHA = []
/EXT = 5.
/FPRO = [RWED,RWED,RWE,R]
/LRU = 3.
/MXF = 10576.
/PRO = [RWED,RWED,RWED,RWED]
/UIC = [1,1]
/WIN = 7.

MAXIMUM POSSIBLE FILES = 12288.
```

INI—INITIALIZE VOLUME

FUNCTION

The INI (INITIALIZE VOLUME) command enables you to produce new Files-11 structured volumes. Files-11 structured volumes are described in the *IAS/RSX-11 I/O Operations Reference Manual.*

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

FORMAT

General Format:

```
MCR>INI[TVOL] dev:[volumelabel] [/keyword(s)]
```

parameter descriptions

dev:

Device specfication of the device where the volume is to be loaded.

volumelabel

Label to be assigned to the volume. Volume labels can be from 1 to 12 alphanumeric characters long. The only exception is magnetic tape volumes from 1 to 6 alphanumeric characters long. The volume label must be specified for both disk and magnetic tape.

/keyword(s)

Defines the volume characteristics. If no keywords are specified, the default characteristics are used. See the individual keyword description for default characteristics. The following keyword options are provided.

NOTE: An asterisk (*) to the left of a keyword option means that keyword option is valid for disk and DECtape only.

Keyword	Explanation
/UIC=[m,n]	m is group code and n is programmer code. The default is [1,1]. This defines the volume owner.
/PRO=[system, owner, group, world]	This keyword enables you to establish volume access privileges. For example: /PRO=[RWED,RWED,RW,RW] In this example, group and world are denied extend and delete access. For further information see Technical note 2.

Keyword	Explanation
	Default: /PRO=[RWED,RWED,RWED,RWED]
* /MXF=n	n is the maximum number of files enabled on this volume. See Table 7–2 for details of the maximum number of files per volume for each Files-11 device. Default: See Table 7–2 for details of the default allocations.
* /EXT=n	n is the default file extension size in blocks. For further information see the *IAS Performance and Tuning Guide*. Default: /EXT=5
* /FPRO=[default file protection]	To enter this parameter, use the same format as the /PRO keyword. Default: /FPRO=[RWED,RWED,RWE,R]
* /CHA=[param1,param2]	This keyword defines the device characteristics. The possible parameters for this keyword are as follows: • ATCH–Device can be attached for exclusive use by one task. • DCF–Device control functions permitted; for example, read/write logical. For example: /CHA=[ATCH,DCF] Default: No ATCH and no DCF
* /INF=n	n is the number of file headers to allocate in the initial index file. Default: See Table 7–2 for details of the default allocations.
* /WIN=n	n is the default window size for file access to this volume. This value represents the number of retrieval pointers that will be maintained in memory for each open file. For further information see the *IAS Performance and Tuning Guide*. Default: /WIN=7
* /LRU=n	n is the number of directories to be kept preaccessed while the volume is in use; that is, the number of entries in the LRU (least recently used) list for each volume. Preaccessing directories by entering them in the LRU normally saves three disk accesses for every directory operation. Ideally, n should be equal to the expected number of concurrent users of the volume; however, this will be at the expense of system dynamic memory. For further information see the *IAS Performance and Tuning Guide*. Default: /LRU=3
/DENS=density	(For magnetic tape) density is 800 for 800 BPI, or 1600 for 1600 BPI. For a floppy disk in an RX02 drive, density is HIGH or LOW. The drive density is compared against the specified density for consistency. Single (or RX01-compatible) density is LOW. Double density is HIGH.
* /INDX=option	option is the index file position. The options are: • BEG–Beginning of volume • MID–Middle of volume • END–End of volume • BLK:number–Logical block number For example: /INDX=BLK:100 Default: /INDX=MID

Keyword	Explanation
/BAD=[options]	The keyword /BAD indicates that bad-block processing is to be included in the volume initialization. Consequently, bad blocks on the volume are marked as in use and cannot be allocated to files. The brackets surrounding each option are required syntax. The options are: • [MAN]–Accept a bad block list specified from the terminal. • [AUTO]–Read the bad-block descriptor file on the last track of the volume created by the manufacturer's diagnostic routines or the bad block locator utility (BAD). • [NOAUTO]–Ignore the bad-block descriptor file and perform no bad-block processing. • [AUTO,MAN]–Read the bad-block file and, when done, accept blocks specified from the terminal. • [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 contain manufacturer-recorded bad block data in the last tracks (DL:, DM:, and DR:). • [OVR,MAN]–Override the manufacturers bad block descriptor file and accept blocks specified from the terminal.
/VI	Shows the current value of all keywords.

COMMAND VARIATIONS

Not applicable.

Table 7–2 details the maximum files per volume for each Files-11 device.

Table 7–2 Maximum Files Per Device

Disk Name	Volume Size	Maximum Files	Default Files	Maximum Indx. Hdrs.	Default Indx. Hdrs.	Default Allocation
DECtape	576	278	34	1	1	16
DECtape II	512	247	30	1	1	16
RF	1024	499	62	1	1	16
RK05	4800	2357	294	1	1	147
RK06	27126	13344	1668	1	1	834
RK07	53790	26466	3308	2	1	1654

Maximum Files—Maximum value that can be specified for /MXF=n.
Default Files—Default value for /MXF=n.
Default Allocation—Default value for /INF=n.

Table 7–2 (Cont.) Maximum Files Per Device

Disk Name	Volume Size	Maximum Files	Default Files	Maximum Indx. Hdrs.	Default Indx. Hdrs.	Default Allocation
RL01	10240	5034	629	1	1	314
RL02	20480	10074	1259	1	1	629
RM02	131680	64798	8099	3	1	4049
RM03	131680	64798	8099	3	1	4049
RM05[1]	500384	65500	30781	3	2	25593
RP02	40000	19680	2460	1	1	1230
RP03	80000	39365	4920	2	1	2460
RP04	171798	65500	10567	3	1	5283
RP05	171798	65500	10567	3	1	5283
RP06	340670	65500	20956	3	1	10478
RS03	1024	499	62	1	1	16
RS04	2048	1003	125	1	1	16
RX01	494	238	29	1	1	16
RX02	988	481	60	1	1	16

[1] This device is not backwards compatible by default.

Maximum Files—Maximum value that can be specified for /MXF=n.
Default Files—Default value for /MXF=n.
Default Allocation—Default value for /INF=n.

TECHNICAL NOTES

1 When you specify MAN as the /BAD keyword option, the program prompts for bad blocks as follows:

```
INI>LBN(S)=
```

You can then enter bad blocks in the following format:

```
blocknum:number
```

where:

- blocknum = Bad block number

- number = Number of sequential bad blocks beginning at blocknum

INI requires the colon when you specify a sequence of bad blocks.

Both blocknum and number default to decimal unless preceded by a pound sign (#).

To specify a single block, enter the block number and omit the colon.

You can also specify a sequence of bad blocks as well as a single bad block on the same command line. For example, the command line:

```
INI>LBN(S)=100:2,3,200:100,45:1
```

enters blocks 100, 101, 3, 200 through 299, and 45 as bad blocks. You can separate bad block series with a space, tab, a comma.

A null line (a carriage return in response to the prompt) displays the bad blocks. The first number in the display represents the beginning block of the sequence; the second number is the number of bad blocks in the sequence. (Note that the numbers are decimal.)

To terminate bad block input, type Ctrl/Z or ESC.

The default is BAD=[AUTO]. INI obtains the bad block information from a factory-recorded bad block descriptor file, located on the last track of an RK06, RL01 or RM03 disk. Note that INI also searches the last track for a bad block descriptor file created by the BAD utility.

2 Because of the large number of options available with the INI command, it might not be possible to enter the entire command string on a single line. For this reason, INI can accept multiline commands. To use this, you must follow the procedure outlined below. INI must be initiated as follows:

```
MCR>INI
```

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

```
INI>
```

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

3 Each entry in the /PRO keyword consists of from one to four letters which have the following meaning:

R—Read access
W—Write access
E—Extend access
D—Delete access

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

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

NOTE: On magnetic tape, the protection option delete is equated to write, since write access implies total access because of the sequential nature of magnetic tape. In addition, extend access may be specified without write access. This implies that the user may extend the last file on the volume, but may not write on the volume in any other manner.

EXAMPLE(S)

- Example 1:

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

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

Volume label	TESTPACK
UIC	[1,1]
Volume protection	System = RWED
	Owner = RWED
	Group = RWED
	World = RWED
Maximum number of files	294
Default file extension	5
File protection	System = RWED
	Owner = RWED
	Group = RWE
	World = R
Characteristic word	ATCH and DCF
Index file position	MID
Bad blocks	Automatically accounted for
Number of headers in index file	16
Window size	7
Number of preaccessed directories	3

- Example 2:

```
MCR>INI DK1:TESTPACK/PRO=[RWED,RWED,RW,R]/CHA=[ATCH,DCF]/BAD=[MAN]
INI>BAD=6.,10.
INI>BAD=23.
INI>BAD=
```

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

Volume label	TESTPACK
UIC	[1,1]
Volume protection	System = RWED
	Owner = RWED
	Group = RW
	World = R
Maximum number of files	294
Default file extension	5

File protection	System = RWED
	Owner = RWED
	Group = RWE
	World = R
Characteristic word	ATCH and DCF
Index file position	MID
Bad blocks	6 to 17, and 23
Number of headers in index file	16
Window size	7
Number of preaccessed directories	3

- Example 3:

```
MCR>INI
INI> DK2:FRED/UIC=[310,275]/LRU=2-
INI> /PRO=[RWED,RWED,R,]/INDX=BEG-
INI> /MXF=1000.
MCR>
```

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

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

INS—INSTALL TASK

FUNCTION

The INS (INSTALL TASK) command installs tasks and shareable global areas (SGAs) in the system. The INS command can also override or extend some of the attributes assigned to the task by the Task Builder. Overriding is specified in the form of keyword parameters appended to the task image file specification(s) in the INS command line. Shareable global areas are described in the *IAS Executive Facilities Reference Manual.*

FORMAT

MCR>INS[TALL] *filespec[/keyword(s)][,filespec[/keyword(s)],...]*

or:

MCR>INS[TALL] *@indirect*

parameter descriptions

filespec

File specification for the image (task or SGA) being installed. Table 7–3 contains a list of defaults in INS file specifications.

/keyword(s)

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

Option	Description
/PAR=partition name	Specifies the partition where the task or shareable global area is to be installed. When an SGA is being installed, the INS command recognizes the special pseudo-partition name $$$EXT and aligns that SGA on the external page in real memory. Also, the SGA is flagged as if it were loaded, although no actual load occurs. This function enables a task to bind reference locations in the external page without being executive privileged. For example: /PAR=GEN /PAR=$$$EXT (for SGAs only) Default: The partition specified when building the task or SGA, or if none, the partition specified as the default during system generation.

Option	Description
/PRI=priority-number (decimal)	Specifies the execution priority to be assigned to the task. Priority ranges from 1 to 250. For example: /PRI=200 Default: /PRI=50
/TASK=taskname	This option enables you to assign a name to the task or SGA being installed. This name overrides the one assigned by the task builder. Task names can be from 1 to 6 alphanumeric characters in length. For example: /TASK=RICK
/POOL=pool-limit (decimal)	Enables you to assign a new pool limit to the task being installed. The pool limit value can be from 0 to 255 decimal, and represents the maximum number of 8-word nodes that the task is enabled to use at one time. For example: /POOL=100 Default: /POOL=40 (established by the task builder)
/UIC=[uic]	Enables you to change the task's UIC, or the owning UIC of an SGA. For example: /UIC=[11,11]
/RUN[=REM] (run-remove)	This option is different from the above mentioned options, in that it does not change any task attributes. When /RUN is specified, it directs INS to install the specified task and run it. If the optional subparameter, =REM, is appended to /RUN, it specifies that the named task is to be removed after execution.
/ACC = non-owner-access	(Applicable only when installing SGAs). The non-owner access can be expressed in the usual form for protection; that is, [RWED,RWED,RWED,RWED]. The protection groups refer respectively to system (UIC group code less than 10), owner, group and world. The protection categories are: R–Read access W–Write access E–Extend access D–Delete access Alternatively, the following abbreviated protection codes can be specified: NA–No access to non-owner, equivalent to [D,RWED,,] RO–Read-only access to non-owner equivalent to [RD,RWED,R,R] RW–Read-write access to non-owner, equivalent to [RWD,RWED,RW,RW] For example: /ACC=RO /ACC=[RWED,RWED,R,] The system adds delete access to the system and owner protection groups if it is not specified by the user. This prevents an SGA being created that cannot be deleted when required. Default: /ACC=NA
/LI, /CM and /RG	(Applicable only when installing an SGA.) These options are used to specify that the entity being installed is either a library SGA (/LI), a common area SGA (/CM) or a region SGA (/RG). See the *IAS Task Builder Reference Manual* for a description of the different types of SGA. INS determines that the file being installed is an SGA (that is, that there is no task header). Switches /LI, /CM, and /RG are ignored if used when installing tasks.

Option	Description
	Default: /CM
/INC=task size increment	Overrides the EXTTSK Task Builder option. It specifies the decimal number of words by which the upper read/write area of the task is to be extended. The value specified will be rounded up to the next 32 word boundary. This option is illegal for an SGA. For example: /INC=2048 Allocates an additional 2K to the task's address space. Default: Either 0 or the EXTTSK value.
@indirect	Indirect command file specification for a file that contains a list of valid INS commands. Three levels of indirect command file structure are enabled.

COMMAND VARIATIONS

Not applicable.

TECHNICAL NOTES

1 The *IAS Task Builder Reference Manual* describes the building of an SGA and the characteristics of the different types of SGA.

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

Table 7–3 Defaults in INS File Specifications

Specification	Default
dev:	If dev is omitted in the first or only file specification, SY: is used. If dev is omitted in the second to nth file specification, the device specified or defaulted from the previous file specification is used.
[ufd]	If [ufd] is omitted in further file specification, the UFD that corresponds to the UIC under which the user logged in is used.
filename	Must be specified.
.type	.TSK
;ver	The latest version for the file.

EXAMPLE(S)

- Example 1:

```
MCR>INS SCAN
```

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

- Example 2:

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

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

LOA—LOAD

FUNCTION

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

FORMAT

MCR>LOA[D] *dd[n][:]*

parameter descriptions

dd
Device whose handler is to be loaded.

n
Unit number whose handler is to be loaded. This can only be specified for multiuser handlers (see the *IAS Guide to Writing a Device Handler Task*.

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

- Example 1:

  ```
  MCR>LOAD CR
  ```

  The card reader device handler is loaded into memory. I/O requests to the card reader can now be honoured.

- Example 2:

  ```
  MCR>LOA LP2
  ```

  The handler for unit 2 of the line printer is loaded into memory.

MEM—MEMORY UNLOCK

FUNCTION

The MEM (MEMORY UNLOCK) command enables you to unlock tasks that the Executive locked in memory as a result of a memory parity error. Parity error handling is described in the *IAS System Management Guide*.

FORMAT

MCR>MEM[ORY] *[taskname][/switches][,taskname[/switches]] [,...]*

parameter descriptions

taskname

Optional. If specified, it is the name of the task to be freed. MEM searches the Active Task List (ATL) and unlocks the first locked task encountered with the specified name. The task name must be included if a series of specifications are included in the command.

/switches

Can be either or both of the following options:

Option	Explanation
/TI=xxn	Indicates that the first locked task encountered in the ATL with the specified TI is to be unlocked. The value xxn is the device mnemonic; for example, TT3. If both task name and TI are specified in the MEM command, the first locked task encountered in the ATL with the specified task name and TI assignment is freed.
/AL	indicates that all locked tasks are to be freed. If both task name and /AL are specified, all locked tasks with the specified task name are freed. If both /TI and /AL are specified, all locked tasks with the specified TI are freed. If task name, /TI, and /AL are specified, all locked tasks with the specified task name and TI are freed.

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

- Example 1:

```
MCR>MEM /AL
```

- Example 2:

```
MCR>MEM ABC/TI=TT4
```

MOU—MOUNT VOLUME

FUNCTION

The MOU (MOUNT VOLUME) command has the same format as MOU for the non-privileged user. You do, however, have three switch options that are not available to the non-privileged user. These options are as follows:

Option	Description
/UIC=[uic]	Specifies the volume UIC to be used for the duration that the volume is mounted. For example: /UIC=[11,11]
/OVR	Enables you to override the volume label check, in which case the volume label need not be specified. This option must be specified when mounting magnetic tapes with blank volume labels.
/PRO=[system, owner, group, world]	Enables you to change the volume access privileges. Each entry consists of from one to four letters which have the following meaning: R–Read access W–Write access E–Extend access D–Delete access Omitting one of these letters in an entry signifies that the access right is denied to the user. Protection code subparameters (system, owner, group, world) are positional. The location of an entry in the parameter string defines the user to whom the codes apply. For example: /PRO=[RWED,RWED,RW,RW] In this example, group and world are denied extend and delete access.

COMMAND VARIATIONS

See the MOU command for nonprivileged users (Chapter 6).

EXAMPLE(S)

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

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

OPE—OPEN REGISTER

FUNCTION

The OPE (OPEN REGISTER) command enables you to open a memory address for examination and optional modification.

When a location is opened, the specified memory address and the contents of the address are listed on your terminal as follows:

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

where:

- memory-address (printed by the system) = 6-digit (octal) virtual address.
- contents/ (printed by the system) = 6-digit (octal) valule at the memory-address, followed by a slash (/).
- new-value (typed by the user) = New value entered into the memory-address. If no value is specified and a line terminator is input, location value remains the same.
- line-terminator (typed by the user) = One of the optional line terminators, all of which follow.
  1. ESC—ESCape or ALTMODE: end of command. This is the only means of exit from the OPEN function.
  2. RET—Carriage return: the next sequential location is opened.
  3. ^ RET—Up arrow carriage return: the previous location is opened.
  4. * RET—Asterisk carriage return: the location pointed to by the final contents of the opened location is opened.

FORMAT

MCR>OPE[N] *memory-address[+n][/option(s)]*

parameter descriptions

memory-address

Octal location to be opened.

n

Optional number to be added to the memory address, if positive, or subtracted from the memory address, if negative.

/option(s)

The following options specify the address space to which the memory address applies: if no option is specified, the address is interpreted as an absolute memory address. If an option is specified, the address is interpreted as a virtual address within the specified address space.

Option	Description
/TASK=task name[/TI=dev]	The specified task must be fixed in memory, where dev is the device name.
/PAR=partition name	Specifies a partition in the system.
/KNL	specifies kernel virtual address space.

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

- Example 1:

```
MCR>OPE 16014/KNL
16014 060014/350
16016 071005/ESC
```

In this example, the memory address is 16014. The option specified is Kernel Virtual. The system responds by printing the address (16014) and the contents of the address (060014), followed by a slash. You respond by entering the new value "350", followed by a carriage return. Terminating the line by a carriage return causes the next sequential location to be displayed. The dialogue is then terminated by entering ESC.

- Example 2:

```
MCR>OPE 4/TASK=ABC
000004 111111/
```

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

- Example 3:

```
MCR>OPE 4
000004 000305/20054ESC
```

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

OPR—OPERATE

FUNCTION

The OPR (OPERATE) command, for the privileged user, has the same format as OPR for the nonprivileged user. You do, however, have the following three switch options that are not available to the nonprivileged user.

Option	Description
/CHG	Set the forms type of the specified device to 7. When the current file is printed, you can mount new forms on the device and queue a test pattern file to test for forms alignment. See the QUE command option /TE.
/FO:n	Set the forms of the specified device to n (where n is a decimal number number ranging from 0 to 6). Processing of files requiring the specified forms type and device type begins automatically.
/RE	Make all queue requests for the specified device inactive. OPR then despatches queued files whose forms type match the forms type currently specified for this device. Use this switch only when you know that print files are queued, and that the system task SPR2.. is not active. This is normally the case when the system is restarted after a halt that occurred while a queued file was being printed.

PWD—ENTER PASSWORD

FUNCTION

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

COMMAND VARIATIONS

Not valid in MCR mode.

QUE—QUEUE

FUNCTION

The QUE (QUEUE) command for the privileged user has the same format as the QUE command for the nonprivileged user. You do, however, have the option of deleting (/KI) or modifying (/MO) any entry in the queue.

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

REA—REASSIGN

FUNCTION

The REA (REASSIGN) command enables you to deassign a task's LUNs from one physical device unit and assign them to another.

FORMAT

REA[SSIGN] *taskname lunlist devunit:*

parameter descriptions

taskname
Name of the task whose LUNs are being reassigned.

lunlist
List of one or more LUNs separated by commas.

devunit
New device unit symbol followed by the unit number or NONE to deassign the LUNs.

COMMAND VARIATIONS

Not applicable.

TECHNICAL NOTES

1 The REA command performs reassignments in the indicated task's header image on disk.

2 The REA command does not affect any active versions of the specified task.

3 Removing and reinstalling a task will restore its LUNs to their original assignments specified at task build time.

4 If the same task image is also installed with a different name, the LUN assignments for all such tasks will be changed.

EXAMPLE(S)

- Example 1:

```
MCR>REA JOE 3 TT0:
```

  Reassign, in task JOE, LUN 3 to device TT0.

- Example 2:

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

  Reassign, in task BILL, LUNs 2, 3 and 4 to device TT1.

- Example 3:

```
MCR>REA SAM 5 NONE
```

  Deassign LUN 5 of task SAM. Any requests directed to LUN 5 will fail with the message LUN NOT ASSIGNED being issued.

RED—REDIRECT

FUNCTION

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

FORMAT

MCR>RED[IRECT] *new-dev[:]=old-dev[:]*

parameter descriptions

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

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

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

COMMAND VARIATIONS

Not applicable.

TECHNICAL NOTES

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

2 If, through a sequence of RED commands, you establish a redirect chain which returns to old-dev, the following message is issued, (and the RED command is rejected):

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

3 A device which has been redirected may be restored to normal by specifying it as both new-d and old-dev.

4 A device cannot be redirected if it is mounted or attached, or if it is already redirected to a device which is mounted or attached.

5 An output spooled device can only be redirected to another output spooled device.

6 When timesharing is active, only spooled devices can be redirected.

EXAMPLE(S)

- Example 1:

```
MCR>RED TT3:=TT6:
```

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

- Example 2:

```
MCR>RED TT:=LP:
```

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

If LP is a spooled device, you must dismount SP0: before you can redirect LP.

- Example 3:

```
MCR>RED LP:=LP:
```

Undo the redirection specified in Example 2.

- Example 4:

```
MCR>RED TT2:=TT1:
MCR>RED TT3:=TT2:
MCR>RED TT1:=TT3:
RED -- CIRCULAR REDIRECT CHAIN
```

The last RED command (TT1:=TT3:) is not accepted.

REM—REMOVE

FUNCTION

The REM (REMOVE) command enables you to take an inactive, unfixed task out of the system.

The REM command also enables you to remove an SGA when the tasks that link to it are not installed.

FORMAT

MCR>REM[OVE] *name1 [/switch][,name2[/switch],...]*

or:

MCR>REM[OVE] *@indirect*

parameter descriptions

name

Name of a task or SGA being removed.

/switch

The following switches are available:

Switch	Function
/LI	Identifies the named SGA as being a library.
/CM	Identifies the named SGA as being a common area.
/RG	Identifies the named SGA as being an installed region.
/NH	Enables a task to be removed even though its task header has been corrupted or deleted. The task header will not be accessed, and therefore the SGAs to which the task is linked will not have their installed task counts decremented. This switch should only be used if REM gives the error message ERROR READING TASK HEADER.

@indirect

Indirect command file specification for a file that contains a list of REM commands. Three levels ol indirect command structure are enabled.

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

- Example 1:

  ```
  MCR>REM SCAN
  ```

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

- Example 2:

  ```
  MCR>REM SCAN,XKE,JAG
  ```

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

- Example 3:

  ```
  MCR>REM SOMLIB/LI
  ```

  In this example, the library SGA SOMLIB is removed, provided that none of the tasks that link to it is installed.

- Example 4:

  ```
  MCR>REM MYREG/RG
  ```

  In this example, the installed region SGA MYREG is removed.

- Example 5:

  ```
  MCR>REM OLDTSK
  REM -- ERROR READING TASK HEADER
  MCR>REM OLDTSK/NH
  ```

  In this example, the task header for task OLDTSK cannot be accessed. Therefore, the task is removed with the 'noheader' option.

RES—RESUME

FUNCTION

The RES (RESUME) command enables you to resume the execution of any task in the system. (See also the RES command for the nonprivileged user in Chapter 6.)

FORMAT

MCR>RES[UME] *taskname[/TI=dev][,taskname-1[/TI=dev],...]*

parameter descriptions

taskname

Name of the task being resumed.

/TI=dev

Optional switch (appended to the task name) that enables you to resume any task in the system b} specifying its TI (dev is the device mnemonic for the terminal where the task is run).

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

```
MCR>RES TK32B/TI=TT12
```

Resume task TK32B, initiated on terminal TT12.

RUN

FUNCTION

The RUN command enables you to initiate tasks in any of six different forms as follows:

1 Request the execution of a task when memory becomes available (REQUEST—Format 1).

2 Execute a task, contingent upon immediate memory availability (EXECUTE—Format 2).

3 Run a task at a specified time and, optionally, rerun it periodically (RUN—Format 3).

4 Schedule the task for running at a specified time and, optionally, rerun it periodically (SCHEDULE—Format 4).

5 Synchronize the running of a task with a time unit and, optionally, rerun it periodically (SYNCHRONIZE—Format 5).

6 Run a task that is not installed in the system. This form of the RUN command is described in Chapter 6.

The default UIC for the general user is the UIC where the user logged in. For the privileged user, the default UIC of the task being run is used (see Table 2–1).

COMMAND VARIATIONS

In MCR mode, the commands are identical except that RRT should be used instead of RUN.

FORMAT

Format 1:

RUN REQUEST

The REQUEST form of the RUN command enables you to request the execution of tasks. If memory is not available, the corresponding tasks are queued until memory becomes available.

MCR>RUN taskname *[/options][,taskname[/options],...]* ESC

parameter descriptions

taskname

Name of the installed task being run.

/options

Any of the following:

/PRI = Priority number (decimal).
/PAR = Partition name.
/TI = Terminal for which the task is to be run.
/UIC = [user identification code].

TECHNICAL NOTES

1 Execution is contingent on priority and memory availability.

2 See Table 2–1 for a description of the UIC where the task is to run.

EXAMPLE(S)

- Example 1:

```
MCR>RUN SCAN ESC
```

- Example 2:

```
MCR>RUN SCAN/PAR=XYZ/PRI=58 ESC
```

- Example 3:

```
MCR>RUN XKE,NK111/PRI=10,SCAN ESC
```

- Example 4:

```
MCR>RUN XKE/UIC=[11,1] ESC
```

- Example 5:

```
MCR>RUN JEN/PRI=50/TI=TT21 ESC
```

FORMAT

Format 2:

RUN EXECUTE

The EXECUTE form of the RUN command enables you to request task execution if memory is immediately availble. If no memory is available in the specified or default partition, the request is denied.

MCR>RUN taskname/MEM *[/options][,taskname/MEM[/options],...]* ESC

parameter descriptions

taskname

Name of the installed task being run.

/options

Any of the following:

/PRI = Priority number (decimal).
/PAR = Partition name.
/TI = Terminal where the task is to be run.
/UIC = [user identification code].

TECHNICAL NOTES

1 Execution is contingent on priority.

2 The EXECUTE form of the RUN command does not cause checkpointing to occur.

EXAMPLE(S)

- Example 1:

```
MCR>RUN SCAN/MEM ESC
```

- Example 2:

```
MCR>RUN SCAN/MEM/PRI=100/PAR=XXX/UIC=[1,3] ESC
```

- Example 3:

```
MCR>RUN SCAN/MEM/PRI=25,240Z/PAR=Z/PRI=99 ESC
```

- Example 4:

```
MCR>RUN SCAN/MEM/TI=TT12 ESC
```

FORMAT

Format 3:

RUN FORM OF RUN COMMAND

The RUN form of the RUN command enables you to schedule tasks in terms of *delta time from now* with the option to specify periodic rescheduling, run priority, and memory partition.

MCR>RUN taskname time *[/options] [,taskname time[/options],...]* ESC

parameter descriptions

taskname

Name of the installed task being run.

time

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

/options

Any of the following:

/PRI = Priority number (decimal).
/PAR = Partition name.
/RSI = Reschedule interval, expressed as nH, nM, NS, or nT.
/TI = Terminal where the task is to be run.
/UIC = [user identification code].

EXAMPLE(S)

- Example 1:

```
MCR>RUN XKE 15M ESC
```

Run task XKE 15 minutes from now.

- Example 2:

```
MCR>RUN XKE 15M/RSI=90S/UIC=[3,1] ESC
```

Run task XKE in 15 minutes, rescheduling it every 90 seconds thereafter. The UIC is [3,1].

- Example 3:

```
MCR>RUN XKE 15M/RSI=90S/PRI=150/PAR=XYZ ESC
```

Run task XKE in 15 minutes, rescheduling it every 90 seconds assigning it a priority of 150, and running it in partition XYZ.

- Example 4:

```
        MCR>RUN XKE 15M,NK111 20M,240Z 25M ESC
```

Run task XKE in 15 minutes, run task NK111 in 20 minutes, and run task 240Z in 25 minutes.

- Example 5:

```
        MCR>RUN NK11 30M/TI=TT5 ESC
```

Run task NK11 in 30 minutes, for terminal TT5.

FORMAT

Format 4:

RUN SCHEDULE

The SCHEDULE form of the RUN command enables you to schedule tasks in terms of absolute time of day, with the option to specify periodic rescheduling, run priority, and memory partition.

MCR>RUN taskname time *[/options]* ESC

parameter descriptions

taskname

Name of the installed task being run.

time

The absolute time of day when the task is to begin execution. Time is expressed in hours: minutes: seconds [:ticks].

/options

Any of the following:

/PRI = Priority number (decimal).
/PAR = Partition name.
/RSI = Reschedule interval, expressed as nH, nM, NS, or nT.
/TI = Terminal where the task is to be run.
/UIC = [user identification code].

EXAMPLE(S)

- Example 1:

```
MCR>RUN XK1 12:23:15 ESC
```

Schedule task XK1 at 12:23:15.

- Example 2:

```
MCR>RUN XKE 12:00:00/RSI=6M/PRI=99/PAR=XYZ ESC
```

Schedule task XKE at 12:00, reschedule it every 6 minutes, assign it a priority of 99, and execute it in partition XYZ.

- Example 3:

```
MCR>RUN SCAN 23:30:00/RSI=10M/PRI=200 ESC
```

Schedule task SCAN at 23:30, reschedule it every 10 minutes, assign it a priority of 200.

- Example 4:

```
MCR>RUN XKE 1:00:00,240Z 1:30:00/UIC=[111,1] ESC
```

Schedule task XKE at 1:00, task 240Z at 1:30 with a UIC of [111,1].

- Example 5:

```
MCR>RUN SCAN 19:30:00/TI=TT12 ESC
```

Schedule task SCAN at 19:30 for terminal TT12.

FORMAT

Format 5:

RUN SYNCHRONIZE

The SYNCHRONIZE form of the RUN command enables you to synchronize the scheduling of a task in terms of delta time from *clock unit synchronization*, with the option to specify periodic rescheduling, run priority, and memory partition.

MCR>RUN *taskname starttime+time[/options]*ESC

parameter descriptions

taskname

Name of the installed task being run.

starttime

Synchronization clock unit as follows:

H = synchronize on the next hour.
M = synchronize on the next minute.
S = synchronize on the next second.
T = synchronize on the next tick.

time

Delta time increment added to the synchronization clock unit that yields the actual start time. Time is expressed as nH, nM, nS, or nT, where n is the decimal number of hours, minutes, seconds, or ticks.

/options

Any of the following:

/PRI = Priority number (decimal).
/PAR = Partition name.
/RSI = Reschedule interval, expressed as nH, nM, NS, or nT.
/TI = Terminal where the task is to be run.
/UIC = [user identification code].

EXAMPLE(S)

- Example 1:

```
MCR>RUN NK111 H+10M/TI=TT10 ESC
```

Synchronize task NK111, schedule for 10 minutes after the next hour for terminal TT10.

- Example 2:

```
MCR>RUN NK111 S+10M/PRI=10,240Z M+0S/RSI=15S/PRI=90 ESC
```

Synchronize task NK111, schedule for 10 minutes after the next second, assign it a priority of 10; synchronize task 240Z, schedule it for 0 seconds after the next minute, rescheduling it every 15 seconds, and assign it a priority of 90.

SAV—SAVE

FUNCTION

The SAV (SAVE) command records the core image of an IAS system on the disk from which it was originally bootstrapped. This enables a bootstrap to reload it and start up the system in the state in which it was saved.

WARNING: Use the SAV command only when the system is quiescent.

FORMAT

MCR>SAV[E] *[/switch]*

parameter descriptions

/switch

Is one of the following optional switches:

Switch	Function
/MOU=dev:[dev:...:dev:]	This switch is used to enable the system to be saved with the specified devices mounted. See Technical Note 4.
/LOG=TTn:[TTn:...:TTn:]	This switch is used to enable the system to be saved with the specified terminals logged in.
	/NO, /NOX, /NOT, /NOXT
	All four switches have the same effect; that is, they inhibit automatic memory expansion or truncation when the saved image is rebooted. If none of these switches is used, SAV determines exactly how much memory exists and expands or truncates the highest partition in the system.
/NOIN	This switch inhibits the automatic reinstallation of tasks and SGAs when the system is booted. The switch can be used to speed up booting the system as long as no installed task files are to be moved. In particular, if the disk is compressed using the Disk Save and Compress utility (DSC), the disk is longer be bootable if /NOIN was specified.
/NOT	This switch prohibits memory truncation processing during boot.
/NOX	Prohibits memory expansion processing during boot.
/NOXT	Prohibits both memory truncation and expansion processing during boot.
/LGOUT	Logs out system console after boot.
/CMD	Executes SY:[1,1]STARTUP.CMD after boot.
/NOTIM	Prohibits IAS time and date prompt after boot.

COMMAND VARIATIONS

Not applicable.

TECHNICAL NOTES

1 Because the SAV command provides a copy of a completed system configuration, it must be used only when the system is quiescent. This is established by searching the system database for any of the following conditions:

- Mounted devices (other than the system device).
- Users logged into terminals other than the one where the SAV command was issued.
- Tasks with I/O in progress.
- Tasks with send/receive data queued to them.
- Tasks being loaded or checkpointed.
- Shareable global areas (SGAs) being loaded or, in the case of common area SGAs, being recorded on disk.
- Tasks or SGAs installed from a device other than the system disk.
- Tasks loaded or fixed beyond the end of the SAVE file.
- SGAs including the read-only root of multiuser tasks loaded beyond the end of the save file.

If any of these conditions is detected, SAV issues an appropriate error message.

2 If the system disk is mounted when the SAV command is issued, it is automatically dismounted. When the system is booted, the system disk is remounted in a way equivalent to the command MOU SY:/OVR.

3 The SAV command will attempt to record (on the system device) all memory specified at system generation time. If more memory exists than was declared at system generation only the declared memory will be saved. If, however, less memory exists than was declared at System Generation, only the amount that exists will be saved. For a PDP-11/44 or PDP-11/70, no more than 124K will be saved.

4 SAV does not permit the system to be saved with volumes mounted (except the system disk) or terminals, other than the issuing terminal, logged in, unless the appropriate /MOU or /LOG switches are specified.

In the case where the /MOU switch is specified, for Files-11 volumes you must take the following information into account:

- When a volume is mounted, volume control data is established in memory to reflect the volume's current file status. This data is updated with every file operation.
- When a volume is dismounted, the volume's control data is reset.

- If you choose to save the system with volumes mounted, the volume control data for each mounted volume is saved reflecting the current status of the volume. It is imperative that no file activity occurs on the volume between the time the system was saved, and the next bootstrapping of the system. Otherwise, the integrity of the volume will be destroyed.
- When the system is rebootstrapped, the status of the volume must be exactly the same as it was when the system was saved, or the integrity of the volume will be destroyed on the first file operation.
- After the system is rebootstrapped, and before any file activity has occurred, you can execute the following commands to ensure the integrity of any volume:

```
MCR>DMO dev:

MCR>MOU dev:
```

  This resets the volume's control data to reflect the current volume status.
- In view of the above, it is strongly recommended that /MOU should not be used for Files-11 volumes.

  If the /LOG switch is specified when the system is rebootstrapped, the specified terminals are automatically logged in with the same UIC and privilege they had when the system was saved.

5 When the system is saved (unless the /NOIN switch was specified) the disk information for all installed tasks, SGAs and the swap file is converted from the starting logical block number to the file-ID of the corresponding file. When the system is bootstrapped this information is converted back.

Since the file-ID identifies a file independently of its physical position on the disk, the files may be re-arranged in any way and the system will still bootstrap correctly. In particular, a disk can be compressed using the Disk Save and Compress Utility (DSC) and will still function correctly. This process may take several minutes, depending upon the type of disk anc the number of installed tasks. For further information, see the *IAS Installation and System Generation Guide*.

6 When a saved system is rebootstrapped or restarted, the system is expanded or truncated automatically to the amount of physical memory available (unless the /NOXT switch was used when the system was saved).

Because SAV is active when a save is performed, it is active when the re-bootstrap/restart occurs. It is, in fact, the task SAV that restarts a system and performs the memory size calculations.

To expand the system, SAV determines how much physical memory there is by attempting to access memory in 1K word increments, starting at the first word after the save image. If /NOXT is not specified and the address at the top of memory is greater than that at the last save, or at system generation if no save was performed, SAV performs the following steps:

- Adjusts the size of the last (highest addressed) partition.
- Prints an explanatory message on the terminal from which the SAV was initiated.

If the actual memory is less than the amount contained in the SAVE file, SAV attempts to truncate one or more partitions starting from the highest end of memory. Truncation fails and an appropriate message is printed under the following circumstances:

- Any occupied partition size reduces to zero.
- Any truncation occurs in an occupied user-controlled partition.

- Truncation in a system-controlled partition is so extensive that a previously occupied area of memory does not exist.

In summary, unoccupied partitions can be truncated to size zero. Unoccupied parts of system-controlled partitions also can be truncated. Successful truncation results in a descriptive message printed at the terminal.

7 Irrespective of any previous memory expansion or truncation, the SAVE file is always the length specified at System Generation and only that amount of memory is saved. This is the reason for the checking described in Technical note 1 on entities loaded beyond the end of the SAVE file.

8 The *IAS Installation and System Generation Guide* contains more details about SAV.

EXAMPLE(S)

- Example 1:

```
MCR>SAV
```

In this example, the current status of the system is saved on the disk from which it was originally bootstrapped. System changes made by the RED or other MCR commands are also saved with the system core image.

- Example 2:

```
MCR>SAV /MOU=DF0:
```

In this example, the system is saved with DF0 mounted.

SET

FUNCTION

The SET command enables you to:

1 Change the terminal default UIC.

2 Change the MCR timeout.

3 Change the MCR privileges.

4 Set or clear output spooling on a device.

5 Allocate a CLI to a terminal.

6 Control cache memory on a PDP-11/44 and PDP-11/70.

7 Set or clear the software switch register on machines supplied without hardware sense switches.

8 Set or display maximum extended task size.

FORMAT

MCR>SET *[/keyword(s)]*

parameter descriptions

/keyword

One or more optional keywords that, when specified, alter or set the default values they represent. The following keyword options are provided:

Option	Description
/UIC=[group,member]	This option enables you to change the terminal default UIC to the one specified to the right of the equal sign (=). Example: /UIC=[200,200]
/TMO=nnnnn	This option enables you to change the MCR timeout value. The number nnnnn is a decimal value (any number from 1 to 32767) representing the number of seconds MCR is to wait for a command before timing out. Example: /TMO=90
/PRV=dev:[dev:...:dev:]	This option defines the specified device(s) as privileged terminal device(s). Example: /PRV=TT1:TT2:
/-PRV=dev:[dev:...:dev]	This option defines the specified device(s) as nonprivileged terminal device(s).

Option	Description
	Example: /-PRV=TT1:TT2:
/SLV=dev:[dev:...:dev:]	This option defines the specified device(s) as slave terminal device(s). Example: /SLV=TT1:TT2:
/-SLV=dev:[dev:...:dev:]	This option defines the specified device(s) as nonslave (command) terminal device(s). Example: /-SLV=TT1:TT2:
/SP=dev:[dev:...:dev:]	This option defines the specified device(s) as output spool device(s), see the *IAS System Management Guide*. Example: /SP=LP:TT1:
/-SP=dev:[dev:...:dev:]	This option defines the specified device(s) as non-output spooled. The physical device corresponding to SP0: must be dismounted for this command to succeed. Example: /-SP=TT1:TT2:
/CLI=xxx:ddn[:...]	This option defines the named command language interpreter (CLI) to be allocated to a terminal. This is normally DCL or MCR. xxx Is the name of the CLI
ddn[:...]	List of terminals. These terminals must be logged out interactive terminals.
/CAC=n	This option specifies the number of parity errors that can occur in a PDP-11/44 or PDP-11/70 cache group in one minute before the Executive turns the group off. The value n can be from 1 to 32,767 (decimal). The default value for n is 50. Parity error handling is discussed in the *IAS System Management Guide*.
/CAC=ONn	This option enables you to turn on a PDP-11/44 or PDP-11/70 cache memory group that had previously been turned off. The value n indicates the number of the group.
/CAC=OFn	This option enables you to turn off a PDP-11/70 cache memory group. The value n indicates the number of the group.
/SWR=option	The following four options are enabled only on a processor supplied without hardware sense switches.
/SWR=nnnnnn	This option sets an absolute octal value nnnnnn in the Software Switch Register (SWR).
/SWR=SET:m[:n]	This option sets the specified bits to 1.
/SWR=CLR:m[:n]	This option clears the specified bits.
/SWR	This option displays the contents of the SWR in octal.
/MAXEXT=nnnn	This option specifies the maximum size to which a task can extend itself using the EXTK$ directive (see the *IAS System Directives Reference Manual*). The size is given as an octal number of 32 word blocks.
/MAXEXT=nnK	This option specifies the maximum size to which a task can extend itself, given as a decimal number of K words.
/MAXEXT	This option displays the current extension limit, as an octal numbr of 32 word blocks.
/NOMAXEXT	This option enables tasks to extend to the maximum imposed by mapping constraints. It overrides any limit set by the use of the Task Builder option MAXEXT for a particular task. (See the *IAS Task Builder Reference Manual* for more details.)

COMMAND VARIATIONS

1 Do not use SET /UIC in MCR mode. The effect is not predictable. Use the DCL command SET DEFAULT (timesharing), or SET UIC (multiuser) instead (see the *IAS PDS User's Guide*).

2 SET /TMO has no effect in MCR mode.

3 SET /SP and SET /-SP should not be issued while timesharing is active.

4 SET /CLI does not affect CLI allocation in a timesharing system The ALLOCATE/TERMINAL DCL command must be used for this (see the *IAS System Management Guide*).

EXAMPLE(S)

- Example 1:

  ```
  MCR>SET /UIC=[20,20]/TMO=90
  ```

  In this example, the following actions are performed:

  - The terminal UIC is set to [20,20].
  - The MCR timeout is set to 90 seconds.

- Example 2:

  ```
  MCR>SET /SWR=SET:13:15
  ```

  On a processor supplied without hardware sense switches the software switch register bits 13 and 15 are set to 1.

- Example 3:

  ```
  MCR>SET /MAXEXT=20K
  ```

  This example sets the maximum size, to which any task can extend itself, to 20K words.

SWA—SWAP

FUNCTION

The SWA (SWAP) command enables swap files to be created, listed and deleted.

FORMAT

To create a swap file:

MCR> SWA[P] *ddnn:/LE:n[/switches]*

To list swap files:

MCR> SWA[P] */LI[:n]*

To delete swap files:

MCR> SWA[P] */DE:n*

parameter descriptions

ddnn:

Specification of the device where the swap file is to be created. Consists of a two-letter device mnemonic followed by the unit number (in octal). For example: DK0:.

/LE:n

Length of the swap file in k-words. One swap block rfepresents four disk blocks. The total swap space available is four million words.

/switches

Optional, additional switches, which include one or more of the following:

Switch	Description
/DV	The device is dedicated as a swap volume and must not be mounted. The device to be dedicated as a swap volume must not be a removable media device.
/[-]BA	The bad block switch when issued with the /DV switch indicates that bad block information is [not] to be used to determine which blocks are unusable.

Switch	Description
/RT	The swap file is to be used only for checkpointing real-time tasks. The number of the swap file must be less than that for any other swap file which does not have the /RT attribute.
/NU:n	Allocate the number n to the swap file. Normally, when a new swap file is created it is put after all existing files. This switch is used to override this. Existing swap files with a number greater than n are incremented by 1. The maximum number of swap files enabled is 64. **Note:** None of these files can have space allocated from them, or the command will fail.

/LI[:n]

Lists all swap files or only swap file n. The information listed is:

Information	Additional Information
Device	
File identity	
Size	
Space allocated	
Largest hole	
Flags	RT–File is reserved for real-time tasks. DV–File is on a dedicated swap volume. DEL–File has been marked for delete. BAD–File (on a dedicated volume) contains one or more bad blocks that cannot be allocated as swap space.

/DE:n

Deletes the swap file specified by n. If this file or any higher numbered files have space allocated from them, the file is marked for delete and the actual deletion takes place later.

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

- Example 1:

  ```
  SWA SY:/LE:200
  ```

  Create a 200K swap file on the system disk.

- Example 2:

  ```
  SWA DS0:/LE:511/DV/RT/NU:1
  ```

  Create a 511K swap file on DS0:, which is to be used only for swapping. Call the file number 1 and dedicate it to checkpointing real-time tasks.

- Example 3:

  ```
  SWA /LI
  ```

  List information about all swap files.

- Example 4:

  ```
  SWA /DE:2
  ```

  Delete swap file number 2.

TER—TERMINAL

FUNCTION

The TER (TERMINAL) command enables a user logged in under [1,1] to change the characteristics and the default characteristics of specified terminals.

For the options and switches other than /DEFAULT, see the TER command in Chapter 6.

FORMAT

MCR>TER

MCR>TER *TTa:[,TTb:]...[,TTn:] /options*

MCR>TER *TTa:[,TTb:]...[,TTn:] /options/DEFAULT*

MCR>TER *TTa:[,TTb:]...[,TTn:] /terminaltype/DS[/DEFAULT]*

MCR>TER *@indirect*

parameter descriptions

a:[b]...[n]
Unit numbers of the terminals affected.

The /DEFAULT option changes both the current setting(s) and the current default(s) for the terminal(s) so that the change is not lost at the next BYE command or dialup disconnection. In addition, any characteristics not specified in the command are reset immediately to the defaults currently in force for the specified terminals.

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

- Example 1:

```
MCR>TER TT1:TT3:/VT05/DS
```

Set TT1: and TT3: to have the characteristics of a VT05, the speeds to be the default for a VT05.

- Example 2:

```
MCR>TER TT4:/READAHEAD:IP/DEFAULT
```

Set both the readahead type and the default readahead type for TT4 to immediate processing. Set all other characteristics for TT4 to the defaults current for TT4.

TIM—TIME

FUNCTION

The TIM (TIME) command enables you to list the time and date or to alter the time and date values in the system clock calendar.

FORMAT

MCR>TIM[E] *[time] [date]*

parameter descriptions

time

Specified in the format hour:minute[:second]

date

Specified in the format dd-mmm-yy (dd = day, mmm= first three letters of the month, yy = last two figures of the year)

If the TIM command is executed without parameters, the time and date that the system is currently running under is listed.

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

- Example 1:

  ```
  MCR>TIM 9:10:20 21-MAR-78
  ```

  In this example the time and date in the system clock calendar are changed to 9:10:20 and 21-MAR-78, respectively.

- Example 2:

  ```
  MCR>TIM 15-AUG-78 12:11
  ```

  In this example the date and time parameters have been reordered.

UFD—USER FILE DIRECTORY

FUNCTION

The UFD (USER FILE DIRECTORY) command creates a UFD on the specified volume and enters its file name into the Master File Directory (MFD). The UFD command accepts the [group,owner] number as both the UFD name and the file's owner UIC.

FORMAT

```
MCR>UFD device:[uic] [/switch]
```

parameter descriptions

device:
Specifies the device that contains the volume to be acted upon.

[uic]
UIC of the UFD being created. The brackets that enclose the UIC are a required part of the parameter.

/switch
One or more of the following optional UFD switches described in Table 7–4.

COMMAND VARIATIONS

Not applicable.

TECHNICAL NOTES

Table 7–4 UFD Switches

Switch	Description
/PRO	This switch enables the owner of the directory to enable selective access to his file. This switch is specified in the following format: */PRO=[system,owner,group,world]* If the default /PRO switch is not specified, the default file protection for the volume is used.
/ALLOC	This switch enables the owner of the directory to preallocate space for a specified number of directory entries. This switch is specified in the following format: */ALLOC=number-of-entries* The number specified is assumed to be octal; the number may be specified with a trailing decimal point to represent a decimal value. The default is 32 octal.

EXAMPLE(S)

```
MCR>UFD DK:[1,1]/ALLOC=100
```

In this example, the request creates the UFD on disk device DK1 with the UIC [1,1] as the directory name; space is allocated for 100 (octal) directory entries.

UNF—UNFIX

FUNCTION

The UNF (UNFIX) command enables you to free a fixed task from memory.

FORMAT

MCR>UNF[IX] *taskname[/TI=dev] [,taskname[/TI=dev],...]*

parameter descriptions

taskname
Name of the task being unfixed.

/TI=dev
Optional switch (appended to the task name) that enables you to unfix any task in the system by specifying its TI (dev is the device mnemonic for the terminal corresponding to the TI under which the task is run).

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

- Example 1:

```
MCR>UNFIX XKE
```

  Unfix task XKE, freeing it from memory.

- Example 2:

```
MCR>UNFIX 240Z,NK111
```

  Unfix task 240Z and task NK111.

- Example 3:

```
MCR>UNFIX 240Z/TI=TT3
```

  Unfix task 240Z, which was installed from terminal TT3.

UNL—UNLOAD

FUNCTION

The UNL (UNLOAD) command causes the indicated device handler to exit, releasing its memory and making the device inaccessible.

FORMAT

MCR>UNL[OAD] *dd[n][:]*

parameter descriptions

dd

Name of the device whose handler is to be unloaded.

n

Unit number of the device handler to be unloaded. It must be specified only for multiuser handler tasks. If n is not specified, the handler is unloaded for all units.

COMMAND VARIATIONS

Not applicable.

EXAMPLE(S)

```
        MCR>UNLOAD LP
```

In the example above, the request unloads the line printer handler task.

UTL—USER TASK LIST

FUNCTION

The UTL (USER TASK LIST) command enables you to set parameters to control the timesharing scheduler. For a description of the scheduler, see the *IAS System Management Guide*. For advice on tuning the scheduler, see the *IAS Performance and Tuning Guide*.

FORMAT

MCR>UTL */switch(es)*

parameter descriptions

/switches

Include any of the following:

Switch	Function
/AM:n	Set memory size for CPU time allocation to n 32-word blocks.
/AT:n	Set CPU time allocation factor (ticks).
/BQ:n	Set quantum for batch level (ticks), (see Technical Note 3).
/BS:n	Set time between batch schedules (ticks).
/DI	Disable scheduler (see Technical Note 1).
/EN[:par]	Enable scheduler in partition par (see Technical Note 1).
/IT:n	Set idle time to n ticks.
/LV:n	Set number of levels for the scheduler to use. This switch can be issued only when the scheduler is disabled.
/MT:n	Set maximum task size (including task header) to n 32-word blocks.
/PT:n	Set size of promotion table. This switch can be issued only while the scheduler is disabled.
/QC:n	Set quantum constant for CPU time allocation (ticks).
/SH	Display scheduler parameters (see Technical Note 2).
/SP:n	Set number of ticks between scheduler promotion.
/TF:m:n	Set time factor for scheduling level m to n ticks.
/TP:n	Set timesharing ATL priority. This switch can be issued only while the scheduler is disabled.
/TS:n	Set maximum CPU time slice (ticks).

COMMAND VARIATIONS

Not applicable.

TECHNICAL NOTES

1 While the scheduler is enabled, any task which is run at a priority of 100 or less or in the specified partition will run under its control. If no partition name is specified, GEN is assumed.

The scheduler can only be disabled while no tasks are running under its control.

2 Used alone, the /SH switch displays all scheduler parameters. Used in conjunction with any other switch, the corresponding parameter is displayed.

3 Do not set the batch quantum to zero (/BQ:0) with the UTL command. If you want to set the batch quantum to zero you must use the SET BATCH command at system start-up. See the *IAS Installation and System Generation Guide* for the SET BATCH command.

EXAMPLE(S)

- Example 1:

```
MCR>UTL /QC:20
```

Set additive constant for quantum calculation to 20 (decimal) ticks.

- Example 2:

```
MCR>UTL /EN:TSPAR
```

Enable the scheduler to run tasks in partition TSPAR.

- Example 3:

```
MCR>UTL /SH
```

Display all scheduler parameters.

- Example 4:

```
MCR>UTL /SH/QC
```

Display the current setting of the quantum constant.

8 Magnetic Tape Volumes

The magnetic tape file system is a privileged task named MTAACP. It provides file primitive services to FCS. The file system also uses an associated message task named F11MSG that must be installed before a magnetic tape volume can be mounted.

The magnetic tape file system processes tape volumes that have been initialized by the MCR INI command (see Chapter 7). Mounting and dismounting a volume set are accomplished using the MCR MOU and DMO commands. Once a volume set is mounted, all volume or reel switching is transparent to the FCS level of the system. It is not necessary to mount all reels in a volume set on tape units simultaneously.

8.1 Initializing Tape Volumes

Initialization of a magnetic tape volume is an important step in preparing a magnetic tape file system for use. All protection is at the volume level. Therefore, to ensure proper use of volume sets, each volume must have a unique volume identifier (label). The system manager establishes a method of assigning unique volume labels and external identification.

8.2 Mounting a Volume Set

Use the MCR MOU command to mount the volume set. Keep the following information in mind when you mount the volume set:

1 You can assign any number of tape units to the volume set. All units must be compatible with each other, that is, they must be either 9-or 7-track and 1600 or 800 bpi.

2 You can specify all volume labels or only the first volume label in the set. See the examples below. The number of volume labels specified in the MOU command affects the form of the operator mount message printed on the console. The system assigns relative volume numbers sequentially, starting with volume 1 and continuing through the list of volumes. The system assumes that the volumes in the volume set are mounted physically as indicated in the MOU command. However, only relative volume 1 is verified at mount time; the verification of subsequent volumes is performed when these volumes are needed.

The file set identifier, the file section number, and the file sequence number in HDR1 file labels are used by the file system to identify individual volumes in a set of volumes.

The following are examples of MOU commands for volume sets.

MCR>MOU MT0:1STVOL	1STVOL is the first volume of a volume set. It is mounted on MT0 which is the unit on which all subsequent volumes are to be mounted.
MCR>MOU MT(0,1):(1STVOL,2NDVOL)	1STVOL is mounted on MT0. 2NDVOL either is or will be mounted on MT1. If 2NDVOL is not mounted when required, a message requesting operator action is printed on the console. Not available in MCR mode.

8.3 Dismounting a Volume Set

Use the DMO MCR command to dismount a volume set. Keep the following information in mind when dismounting a volume set:

1 All units assigned to the volume set are dismounted by the dismounting of any unit assigned to the volume set.

2 Individual DISMOUNT COMPLETE messages are printed on the operator's console for each unit assigned to the volume set.

3 All units are rewound and placed off-line unless the unit is currently off-line or rewinding.

8.4 Operator Interaction With the Magnetic Tape File System

The magnetic tape file system issues messages to the console output device (CO) through the file system message task (F11MSG). These messages are requests for operator action. The operator is expected to perform the requested action for the specified tape unit, then place the unit online.

The magnetic tape file system automatically detects that the requested unit has been placed online and restarts the process that was attempted before the request was issued.

9 Utilities

9.1 Utilities Available on IAS

IAS provides several kinds of utilities to enable you to work with different types of files and media. This chapter briefly describes each utility and refers you to the manual that fully describes the utility.

You can invoke all utilities from MCR.

9.2 Editing Utilities

IAS supports four editing programs for creating and editing text and source files:

- Line Text Editor (EDI)
- DEC Editor (EDT)
- Source language input program (SLP)
- Keypad editor (KED or K52)

9.2.1 Line Text Editor (EDI)

EDI is a line-oriented, interactive editor used to create and modify text and source files.

See the *RSX-11M/M-PLUS Utilities Manual* for a full description of EDI.

9.2.2 DEC Editor (EDT)

EDT is an interactive text editor. EDT has two features that distinguish it from EDI:

1 EDT provides unlimited access to an entire file at a time.

2 EDT provides character mode editing for use with VDUs. Character Mode allows you to edit at the character and word levels, as well as at line level.

See the *EDT Editor Manual* for a full description of EDT.

9.2.3 Source Language Input Program (SLP)

SLP is a batch-oriented editor used to maintain source files. See the *RSX-11M/M-PLUS Utilities Manual* for a full description of SLP.

9.2.4 Keypad Editor (KED or K52)

The Keypad Editor is an interactive editor used to create and modify text files for users with the FMS-11 optional software. The Keypad Editor has two versions:

1 KED—Used with VT100 terminals.

2 K52—Used with VT52 terminals.

See the *PDP-11 Keypad Editor User's Guide* for a full description of the Keypad Editor.

9.3 File Manipulation Utilities

Among other things, IAS supports two file manipulation utilities to enable you to copy and spool files and transfer files between volumes.

9.3.1 Peripheral Interchange Program (PIP)

PIP copies files and performs several file control functions, for example, deleting, renaming, listing, concatenating, spooling, and unlocking.

See the *RSX-11M/M-PLUS Utilities Manual* for a full description of PIP.

9.3.2 File Transfer Program (FLX)

FLX transfers files from one volume to another, and converts the format of the file being transferred, if necessary. FLX transfers files between Files-11, DOS-11, and RT-11 volumes, with some restrictions.

See the *RSX-11M/M-PLUS Utilities Manual* for a full description of FLX.

9.4 Volume Maintenance Utilities

IAS supports six maintenance utilities that enable you to (among other jobs) back up files onto volumes, locate bad blocks on the volumes, and verify the contents of volumes.

9.4.1 Bad Block Locator Utility (BAD)

BAD determines the number and location of bad blocks on a volume (including magnetic tape). You can use the information gathered from running BAD on a volume in different ways when that volume is initialized.

See the *RSX-11M/M-PLUS Utilities Manual* for a full description of BAD.

9.4.2 Backup and Restore Utility (BRU)

BRU enables you to backup and restore Files-11 volumes to ensure that a copy of the files is available in case the original files are destroyed. BRU transfers files to, and retrieves files from, backup volume(s). BRU is faster than DSC (see Section 9.4.3 or PRESRV in most areas. In addition, BRU compresses data and provides incremental backups where necessary, without having to initialize the volumes.

See the *RSX-11M/M-PLUS Utilities Manual* for a full description of BRU.

9.4.3 Disk Save and Compress Utility (DSC)

DSC copies Files-11 disk files to disk or tape and from DSC-created tape back to disk. While copying the files, DSC also compresses the data storage area, and writes files in contiguous blocks, unless it encounters a bad block. DSC can be run either on-line or stand-alone.

See the *RSX-11M/M-PLUS Utilities Manual* for a full description of DSC.

9.4.4 Disk Volume Formatter (FMT)

FMT formats and verifies DB:, DK:, DM:, DP:, DR:, and DY: disk volumes. FMT can perform the following tasks:

- Write complete headers for each sector of a disk.
- Verify the headers it writes.
- Set the density for DY: floppy disks.
- Set the maximum error limit for a disk pack and terminate processing when the limit is reached.
- Enable spawning of the BAD utility.

See the *RSX-11M/M-PLUS Utilities Manual* for a full description of FMT.

9.4.5 File Structure Verification Utility (VFY)

VFY verifies the consistency and validity of the file structure on a Files-11 volume.

See the *RSX-11M/M-PLUS Utilities Manual* for a full description of VFY.

9.5 Programming Utilities

IAS supports two programming utilities. The utilities enable you to work with library files and to examine file contents.

9.5.1 Librarian Utility Program (LBR)

LBR is a library maintenance program that creates, updates and modifies macro, object, and universal library files.

See the *RSX-11M/M-PLUS Utilities Manual* for a full description of LBR.

9.5.2 File Dump Utility (DMP)

DMP produces a printed listing of the contents of a file or volume so that you can examine file contents. DMP also provides options that control the format of the contents.

See the *RSX-11M/M-PLUS Utilities Manual* for a full description of DMP.

9.6 Program Maintenance Utilities

IAS supports three program maintenance utilities. They enable you to modify, patch and compare files.

9.6.1 File Compare Utility (CMP)

CMP compares two text files, record by record, and lists the differences between the two files.

See the *RSX-11M/M-PLUS Utilities Manual* for a full description of CMP.

9.6.2 Object Module Patch Program (PAT)

PAT is an object module patch utility that updates, or patches, a relocatable binary object module.

See the *RSX-11M/M-PLUS Utilities Manual* for a full description of PAT.

9.6.3 Task/File Patch Program (ZAP)

ZAP is a patch utility that examines and directly modifies locations in a task image or data file.

See the *RSX-11M/M-PLUS Utilities Manual* for a full description of ZAP.

A MCR Command Summary

Table A–1 MCR Command Summary

Command	P/N[1]	Format	Parameters	Mode
ABO	N	ABO taskname[,taskname,...]		
		PDS>> ART taskname[,taskname...]		MCR only
ABO	P	ABO taskname[/TI=dev]		
		PDS>> ART taskname[/TI=dev]		MCR only
ACT	N	ACT [taskname] [switch(es)]	switches: /ALL - All tasks for all terminals /FU - Full listing /SH - Short listing /TI=dev - All tasks for specified TI	
ALT	P	ALT taskname[/TI=dev] /PRI=nnn[,taskname[/TI=dev] /PRI =nnn,...]		
BOO	P	Format 1: BOO		
		BOO>		
		Format 2: BOO filespec		
		Format 3: BOO /WB		
		Format 4: BOO filespec/WB		
BYE	N	BYE		
CAN	P	CAN taskname[/TI=dev] [,taskname [/TI=dev],...]		
CON	N	CON taskname[,taskname,...]		
		PDS>>CRT taskname[,taskname,...]		MCR only
DCL	N	PDS> MCR		MCR mode only
		PDS>> DCL		
DEM	N	With MCR prompt: DEM [mk] or DEM [n] reset start address		

[1] Privileged or Nonprivileged

Table A–1 (Cont.) MCR Command Summary

Command	P/N[1]	Format	Parameters	Mode
		Without MCR prompt: G mK or G n reset size of area referred to by one column		
		C clear screen and renew display		
		E mK or E n change size of memory area displayed		
		E ALL Display all memory		
		I n Update display every n seconds		
		X Exit to MCR (or PDS)		
		Ctrl/Z Exit to MCR (or PDS)		
DIS	P	DIS taskname[,taskname,...]		
DMO	N	DMO dev:[volumelabel] [/UIC=[uic]][/LOCK]/NOUNL		
DMO	P	Same as nonprivileged except that only a privileged user can dismount the system volume.		
ENA	P	ENA taskname[,taskname,...]		
FIX	P	FIX taskname[/TI=dev] [,taskname[/TI=dev],...]		
HEL	N	HEL [uic]		Not in MCR

[1] Privileged or Nonprivileged

Table A–1 (Cont.) MCR Command Summary

Command	P/N[1]	Format	Parameters	Mode
HOM	P	HOM ddn:volume-label[/switches]	switches: /UIC=[m,n] /PRO=[system,owner,group,world] /MXF=maximum number of files allowed on this volume /EXT=default file extension size in blocks /FPRO=[system,owner,group,world] /CHA=[characteristic words] ATCH or DCF or both /WIN=default window size for file /LRU=number of directories to keep preaccessed /DENS=[RX02 floppy disk density] LOW or HIGH /VI show current value of all switches /NAME=volume-label	

[1] Privileged or Nonprivileged

Table A–1 (Cont.) MCR Command Summary

Command	P/N[1]	Format	Parameters	Mode
INI	P	INI dev:[volumelabel] [/keyword(s)]	keywords: /UIC=[m,n] /PRO=[system,owner,group,world] /MXF=maximum number of files allowed on this volume /EXT=default file extension size in blocks /FPRO=[system,owner,group,world] /CHA=[characteristic words] ATCH or DCF or both /INF=number of file headers in file index /WIN=default window size for file /LRU=number of directories to keep pre-accessed /DENS=magnetic tape or RX02 floppy disk density 800 or 1600 for magnetic tapes LOW or HIGH for RX02 floppy disks /INDX=index file position option BEG - beginning of volume MID - middle of volume END - end of volume BLK:nnn - logical block number /BAD=option (Initialization of bad block file) [MAN] Accept a bad block list specified from the terminal. [AUTO] Read the bad-block descriptor file on the last track of the volume created by the manufacturer's diagnostic routines or the bad block locator utility (BAD). [NOAUTO] Ignore the bad-block descriptor file and performs no bad-block processing. [AUTO,MAN] Read the bad-block file and, when done, accept blocks specified from the terminal. [OVR] Include the last track in the BADBLK.SYS file. [OVR,MAN] Override the manufacturers bad block descriptor file and accept blocks specified from the terminal. /VI show current value of all switches	

Table A–1 (Cont.) MCR Command Summary

Command	P/N[1]	Format	Parameters	Mode
INS	P	INS filespec[/keyword(s)] [,filespec[/keyword(s)],...]		
		INS @indirect	keywords: /PAR = partition name /PRI = priority-number (decimal) /TASK = taskname /POOL = pool-limit (decimal) /UIC = [uic] /RUN[=REM] (run-remove) /TIM = allowable run time expressed as nH, ns, nM, or nT. /ACC = non-owner access expressed as RO - read only RW - write only NA - no access (default) /LI = library file /CM = common area /INC = task size increment /RG = a region	
LOA	P	LOA handlername[unit number][:]		
LOG	N	LOG comment-line		Not MCR
		; comment-line		
		! comment-line		
LUN	N	LUN taskname[,taskname,...]		
MCR	N	Format 1: PDS> [$]MCR PDS>> [MCR command string]		PDS only
		Format 2: PDS> [$]MCR [MCR command string]		
MEM	P	MEM [taskname][/switches] [,taskname[/switches]][,...]	switches: /AL - all such tasks /TI =TTn - all tasks with this TI	

[1] Privileged or Nonprivileged

Table A–1 (Cont.) MCR Command Summary

Command	P/N[1]	Format	Parameters	Mode
MOU	N	MOU dev:volumelabel [/keywords(s)]	keywords: /CHA=[characteristic word] FOR - foreign volume ATCH - device can be attached exclusively by one task DCF - device control functions allowed /UNL (volumes index file is unlocked) /DENS = magnetic tape density 800 or 1600 /ACP = taskname (file processor for the volume) /EXT = default file extend increment in blocks /FPRO = [default file protection] /LRU = number or directories to keep pre-accessed /OVRFSID (override the set identifier check) /OVREXP (override the expiration date check) /WIN = number of file extents to be kept in each window block	
		Format for mounting multivolume magnetic tape: MOU MT(n1[,n2,...,nn]):(label1[,label2,...,labeln]) [/keyword(s)]		Not MCR
MOU	P	MOU dev:volumelabel [/keyword(s)]	keywords: All of the above-mentioned keywords, plus the following: /UIC =[uic] /OVR (override the volume label check) /PRO = [system,owner,group,world]	
OPE	P	OPE memory-address[+n][/option(s)]	options: /TASK = taskname[/TI=dev] /PAR = partition name /KNL (specifies kernel virtual address)	

[1] Privileged or Nonprivileged

Table A–1 (Cont.) MCR Command Summary

Command	P/N[1]	Format	Parameters	Mode
OPR	N	OPR dev:/switch[,dev:/switch,...]	switches: /SP - Start printing /AB - Abort printing /ST - Stop printing /RS - Resume printing /RS:T - Resume printing from last encountered form feed. /RS:TOF - Resume printing from top-of-file	
OPR	P	OPR dev:/switch[,dev: /switch,...,dev/switch]	switches: All of the above mentioned switches plus the following: /CHG - Change forms /FO:n - Set the forms type to n /RE - Make all queue requests inactive and dispatch those with matching forms types.	
PWD	N	PWD [ufd] PASSWORD>password		Not MCR
PWD	P	Same as non-privileged except the privileged user can change the password for any UFD on any volume not protected against system access.		
QUE	N	1. Queue a File For Printing: QUE [dev:=]infile-1 [,infile-2,...,infile-n][/switch(es)] or QUE @indirect	switches: /PR:nnn—Set the priority of output selection to nnn (default /PR:50) /FO:n—Set form type to n (default /FO:0) /CO:nn—Set number of copies to nn (default /CO:1) /DE—Delete file after printing (default /-DE) /TE—Test for forms alignment /AF:[dd-mmm-yy]hh:mm—After the specified time is reached, file is queued for immediate processing (default /AF:0:00)	
		2. List Elements in Queue: QUE /LI		
		3. List All Elements in Queue: QUE /AL		

[1] Privileged or Nonprivileged

Table A–1 (Cont.) MCR Command Summary

Command	P/N[1]	Format	Parameters	Mode
		4. Kill (Delete) Elements in Queue: QUE /KI:seq[:seq:...:seq]		
		5. Modify Elements in Queue: QUE [dev:=]/MO:seq[:seq:...:seq] [/switch(es)]	switches: /PR:nn—Change priority to nnn. /FO:n—Change form type to n. /CO:nn—Change number of copies to n. /DE—Change delete/preserve indicator to delete. /AF:[dd-mmm-yy]hh:mm—After the specified time is reached file is queued for immediate processing (default /AF:0:00)	
REA	P	REA taskname lunlist devunit:		
RED	P	RED new-dev[:]=old-dev[:]		
REM	P	REM name1[/switch] [,name2[/switch],...] or REM @indirect		
RES	N	RES taskname[,taskname-1,...]		
RES	P	RES taskname[/TI=dev] [,taskname-1[/TI=,...]		
RUN	N	The format in MCR mode is as shown below, except that RUN is replaced by RRT: RUN filename[/TIM=NX] ESC or RUN $filename ESC		
RUN	P	The format in MCR mode is as shown below for each of the different forms of the RUN command, except that RUN is replaced by RRT. Request: RUN taskname[/options] [,taskname[/options],...] ESC	options: /PRI=priority number (decimal) /PAR=partition name /TI=terminal /UIC=[user identification code]	
		Execute: RUN taskname/MEM[options] [,taskname/MEM[/options],...] ESC	options: /PRI=priority number (decimal) /PAR=partition name /TI=Terminal /UIC=[user identification code]	

[1] Privileged or Nonprivileged

Table A–1 (Cont.) MCR Command Summary

Command	P/N[1]	Format	Parameters	Mode
		Run: RUN taskname time[/options] [,taskname time[/options],...] ESC	options: /PRI=priority number (decimal) /PAR=partition name /RSI=reschedule interval expressed as nH, nM, nS or nT /TI=terminal /UIC=[user identification code]	
		Schedule: RUN taskname time[/options] ESC	options: /PRI=priority number (decimal) /PAR=partition name /RSI=reschedule interval expressed as nH, nM, nS, or nT /TI=terminal /UIC=[user identification code]	
		Synchronize: RUN taskname startime +time[/options] ESC	options: /PRI=priority number (decimal) /PAR=partition name /RSI=rescheduling interval expressed as nH, nM, nS, or nT /TI=terminal /UIC=[user identification code]	
SAV	P	SAV [/switch]	switches: /MOU=dev:[dev:...dev:]—save system with dev: mounted /LOG=TTn:[TTn:...:TTn:]—save system with specified terminals logged on. /NO, /NOX, /NOT, /NOXT—all inhibit memory expansion/truncation on reboot. /NOIN—inhibit automatic reinstall on reboot. /LGOUT—log out system console after boot. /CMD—execute SY:[1,1]STARTUP.CMD after boot. /NOTIM—prohibit IAS time and date prompt after boot.	

[1] Privileged or Nonprivileged

Table A–1 (Cont.) MCR Command Summary

Command	P/N[1]	Format	Parameters	Mode
SET	P	SET [/keyword(s)]	keywords: /UIC=[group,member]—Change terminal UIC. This keyword must not be used in MCR mode, see SET UIC or SET DEFAULT in the *IAS PDS User's Guide.* /TMO=nnnnn—Change MCR timeout value to nnnnn. /PRV=dev:[dev:...:dev:]—Device is defined as a privileged terminal device. /SLV=dev:[dev:...:dev:]—Device is defined as a slave terminal device. /-PRV=dev:[dev:...:dev:]—Device is defined as a nonprivileged terminal device. /-SLV=dev:[dev:...:dev:]—Device is defined as a non-slave (command) terminal device. /SP=dev:[dev:...:dev:]—Device is defined as output spooled. /-SP=dev:[dev:...:dev:]—Device is no longer defined as output spooled. /CLI=xxx:ddn[:...]—Define name of CLI to be allocated to a terminal. /CAC=n—Number of parity errors in cache group allowed in one minute. /CAC=OFn—Cache group n turned off. /CAC=ONn—Cache group n turned on. /SWR—Display software switch register. /SWR=nnnnnn—Set octal value in software switch register. /SWR=SET:m[:n]—Set bits m [through n] in SWR. /SWR=CLR:m[:n]—Clear bits m [through n] in SWR. /MAXEXT=nnnn—Set maximum task extension size in octal number of 32- word blocks. /MAXEXT=nnK—Set maximum task extension size in decimal K words. /MAXEXT—Display current extension limit. /NOMAXEXT—Enable task to extend to the maximum imposed by mapping constraints.	

Table A–1 (Cont.) MCR Command Summary

Command	P/N[1]	Format	Parameters	Mode
SYS	N	SYS [/switch[:opt]]	switches: /ATL—List names of tasks in the active task list. /CKQ—List name of task in the clock queue. /GCD—List names of libraries and common areas in the global common directory. /IRQ[:opt]—List names of tasks with entries in the I/O request queue. /CTL[:opt]—List names of tasks in the checkpointable task list. /FTL—List names of tasks in the fixed tasks list. /SRQ—List names of tasks that have SENDS queued for them. /ASQ—List names of tasks in the ATL that have ASTs queued for them. /SGA—List names of tasks in STD and the SGAs to which each is linked. /BRF—List names of tasks in the ATL, MRL, or CKQ. /FUL—List all the above except /SGA. /TAS—Same as TAS command. /DEV—Same as DEV command. /PAR—Same as PAR command. /COM—Same as COM command.	
SWA	P	SWA ddnn:/LE:n[/switches]	switches: /DV—Dedicates device as a swap volume. /[-]BA—Indicates that bad block information is [not] to be used. /RT—Swap file to be used for real-time tasks only. /NU:n—Allocates the number n to the swap file.	

[1] Privileged or Nonprivileged

Table A–1 (Cont.) MCR Command Summary

Command	P/N[1]	Format	Parameters	Mode
TER	N	TER/[NO]option - Set user's terminal characteristic. TER/option:value—Set characteristic value. TER/terminaltype[/DS]—Set all user's terminal characteristics excluding or including default speed. TER @indirect—Indirect command file specification.		
TER	P	As for TER (Non-privileged) but with TER replaced by: TER TTa:[TTb:...TTn:] /options—To set named terminal, and with additional option /DEFAULT, or TER TTa:[TTb:...TTn:] /terminaltype/DS[/DEFAULT]—To set characteristics other than the speed to default values. /DS also sets the speed to the default value.		
TIM	N	TIM		
TIM	P	TIM [time] [date]		
UNF	P	UNF taskname[/TI=dev] [,taskname[/TI=DEV],...]		
UNL	P	UNL handlername[unit number][:]		
UFD	P	UFD dev:[uic][/switch]	switches: /PRO=[system,owner,group,world] /ALLOC=number-of-entries	

[1] Privileged or Nonprivileged

Table A-1 (Cont.) MCR Command Summary

Command	P/N[1]	Format	Parameters	Mode
UTL	P	UTL/switch(es)	switches: /AM:n—Set memory size for CPU time allocation to n. /AT:n—Set CPU time allocation factor. /BQ:n—Set quantum for batch level. /BS:n—Set time between batch schedules. /DI—Disable scheduler. /EN[:par]—Enable scheduler in partition par. /IT:n—Set idle time to n ticks. /LV:n—Set number of levels for scheduler. /MT:n—Set maximum task size. /PT:n—Set size of promotion table. /QC:n—Set quantum constant for CPU time allocation. /SH—Display scheduler parameters. /SP:n—Set number of ticks between scheduler promotion. /TF:m:n—Set time factor for scheduling level m to n ticks. /TP:n—Set timesharing ATL priority. /TS:n—Set maximum CPU time slice.	
WHO	N	WHO		

[1] Privileged or Nonprivileged

B MCR Error Messages

B.1 MCR Error Messages

The error messages in this appendix are listed in alphabetic order by issuing task. The only exceptions are the messages issued by the task MCRERR. This task prints the user-entered command that caused the error and then, on the following line, prints the error message. Messages with no three-letter prefix can be identified as coming from MCRERR. These messages are listed in Section B.2.

ACT—INVALID ADDRESS IN A.TI

Explanation: The active task list contains a TI that has a invalid address. Notify the Digital software specialist.

ACT—INVALID DEVICE

Explanation: The device mnemonic, specified by the user, is either invalid or specified in a nonstandard format.

ACT—INVALID SWITCH

Explanation: The user specified a switch that ACT does not recognize. Either the switch is not an ACT switch, or it was specified in the wrong format.

ACT—INVALID SWITCH COMBINATION

Explanation: The user specified switches that are mutually exclusive; see the description of the ACT command in Chapter 6 to determine which switches can be used together.

ACT—INVALID TASK NAME

Explanation: The user specified a task name that either contains more than 6 alphanumeric characters or one that contains a non-alphanumeric character.

ACT—NO TASKS ACTIVE ON TERMINAL

Explanation:

or

ACT—TASK NOT ACTIVE

Explanation: Self-explanatory.

ACT—TOO MANY TASKS FOR BUFFER

Explanation: ACT can process a maximum of 64 active tasks, and the active task list contains more than 64. Only the first 64 tasks in the queue are listed at the user terminal.

To make ACT handle more than 64 active tasks, the system manager must task build ACT to accommodate more tasks. This is accomplished by modifying ACT's size, and the size of its processing buffer (MAXTSK).

ACT—THE /FU SWITCH REQUIRES A TASK NAME

Explanation: Self-explanatory.

BOO—ASSIGN LUN ERROR—CODE -nn

Explanation: The system is unable to assign a LUN to the device being booted. The reason for the failure is determined by -nn, where -nn is a system standard error code. See Appendix D.

BOO—CANNOT FIND SYSTEM DEVICE PUD

Explanation: The system device (SY) PUD cannot be found. This will occur if SY was redirected to itself, or if the PUD entry itself is corrupt.

Redirect SY to a valid system device, then retry the command. If the error recurs, call the Digital software specialist.

BOO—COMMAND I/O ERROR CODE, CODE = -nn

Explanation: The TTY handler is unable to obtain a command line from the user terminal. See Appendix D for code -nn.

Notify the Digital software specialist.

BOO—COMMAND SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

BOO—ERROR CLOSING FILE filename CODE = -nn

Explanation: The file system is unable to close the specified file. The reason for the failure is determined by -nn, where -nn is a system standard error code. See Appendix D.

BOO—FILE NOT AN IAS IMAGE

Explanation: Self-explanatory.

BOO—FILE NOT CONTIGUOUS

Explanation: The file is either corrupt, or not an IAS system image file.

BOO—GET LUN ERROR—CODE = -nn

Explanation: The system is unable to get a LUN for the device being booted. The reason for the error is determined by -nn, where -nn is a system standard error code. See Appendix D.

BOOT—INDIRECT COMMAND SYNTAX ERROR

Explanation: The user specified an indirect command file specification in a format that does not conform to syntax rules.

Refer to the command description for the correct syntax.

BOO—INDIRECT FILE DEPTH EXCEEDED

Explanation: The indirect command file, specified by the user, has an indirect command file specification as one of its commands. BOO can only process one indirect command file at a time.

Recreate the indirect command file, or enter the commands manually.

BOO—INDIRECT FILE OPEN FAILURE, CODE = -nn

Explanation: BOO is unable to open the specified indirect command file. The reason for the failure is determined by -nn, where -nn is a system standard error code. See Appendix D.

BOO—INVALID SWITCH

Explanation: The user specified a unrecognizable switch. The only switch recognized by BOO is /WB.

BOO—ILLEGAL DEVICE/VOLUME

Explanation: The device mnemonic, specified by the user, does not conform to device syntax.

BOO—ILLEGAL DEVICE/UNIT NUMBER

Explanation: Either the device mnemonic is illegal; not recognized by the system, or the unit number exceeds the prescribed range for the device.

BOO—ILLEGAL FILENAME

Explanation: The user specified a file name that either contains a non-alphanumeric character, or one that is greater than 9 alphanumeric characters in length.

BOO—ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

Explanation: Self-explanatory.

BOO—ILLEGAL GET COMMAND LINE ERROR, CODE -nn

Explanation: The system, while processing a get command line directive, encountered an inexplicable error condition. Either the system software is corrupt, or the hardware has failed. -nn is the get command line error code.

Notify the Digital software specialist.

BOO—ILLEGAL UIC

Explanation: The UIC specified by the user is either illegally formatted or nonexistent.

BOO—I/O ERROR IAS IMAGE FILE filename, CODE = -nn

Explanation: The file system encountered an error while reading the specified file. The reason for the failure is determined by -nn, where -nn is a system standard error code. See Appendix D.

BOO—I/O ERROR (CODE = nn) WRITING BLOCK 0

Explanation: Probable hardware error; the QIO issued by BOO was valid, but the write to block 0 failed. The reason for the failure is determined by -nn, where -nn is a system standard error code. See Appendix D.

BOO—NO WILDCARDS ALLOWED

Explanation: Self-explanatory.

BOO—NOT BOOTABLE DEVICE

Explanation: The specified device is not a mass storage device.

BOO—ONLY ONE FILESPEC (OUTPUT) ALLOWED

Explanation: Self-explanatory.

BOO—OPEN FAILURE FILE filename, CODE = -nn

Explanation: BOO failed to open the specified file. The reason for the failure is determined by -nn, where -nn is a system standard error code. See Appendix D.

BOO—QIO FAILURE (CODE = -nn) WRITING BLOCK 0

Explanation: A QIO issued by BOO was rejected; BOO may be corrupted. The reason for the failure is determined by -nn, where -nn is a system standard error code. See Appendix D.

BOO—SYSTEM IS NOT QUIESCENT

Explanation: The user attempted to execute BOO while tasks were still active in the system.

Execute ACT to determine which tasks remain active. Terminate the active tasks or wait for them to finish executing, then retry the BOO command.

CON—COMMAND SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

CON—ILLEGAL ERROR/SEVERITY CODE n1 n2 n3

Explanation: CON failed in its attempt to issue a message to the user terminal. n1 contains the value in R1 at the time of the failure; n2 contains the value in R2; and n3 contains the return address after calling the error message routine.

NOTE: The high order byte in R1 reflects the severity code between 0 and 2.

The low order byte in R1 reflects the error message CON was trying to issue.

Notify the Digital software specialist.

CON—TASK taskname NOT ACTIVE

Explanation:

or

CON—TASK taskname NOT IN SYSTEM

Explanation: Self-explanatory.

CON—UNABLE TO READ MCR COMMAND LINE

Explanation: CON failed to obtain the command line with which it was invoked. This will occur if CON is invoked using the RUN command, as opposed to the prescribed method.

Retry the command using the prescribed format. If the failure recurs, notify the Digital software specialist.

DMO—COMMAND I/O ERROR

Explanation: The DMO task is unable to obtain a command line from the user terminal.

Notify the Digital software specialist.

DMO—DEVICE NOT IN SYSTEM

Explanation: The user specified a device that was not included as a usable device when the system was generated. That is, there is no PUD entry for the device. You possibly typed the device mnemonic incorrectly.

DMO—DEVICE NOT MOUNTABLE

Explanation: The user specified a device that cannot be mounted; for example, a card reader.

DMO—HANDLER NOT RESIDENT

Explanation: The user specified a device for which the appropriate device handler is not resident.

Load the required device handler task and retry the command.

DMO—ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

Explanation: Self-explanatory.

DMO—INVALID UIC

Explanation: The UIC specification, entered by the user, either does not match the volume's UIC, or it was entered in the wrong format.

DMO—INVALID VOLUME LABEL

Explanation: The volume label, specified by the user, does not match the volume's label.

DMO—SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

DMO—VOLUME BUSY - TRY AGAIN

Explanation: There are open files on the volume.

Wait for the files to be properly closed, then retry the command.

DMO—VOLUME NOT MOUNTED

Explanation: Self-explanatory.

HEL—DIRECTIVE FAILURE

Explanation: An undefinable error occurred during the execution of HEL.

Retry the command. If the error recurs, notify the Digital software specialist.

HEL—ILLEGAL PASSWORD

Explanation: System access is denied. The specified password does not match the password stored in the requested UFD.

HEL—READ FAILURE ON DIRECTORY ENTRY

Explanation: The specified UIC does not correspond to a UFD on the system device; the system device is not mounted or the hardware has failed.

Notify the system manager.

HEL—SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

INI—BAD BLOCK FILE CORRUPT—DATA IGNORED

Explanation: The volume's bad block data is inconsistent. Re-execute BAD, then retry the INI command.

INI—ALLOCATION FOR SYS FILE EXCEEDS VOLUME LIMIT

Explanation: Either the user specified /INDX=BLK:n where the value for n was too high; or the user specified /INDX=END and the value specified in /INF=n was too large.

Retry the command with a correct value specified.

INI—BAD BLOCK HEADER I/O ERROR

Explanation: INI is unable to retrieve the bad block data area from the specified disk volume.

Retry the command. If the problem persists, perform the following:

1 Rerun BAD.

2 Rerun INI with /BAD=[AUTO] specified.

If the problem still persists, the hardware has failed. Notify field service.

INI—BAD BLOCK FILE FULL - TASK ABORTED

Explanation: Either the bad block file was full from a previous BAD execution, or the user specified more bad blocks than the bad blocks file is able to store. The disk is not usable.

INI—BITMAP TOO LARGE - INCREASE CLUSTERFACTOR

Explanation: The number of bits in the bit map is too large. Increase the Cluster factor, that is, the number of blocks represented by one bit.

INI—BLOCK(S) EXCEED VOLUME LIMIT

Explanation: The user specified a range of bad blocks that exceeds the range of blocks available on the desired volume.

INI—BOOT BLOCK WRITE ERROR

Explanation: INI has failed in its attempt to format the disk's bootstrap block. One of the following conditions exists:

1 The disk device is off line.

2 The bootstrap block (block 0) is bad.

3 The device is write locked.

4 The hardware has failed.

INI—CHECKING ddn

Explanation: INI is looking for bad block data on device ddn. This message is for information purposes only.

INI—CHECKPOINT FILE HEADER I/O ERROR

Explanation: INI is unable to write the volume's checkpoint file header area. One of the following conditions exists:

1 The device is off line.

2 The device is not write enabled.

3 The volume has a bad block.

4 The hardware has failed.

INI—COMMAND I/O ERROR

Explanation: The system is unable to get a command line from the user terminal.

Notify the Digital software specialist.

INI—DATA ERROR

Explanation: The user entered bad block data in an incorrect format.

Reenter the command using the correct format.

INI—DEVICE NOT IN SYSTEM

Explanation: The specified device was not included as a usable device when the system was generated. That is, there is no PUD entry for the device. You possibly typed the device mnemonic incorrectly.

INI—DEVICE NOT READY

Explanation: The requested device is not completely cycled up.

Wait for the device to become ready, then retry the command.

INI—DEVICE WRITE LOCKED

Explanation: Set the device write enable switch, then retry the command.

INI—DISK IS ALIGNMENT CARTRIDGE - TASK ABORTED

Explanation: The disk has been recognized as an alignment cartridge and cannot be initialized.

INI—DUPLICATE BLOCK(S) FOUND

Explanation: The user specified bad block data for blocks that already appear in the bad blocks file.

INI—FAILED TO ATTACH DEVICE

Explanation: The system has failed to gain control of the device unit.

Determine whether another task has attached the unit. If so, wait until the task exits or abort the task.

INI—FAILED TO READ BAD BLOCK FILE - TASK ABORTED

Explanation: The bad block file in the last track of the disk could not be read.

INI—HANDLER NOT RESIDENT

Explanation: The required device handler is not resident.

Load the required device handler and retry the command.

INI—HOME BLOCK ALLOCATE WRITE ERROR

Explanation: INI has failed in its attempt to allocate space for the volume's home block. One of the following conditions exists:

1 The device is off line.

2 The device is not write enabled.

3 The volume has bad blocks.

4 The hardware has failed.

INI—HOME BLOCK WRITE ERROR

Explanation: INI has failed in its attempt to write the volume's home block data. One of the following conditions exists.

1 The device is off line.

2 The device is not write enabled.

3 The volume has bad blocks.

4 The hardware has failed.

INI—INDEX FILE BIT MAP I/O ERROR

Explanation: INI has failed in its attempt to write the volume's index bit map. One of the following conditions exists:

1 The device is off line.

2 The device is not write enabled.

3 The volume has bad blocks.

4 The hardware has failed.

INI—INDEX FILE HEADER I/O ERROR

Explanation: INI has failed in its attempt to write the index file header. One of the following conditions exists:

1 The device is off line.

2 The device is not write enabled.

3 The volume has bad blocks.

4 The hardware has failed.

INI—ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

or

INI—ILLEGAL KEYWORD VALUE

Explanation: Self-explanatory.

INI—ILLEGAL UIC

Explanation: The user specified a UIC that either does not conform to UIC syntax, or is not in the range of valid UICs.

INI—I/O ERROR SIZING DEVICE

Explanation: Irrecoverable error when sizing the device.

INI—MAGTAPE DEVICE ERROR

Explanation: The specified magnetic tape device cannot be accessed for one of the following reasons:

1 The device is off line.

2 The device is not write enabled.

3 The hardware has failed.

INI—MAGTAPE LABEL MUST BE SPECIFIED

Explanation: Self-explanatory.

INI—MAGTAPE WRITE ERROR

Explanation: INI is unable to initialize a magnetic tape volume for one of the following reasons:

1 The device is off line.

2 The device is write protected.

3 The hardware has failed.

INI—MFD FILE HEADER I/O ERROR

Explanation: INI is unable to write the MFD file header for one of the following reasons:

1 The device is off line.

2 The device is not write enabled.

3 The volume has bad blocks.

4 The hardware has failed.

INI—MFD WRITE ERROR

Explanation: INI is unable to write the MFD file for one of the following reasons:

1 The device is off line.

2 The device is not write enabled.

3 The volume has bad blocks.

4 The hardware has failed.

INI—NO BAD BLOCK DATA FOUND

Explanation: The /BAD=[AUTO] option was used but no bad block data could be found on the disk.

INI—NOT FILES-11 DEVICE

Explanation: Self-explanatory.

INI—NULL FILE HEADER I/O ERROR

Explanation: The hardware has failed. Notify field service.

INI—PRIVILEGED COMMAND

Explanation: A privileged command has been entered by a non-privileged user.

INI—STORAGE BIT MAP FILE HEADER I/O ERROR

Explanation: INI is unable to format the bit map for one of the following reasons:

1 The device is off line.

2 The device is not write enabled.

3 The volume has bad blocks.

4 The hardware has failed.

INI—SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

INI—UNIT DOES NOT SUPPORT 1600 BPI

Explanation: The /DENS=1600 switch was specified for a drive that does not support this density.

INI—UNRECOGNIZED DISK TYPE

Explanation: The system does not recognize the disk type specified by the user.

Enter the correct disk type.

INI—VOLUME MOUNTED

Explanation: INI cannot process a logically mounted volume.

INI—WARNING - BLOCK 0 IS BAD

Explanation: Block 0, the bootstrap block, is bad. Disk cannot be used as a system disk.

If the device mnemonic was correctly specified, execute DMO to dismount the volume, then retry the command; otherwise, retry the command with the device mnemonic correctly specified.

INS or INV—COMMAND I/O ERROR

Explanation: INS or INV is unable to get a command line from the user terminal.

Notify the Digital software specialist.

INS or INV—COMMAND SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

INS—DATE MISMATCH GLOBAL AREA area-name FILE filename

Explanation: The task being installed is linked with a shareable global area that is a different version from the one currently installed.

Remove the current version; replace it with the latest version, then retry the command.

INS—FILE filename IS AN EXEC PRIVILEGED TASK

Explanation: Privilege violation (timesharing system only). See the IAS

System Management Guide, Chapter 7.

INS—FLAGWORD MISMATCH GLOBAL AREA area-name FILE filename

Explanation: Either the flag word field of the shareable global area is corrupt, or the corresponding field in the task image is corrupt.

Rebuild either the offending task or shareable global area, and retry the command.

INS—GLOBAL AREA ACCESS REQUEST VIOLATION FILE filename

Explanation: The task being installed is denied access to the specified shareable global area. One of the following conditions exists:

1 The task contains an illegal reference or link to the shareable global area.

2 The shareable global area is protected against access.

Rebuild the task correctly and retry the command.

INS—GLOBAL AREA NAME ALREADY IN USE

Explanation: The user attempted to install a shareable global area whose name matches one that is already installed.

Either remove the shareable global area already installed, or rebuild the new one, giving it a new name.

INS—ILLEGAL DEVICE TYPE FOR TASK

Explanation: The user attempted to install a task from a Digitaltape or floppy disk.

INS—INCOMPATIBLE SYSTEM VERSION NUMBER FILE filespec

Explanation: Check that executive privileged tasks are built under the current system version and not a previous one. (This does not apply to non-executive privileged tasks.)

INS or INV—UNKNOWN DEVICE dev: FILE filename

Explanation: A device assigned to a LUN at task build time is not one that was generated into the system. This is a diagnostic message; the task image was successfully installed. The user must, however, reassign the LUN to a valid device before running the task.

INS or INV—ILLEGAL DEVICE/VOLUME dev:

Explanation: The user specified a device that is not recognized by the system. It is either an illegal device, or the device was incorrectly specified.

INS or INV—ILLEGAL DIRECTORY

Explanation: The user specified a UFD in a format that does not conform to syntax rules.

INS or INV—ILLEGAL ERROR/SEVERITY CODE n1 n2 n3

Explanation: INS failed in its attempt to issue a message to the user terminal. n1 contains the value in R1 at the time of the failure; n2 contains the value in R2; and n3 contains the return address after calling the error message routine.

NOTE: The high order byte in R1 reflects the severity code between 0 and 2.

The low order byte in R1 reflects the error message INS was trying to issue.

Notify the Digital software specialist.

INS or INV—ILLEGAL FILENAME filename

Explanation: The specified file name contains non-alphabetic characters, or it is more than 9 characters in length.

INS or INV—ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

Explanation: Self-explanatory.

INS or INV—ILLEGAL GET COMMAND LINE ERROR CODE -nn

Explanation: The system, while processing a get command line directive, encountered an unexplainable error. Either the system software is corrupt, or the hardware has failed. -nn is the error return code.

Notify the Digital software specialist.

INS or INV—ILLEGAL HEADER DATA FILE filename

Explanation: The task image header block is corrupted; the task is unusable.

Rebuild the task, then retry the command.

INS or INV—ILLEGAL LABEL DATA FILE filename

Explanation: The task image contains inconsistent information; the file is corrupt and, therefore, unusable.

Rebuild the task, then retry the command.

INS—ILLEGAL OPTION ON SHAREABLE GLOBAL AREA filename

Explanation: You specified the /INC keyword option for a shareable global area.

Retry the command without the /INC switch specified.

INS or INV—ILLEGAL POOL LIMIT FILE filename

Explanation: The pool limit established in the task image file header is illegal; either file header is corrupt, or the value specified for the /POOL switch exceeds 255.

Rebuild the task, then retry the command.

INS or INV—ILLEGAL PRIORITY FILE filename

Explanation: The task image file header is corrupt; the file is not usable.

Rebuild the file, then retry the command.

INS or INV—ILLEGAL RE-INSTALL FILE filename

Explanation: The file filename is already installed with a different task name and certain install parameters are different.

INS or INV—ILLEGAL SWITCH

Explanation: The user specified a switch that INS does not recognize. Either the switch is not an INS switch, or it was specified in the wrong format.

INS or INV—ILLEGAL TASK STATUS WORD FILE filename

Explanation: The task status word of the task image file's header is corrupt; the file is unusable.

Rebuild the file, then retry the command.

INS or INV—ILLEGAL UIC FILE filename

Explanation: Either the UIC in the task image file's header is corrupt, or the UIC specified in the INS command line is bad.

Either rebuild the task and retry the command, or retry the command with the UIC correctly specified.

INS or INV—IMPROPER SIZE DATA FILE filename

Explanation: The task image file is corrupt.

Rebuild the task, then retry the command.

INS or INV—INDIRECT COMMAND SYNTAX ERROR

Explanation: The user entered an indirect command file specification in a format that does not conform to syntax rules.

Retry the command with the indirect command file correctly specified.

INS or INV—INDIRECT FILE DEPTH EXCEEDED

Explanation: The indirect command file which you specified has an indirect command file specification as one of its commands. INS can only process one indirect file at a time.

Recreate the originally specified indirect command file, replacing the erroneous command file specification with the actual commands it represents.

INS or INV—INDIRECT FILE OPEN FAILURE filename

Explanation: The requested indirect command file cannot be located on the specified or implied device.

Ensure that the indirect file specification was correctly entered, and retry the command.

INS or INV—I/O ERROR TASK IMAGE FILE

Explanation: INS encountered an error while processing the specified task image file. One of the following conditions exists:

1. The device is off line.
2. The device is not write enabled.
3. The volume has bad blocks.
4. The hardware has failed.

INS—LENGTH MISMATCH GLOBAL AREA area-name FILE filename

Explanation: Either the task image file's label block is corrupt, or the GCD in a shareable global area is corrupt.

Rebuild either the task image or shareable global area, then retry the command.

INS or INV—MEMORY MAPPING CONFLICT FILE filename

Explanation: Either the task's memory mapping registers or memory limits are invalid.

Rebuild the file, then retry the command.

INS or INV—NO NODE AVAILABLE FOR FILE filename

Explanation: The system is saturated with active tasks; there are no pool nodes available to run INS.

Either wait for pool nodes to become available, or notify the system manager.

INS or INV—NO ROOM IN STD FOR NEW TASK taskname

Explanation: The number of allowable installed tasks, specified at system generation, has been reached. The named task cannot be installed until some of the previously installed tasks are removed. In the case of INV, the system database or SGN1/INV have been corrupted.

Notify the system manager, if using INS. If using INV, notify the Digital software specialist.

INS or INV—NO SWAP SPACE FOR REGION, FILE filename

Explanation: There is insufficient swap space for the region SGA (/RG switch only).

Create more swap space, or do not install the region.

INS or INV—NON CONTIGUOUS TASK IMAGE FILE

Explanation: Either the file is not a task image file, or the file is corrupt.

Either retry the command with the correct file specified; or rebuild the task, then retry the command.

INS or INV—OPEN FAILURE FILE filename

Explanation: INS failed to open the specified task image file for an update. One of the following conditions exists:

1. The requested file does not exist in the specified UFD.
2. The UFD does not exist.
3. The file is protected against extend access.
4. The volume is not mounted.
5. The volume is off line.
6. The device is not write enabled.
7. The hardware has failed.

INS—SHAREABLE GLOBAL AREA HAS NO OWNING UIC filename

Explanation: All SGAs must be built or installed having an owning UIC.

Either rebuild the SGA specifying a UIC, or retry the INS command with the /UIC switch specified.

INS or INV—SPECIFIED PARTITION IS TOO SMALL FILE filename

Explanation: Self-explanatory.

INS—STARTING APR MISMATCH GLOBAL AREA area-name FILE filename

Explanation: Either the APR field in the shareable global area is corrupt, or the corresponding field in the task image is corrupt.

Rebuild either the offending task image or shareable global area and retry the command.

INS or INV—TASK IMAGE FILE filename HAS NO HEADER

Explanation: An SGA is being installed as a task (with no header).

Reinstall the SGA using the /CM switch.

INS or INV—TASK INCREASE TOO LARGE FILE filename

Explanation: Self-explanatory.

Reduce the size of the task (see the *IAS Task Builder Reference Manual* for task sizes).

INS or INV—TASK NAME ALREADY IN USE FILE filename

Explanation: The user is attempting to install a task whose name is assigned to a previously installed task.

Either remove the previously installed task replacing it with this one, or assign a different name to this task via the /TASK keyword.

INS—TASK REQUIRES MORE THAN 7 GCD'S

Explanation: The specified task image is linked to more than 7 GCDs.

Rebuild the task changing the number of GCDs to 7 or less.

INS—TASK UIC DIFFERS FROM YOUR UIC FILE filename

Explanation: Either the /UIC switch has been specified or the task has a default UIC which is different from the user's UIC, and the user is not executive privilege privileged (timesharing system only).

INS—TICKS PER SECOND IN SCOM INVALID

Explanation: This is a system software problem; notify the Digital software specialist.

INS or INV—TIMEOUT SPECIFIED ON NON ACCOUNTABLE TASK

Explanation: Self-explanatory. This is a diagnostic message; the install was successfully completed.

INS—UNDEFINED GLOBAL AREA area-name FILE filename

Explanation: The specified task is linked to a shareable global area that has not been installed.

Install the required shareable global area, then retry the command.

INS or INV—UNDEFINED PARTITION FILE filename

Explanation: The user specified the /PAR switch either at task build time, or at install time. The specified partition does not exist.

Either rebuild the task, or retry the INS command with the partition name correctly specified.

INS or INV—UNKNOWN KEYWORD IDENTIFIER

Explanation: Self-explanatory.

INS—WATCHDOG TIMER OVERFLOW

Explanation: The value specified in the /TIM switch exceeds 24 hours in hours, minutes, seconds or ticks.

INV—CANNOT INSTALL GLOBAL REGIONS

Explanation: Shareable global areas cannot be installed during System Generation phase 1. Defer installation until phase 2 or later.

INV—CANNOT INSTALL MU TASKS WITH RO AREAS

Explanation: The user is attempting to install a multi-user task that has a read only root, which cannot be done during phase 1 of System Generation. Defer installation of these tasks until phase 2 or later.

INV—CANNOT INSTALL TASKS BOUND TO GLOBALS

Explanation: During phase one of System Generation, tasks that are bound to sharable global areas cannot be installed. Defer installation of these tasks until phase 2 or later.

LUN—CAN'T READ MCR COMMAND BUFFER

Explanation: LUN failed to obtain the command line with which it was invoked. This will occur if LUN is invoked using the RUN command, as opposed to the prescribed method.

Retry the command using the prescribed format. If the failure recurs, notify the Digital software specialist.

LUN—FATAL I/O ERROR

Explanation: The device, on which the specified task is installed, cannot be accessed. One of the following conditions exists:

1 The requested file does not exist in the specified UFD.

2 The UFD does not exist.

3 The file is protected against write access.

4 The volume is not mounted.

5 The volume is off line.

6 The device is not write enabled.

7 The hardware has failed.

LUN—NO LUN ASSIGNMENT

or

LUN—NO LUNS

Explanation: Self-explanatory.

LUN—SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

MEM—GET LINE ERROR

Explanation: MEM failed to obtain the command line with which it was invoked. This situation can occur if MEM is invoked using the RUN command.

Retry the command by typing the following line.

```
MCR>MEM   options
```

If the failure recurs, notify the Digital software specialist.

MEM—ILLEGAL SWITCH

Explanation: An illegal switch or a nonexistent device for TI= was specified, or a series of specifications were included with one or more task names omitted.

Retry the command.

MEM—PRIVILEGED USER FUNCTION ONLY

Explanation: Self-explanatory.

MEM—taskname SPECIFIED VERSIONS NOT LOCKED

Explanation: Either the task specified in the MEM command was not locked in memory or a typing error was made when typing the command.

MEM—SYNTAX ERROR

Explanation: Self-explanatory.

MOU—ACP TASK NOT INSTALLED

Explanation: Self-explanatory. Install the desired ACP task and/or verify that the ACP task name was correctly specified, then retry the command.

MOU—ALL UNITS MUST HAVE SAME NUMBER OF TRACKS

Explanation: All units dedicated to a magnetic tape volume set must be the same; that is, all 800 BPI or all 1600 BPI.

Retry the command with the correct device types specified.

MOU—BAD BLOCK ON DEVICE

Explanation: Either the hardware has failed, or there is an inconsistency with the medium; for example, a seven track tape on a nine track drive.

MOU—COMMAND I/O ERROR

Explanation: The TTY handler is unable to obtain a command line from the user's terminal.

Notify the Digital software specialist.

MOU—DEVICE IS NOT MOUNTABLE

Explanation: Self-explanatory.

MOU—DEVICE NOT READY

Explanation: The requested device is not completely cycled up (in a ready state).

Wait for the device to become ready, then retry the command.

MOU—DEVICE NOT IN SYSTEM

Explanation: The specified device was not included as a usable device when the system was generated. That is, there is no PUD entry for the device. The user has probably typed the device mnemonic incorrectly.

MOU—FILE SYSTEM MESSAGE TASK (F11MSG) MUST BE INSTALLED

Explanation: Self-explanatory; install task F11MSG, then retry the command.

MOU—HANDLER TASK NOT RESIDENT

Explanation: The handler task for the device being mounted is not resident.

Load the required device handler task, then retry the command.

MOU—OTHER VOLUME STILL ON LINE

Explanation: The user is attempting to mount a volume on a device that previously contained a volume that was never dismounted. The user possibly removed the volume from the device and replaced it with a new volume, without having issued a DMO command for the previous volume.

Replace the current volume with the previous one. Issue the appropriate DMO command, then replace the dismounted volume with the next volume to be mounted. Retry the MOU command.

MOU—ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

Explanation: Self-explanatory.

MOU—HOME BLOCK NOT FOUND

Explanation: MOU is unable to obtain the required mount information from the volume's home block. One of the following conditions exist:

1. The volume is not a Files-11 volume.
2. The hardware is malfunctioning. For example, the disk drive's read/write heads are misaligned.

MOU—I/O SYSTEM ERROR

Explanation: An indefinable error occurred during the execution of MOU.

Retry the command. If the error recurs, notify the Digital software specialist.

MOU—NO SPACE FOR VCB

Explanation: There is no space available in the system node pool for the volume control block.

Retry the command. If the failure persists, the system should be rebuilt with a larger node pool.

MOU—PARITY ERROR ON DEVICE

Explanation: This is a hardware failure; notify field service.

MOU—SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

MOU—UNIT DOES NOT SUPPORT 1600 BPI

Explanation: The user specified 1600 BPI for a tape drive that can only support 800 BPI. MOU defaults to 800 BPI and continues processing the command.

MOU—VOLUME IS NOT ANSI FORMAT

Explanation: Either the hardware is malfunctioning or the volume is not an ANSI standard volume. In either case, the volume cannot be mounted.

MOU—WRONG VOLUME

Explanation: The volume's label does not match the label specified in the MOU command line. The volume is not mounted.

Retry the command with the correct volume label specified.

OPE—BYTE ADDRESS

Explanation: The user is attempting to open an odd numbered address location.

OPE—CAN'T READ MCR COMMAND BUFFER

Explanation: OPE failed to obtain the command line with which it was invoked. This will occur if OPE is invoked using the RUN command, as opposed to the prescribed method.

Retry the command using the prescribed format. If the failure recurs, notify the Digital software specialist.

OPE—ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

Explanation: Self-explanatory.

OPE—INPUT I/O ERROR

Explanation: The TTY handler is unable to obtain a command line from the user terminal.

Notify the Digital software specialist.

OPE—INVALID ADDRESS

Explanation: The address specified by the user either is not in the range of valid memory addresses, or was incorrectly specified.

OPE—NUMBER TOO LARGE

Explanation: Self-explanatory.

OPE—OUTPUT ERROR

Explanation: The TTY handler is unable to issue a message to the user's terminal.

Notify the Digital software specialist.

OPE—PARTITION NOT IN SYSTEM

Explanation: Self-explanatory.

OPE—SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

OPE—TASK NOT FIXED IN CORE

or

OPE—TASK NOT IN SYSTEM

Explanation: Self-explanatory.

OPR—ILLEGAL ERROR CODE RECEIVED FROM SPR2..

Explanation: An internal error has occurred. The multiple device output spooler task has delivered an error code that is not within the range of acceptable error codes.

Notify the Digital software specialist.

OPR—ILLEGAL SWITCH COMBINATION

Explanation: The user specified a stop switch that conflicts either with a begin spooling switch, and/or a resume switch, for the same device.

OPR—INVALID DEVICE SPECIFICATION

Explanation: The user has specified a device that is not a system generated device or the device specified is not a carriage controlled unit record device; for example, line printer or teleprinter.

OPR—INVALID OR NONEXISTENT FORM TYPE SPECIFIED

Explanation: The user specified a form type that is not in the range of 0 to 6.

OPR—INVALID SWITCH OR SWITCH VALUE

Explanation: Self-explanatory.

OPR—NO COMMAND LINE SPECIFIED

Explanation: The user typed the OPR command without following it with a command specification.

OPR—NO DEVICE SPECIFIED

or

OPR—NO SWITCH SPECIFIED

Explanation: Self-explanatory.

OPR—SEND/REQUEST TO SPOOLER TASK FAILED

Explanation: The send/request directive to either the queue manager task (SPR) or the multiple device output spooler task (SPR2) has failed. Either or both of the tasks are not installed.

OPR—SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

OPR—UNKNOWN ARGUMENT TO /RS: SWITCH

Explanation: Self-explanatory.

PWD—DIRECTIVE FAILURE

Explanation: An undefined error occurred during the execution of PWD.

Retry the command. If the error recurs, notify the Digital software specialist.

PWD—NONEXISTENT ACCOUNT

Explanation: The user is attempting to assign a password to a nonexistent UFD.

PWD—NON SYSTEM ACCOUNT MAY ONLY MODIFY OWN PASSWORD

Explanation: Self-explanatory.

PWD—READ OR WRITE FAILURE ON DIRECTORY ENTRY

Explanation: One of the following conditions exists:

1 The UFD does not exist.

2 The file is protected against write access.

3 The volume is not mounted.

4 The volume is off line.

5 The device is not write enabled.

6 The hardware has failed.

PWD—SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

QUE—DEVICE dev: DOES NOT CONTAIN A DIRECTORY

Explanation: The user has queued a file to be printed, however the device designated as the input device does not contain a directory. All queue orders up to the device in error have been processed.

Reenter the command from the point of error specifying a correct input device.

QUE—DEVICE dev: IS NOT SPOOLABLE

Explanation: The user has queued a file to be printed onto a device that is not a spoolable device. Spoolable devices are unit-record carriage controlled devices, for example, a line printer or a teleprinter.

QUE—ERROR OPENING filename -nn

Explanation: An error has occurred while opening the file designated by filename. The reason for the failure is determined by -nn, where -nn is a system standard error code. See Appendix D.

All file specifications up to the file in error have been processed. Correct the error and re-issue the command from the point of failure.

QUE—ERROR READING SPRQUEUE.SYS FILE -nn

Explanation: An error has occurred while reading the queue file. The reason for the error is determined by -nn, where -nn is a system standard error message. See Appendix D.

Notify the system manager.

QUE—GET COMMAND LINE ERROR -1

Explanation: An error occurred while reading the indirect command file. The device has cycled down or the file is corrupted.

Obtain a listing of the queue file to determine which commands of the indirect file have been processed. Correct the errant indirect file condition and resume processing from the point of failure.

QUE—GET COMMAND LINE ERROR -2

Explanation: An open failure has occurred on the specified indirect command file. One of the following conditions exists:

1 The user directory does not permit read access to the file.

2 There is a problem with the physical device (for example, device cycled down).

3 The volume is not mounted.

4 The specified file directory does not exist.

5 The file does not exist as specified.

QUE—GET COMMAND LINE ERROR -3

Explanation: The user specified an indirect file in a format that does not conform to syntax rule

QUE—GET COMMAND LINE ERROR -4

Explanation: The maximum depth of three indirect files has been exceeded.

Locate the point within the indirect files where the fourth level of nesting has been specified. Delete this level of nesting. All of the commands up to this point have been processed. Specify th fourth level of nesting as a separate command to QUE. Resume processing of the original indirect command files at the command following the fourth level of nesting.

QUE—NO FILES QUEUED

Explanation: You requested a list of the queue file, but no files are queued.

This message will also occur if the file [1,4]SPRQUEUE.SYS is locked. If the file is locked, notify the system manager who will take the appropriate action.

QUE—SEQUENCE NUMBER NOT SUPPLIED FOR MODIFY OR KILL

Explanation: You entered a modify or kill command omitting the required sequence number.

QUE—SYNTAX ERROR [text string in error]

Explanation: Self-explanatory; refer to the command description for the correct syntax.

If the command was a modify or kill command, obtain a listing of the queued file to determine wha part of the command has been processed. Re-enter the modify or kill command from the point of failure, correcting the syntax error. For all other queue commands no processing has occurred. Re-enter the command correctly.

REA—DEVICE NOT IN SYSTEM

Explanation: The user specified a device mnemonic that has not been generated into the system

REA—FATAL I/O ERROR

Explanation: REA, either while reading or writing the task header, encountered an error. One of the following conditions exists:

1 The requested file does not exist in the specified UFD.

2 The UFD does not exist.

3 The file is protected against write access.

4 The volume is not mounted.

5 The volume is off line.

6 The device is not write enabled.

REA—ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

or

REA—ILLEGAL LUN NUMBER

Explanation: Self-explanatory.

REA—SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

REA—TASK NOT IN SYSTEM

Explanation: Self-explanatory.

RED—CAN'T READ MCR COMMAND BUFFER

Explanation: REA failed to obtain the command line with which it was invoked. This will occur if RED is invoked using the RUN command, as opposed to the prescribed method of invoking RED.

Retry the command using the prescribed format. If the failure recurs, notify the Digital software specialist.

RED—CIRCULAR REDIRECT CHAIN

Explanation: The user, through a sequence of RED commands, has established a chain of redirection which returns to old-dev.

RED—DEVICE NOT KNOWN TO SYSTEM

Explanation: The user specified a device mnemonic that has not been generated into the system.

RED—HANDLER NOT RESIDENT

Explanation: This message informs the user that the device handler for the redirected device is not resident. The redirect has been successfully completed.

RED—ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

Explanation: Self-explanatory.

RED—SOURCE DEVICE, OR ITS EXISTING REDIRECT, IS ATTACHED

Explanation: Self-explanatory; wait until the device becomes unattached, or abort the process that has it attached.

RED—SOURCE DEVICE, OR ITS EXISTING REDIRECT IS ATTACHED

Explanation: Self-explanatory; dismount the device and retry the command.

RED—SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

REM—COMMAND SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

REM—taskname - ERROR READING TASK HEADER

Explanation: Either the task header is corrupt or the task image file has been deleted.

Rectify using REM taskname /-HD.

REM—FAILED TO OPEN @ FILE

Explanation: The requested indirect command file cannot be found on the volume indicated. Either the file does not exist or the file was specified incorrectly.

REM—ILLEGAL ERROR/SEVERITY CODE n1 n2 n3

Explanation: REM failed in its attempt to issue a message to the user terminal. n1 contains the value in R1 at the time of the failure; n2 contains the value in R2; and n3 contains the return address after calling the error message routine.

NOTE: The high order byte in R1 reflects the severity code between 0 and 2.

The low order byte in R1 reflects the error message REM was trying to issue.

Notify the Digital software specialist.

REM—ILLEGAL FUNCTION FOR NON-PRIVILEGED USER

Explanation: Self-explanatory.

REM—ILLEGAL SWITCH

Explanation: The user specified a switch that REM does not recognize. Either the switch is not a REM switch, or it was specified in the wrong format.

REM—INCONSISTENT SWITCH SPECIFIED

Explanation: The user specified a switch combination that is inconsistent.

REM—MAX @ FILE DEPTH EXCEEDED

Explanation: The indirect command file, specified by the user, has an indirect command file specification as one of its commands. REM can process only one indirect file at a time.

Either recreate the originally specified file replacing the offending indirect file specification with the commands it represents, or enter the required commands manually.

REM—TASK taskname FIXED IN MEMORY.

Explanation: A task that is fixed in memory cannot be removed.

Unfix the task, then retry the command.

REM—TASK taskname HAS NONZERO POOL USAGE COUNT

Explanation: The user is attempting to remove a task that has initiated an activity, and that activity is ongoing; for example, initiated an other task, or an I/O operation.

Wait for the initiated activity to complete, then retry the command.

REM—TASK taskname IS ACTIVE

Explanation:

or

REM—TASK taskname NOT IN SYSTEM

or

REM—taskname NOT IN SYSTEM

Explanation: Self-explanatory.

REM—library-name HAS NON-ZERO INSTALL COUNT

Explanation: The user is attempting to remove a library that has installed tasks linked to it.

Remove the installed tasks, then retry the command.

RUN—CAN'T READ MCR COMMAND BUFFER

Explanation: RUN failed to obtain the command line with which it was invoked. This will occur if RUN is invoked using the RUN command, as opposed to the prescribed method.

Retry the command using the prescribed format. If the failure recurs, notify the Digital software specialist.

RUN—HANDLER NOT RESIDENT TO LOAD TASK

Explanation: The device handler task, required to load the requested task from the volume on which it resides, is not in the system.

Load the required handler task, then retry the command.

RUN—INSUFFICENT POOL NODES

Explanation: There are not enough available pool nodes to run the desired task.

Either wait for a node to become available, or notify the system manager. If this has been a persistent problem, the system manager may want to reconfigure the system.

RUN—INVALID KEYWORD

Explanation: The user specified a keyword that either is not a valid keyword, or was specified in an illegal format.

RUN—INVALID TIME PARAMETER

Explanation: The value assigned to the /RSI switch is either invalid, or specified in the wrong format.

RUN—INVALID UIC

Explanation: The UIC specified by the user either does not exist, or was specified in an illegal format.

RUN—INVALID PRIORITY

Explanation: The priority specified in the /PRI switch either exceeds the maximum allowable (250), or was specified in an illegal format.

RUN—MEMORY UNAVAILABLE

or

RUN—PARTITION NOT IN SYSTEM

Explanation: Self-explanatory.

RUN—NON-PRIVILEGED USER MAY NOT RUN AN INSTALLED TASK

Explanation: Self-explanatory.

RUN—SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

RUN—TASK ALREADY ACTIVE

or

RUN—TASK DISABLED

or

RUN—TASK NOT INSTALLED

Explanation: Self-explanatory.

RUN—UNRECOGNIZED DIRECTIVE ERROR

Explanation: An indefinable error occurred during the execution of RUN.

Retry the command. If the error recurs, notify the Digital software specialist.

SAV—CHECKPOINT FILE NUMBER ERROR

Explanation: The information in the file header is incorrect.

SAV—CHECKPOINT FILE SEQUENCE NUMBER ERROR

Explanation: The information in the file header is incorrect.

SAV—CHECKPOINT HEADER READ ERROR

Explanation: Self-explanatory.

SAV—CHECKPOINT HEADER CHECKSUM ERROR

Explanation: Self-explanatory.

SAV—CHECKPOINT LBN ERROR

Explanation: An error has occurred while locating the checkpoint file.

SAV—CHECKPOINT VBN ERROR

Explanation: Virtual block 1 of the file does not exist or there is an error in the file header.

SAV—*DIAG* SYSTEM CLOCK CHANGED TO A KW11 [L or P]

Explanation: The system was generated as having a KW11L or KW11P clock. SAV found that the clock present in the system is not the one specified. SAV continues processing and the system is started with the clock found as the system clock.

SAV—EXTENSION HEADER READ ERROR

Explanation: An error has occurred while reading an index file extension header. This may affect several tasks.

SAV—*FATAL* BAD BOOT IN LOW MEMORY

Explanation: The first 32-word block of memory has been corrupted; probably by hardware. One of the following conditions exists:

1 The bootstrap routine was written by a user who failed to adhere to the rules.

2 The hardware failed.

3 Someone or some running task corrupted physical address 0.

SAV—FATAL ERROR DURING RE-INSTALL - REBOOT SYSTEM

Explanation: A fatal error has occurred while translating from absolute disk addresses to file-ids. The save file on the disk has not been written.

SAV—*FATAL* FAILED TO READ MCR BUFFER

Explanation: SAV failed to obtain the command line with which it was invoked. This will occur if SAV is invoked using the RUN command, as opposed to the prescribed method.

Retry the command using the prescribed format. If the failure persists, notify the Digital software specialist.

SAV—*FATAL* ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

Explanation: Self-explanatory.

SAV—*FATAL* dev: IS STILL LOGGED ON

Explanation: The user attempted to execute SAV with the specified terminal(s) still logged on.

Either reenter the command with the /LOG switch specified, or log the terminals off, and retry the command.

SAV—*FATAL* dev: IS STILL MOUNTED

Explanation: The user attempted to execute SAV with the specified device(s) still mounted.

Either reenter the command with the /MOU switch specified, or dismount the devices and retry the command.

SAV—*FATAL* NO SYSTEM CLOCK RESPONSE, IAS CANNOT RUN

Explanation: Neither the line frequency clock nor the programmable clock will respond to SAV. Either there is no clock on the system, or the clock does not work.

Notify your field service engineer.

SAV—*FATAL* REBOOT SYSTEM

Explanation: A fatal error has occurred while translating from file-ids to absolute disk addresses. The system should be rebooted.

SAV—*FATAL* RO ROOT OF taskname IS LOADED BEYOND END OF SAVE FILE

Explanation: The read-only root of the named multi-user task is either active or fixed at an address beyond that which is to be saved. All active versions of the task must be deactivated to unload the read-only root of the task.

SAV—*FATAL* RO ROOT OF taskname IS LOADING

Explanation: The Executive has initiated the I/O operations to load the read-only root of a multi-user task. This is a separate operation from that used for the impure copies of a multi-user task.

SAV—*FATAL* SHAREABLE GLOBAL sganame IS LOADED BEYOND END OF SAVE FILE

Explanation: The named SGA is either active or fixed at an address beyond that which is to be saved. One of the following steps must be performed:

1 Unload all active or fixed tasks that use the SGA to allow the SGA to be unloaded.

2 Reload the SGA at a lower address.

SAV—*FATAL* SHAREABLE GLOBAL sganame IS LOADING OR EXITING

Explanation: The Executive has initiated the I/O operations to load an SGA or write a read/write SGA to disk. The SGA has one of the following states: load request pending (GS.LRP), load request queued (GS.LRQ), or record request queued (GS.RRQ).

The system is not quiescent and a SAV cannot be performed until these states have disappeared.

SAV—*FATAL* SHAREABLE GLOBAL AREA sganame NOT INSTALLED FROM SYSTEM DISK

Explanation: Self-explanatory.

SAV—*FATAL* SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

SAV—*FATAL* SYSTEM HAS INSUFFICIENT MEMORY PARTITION parname WAS OCCUPIED BEYOND AVAILABLE MEMORY, THE SYSTEM IS NOT RUNNABLE.

Explanation: This situation is caused by an attempt to truncate a system in which some entity (task or SGA) was loaded beyond the available memory.

This error can be caused by bad memory that results in SAVE assuming that such memory does not exist. If so, call your field service engineer. If not, the newly booted system was saved too large to run on this configuration.

SAV—*FATAL* TASK taskname HAS I/O IN PROGRESS

Explanation: The named task has a nonzero I/O in progress count; therefore, the system is not quiescent. Wait for the I/O to complete or abort the task.

SAV—*FATAL* SYSTEM HAS NO 'SY' PUD

Explanation: Self-explanatory.

SAV—*FATAL* TASK taskname IS LOADED BEYOND END OF SAVE FILE

Explanation: The named task is either active or fixed at an address beyond that which is to be saved. One of the following must be done:

1 Unfix the task, if it is fixed.

2 Unload the handler, if the task is a handler, and load it at a lower address.

3 Terminate the active task and perhaps run it at a lower address.

4 Wait for the task to terminate normally.

SAV—*FATAL* TASK taskname HAS RECEIVE DATA QUEUED

Explanation: Self-explanatory.

SAV—*FATAL* TASK taskname IS LOADING OR CHECKPOINTING

Explanation: The Executive has initiated the I/O operations to load or checkpoint the named task. The task has one of the following states: load request pending (TS.LRP), load request queued (TS.LRQ),

The system is not quiescent and a SAV cannot be performed until these states have disappeared.

SAV—*FATAL* TASK taskname NOT INSTALLED FROM SYSTEM DISK

Explanation: Self-explanatory.

SAV—*FATAL* 11/40 TRAP ECO (KT11-D) NECESSARY TO RUN IAS

Explanation: SAV has detected that a PDP-11/40 hardware modification (ECO) necessary for operation of IAS is either not present or is failing to operate correctly.

Notify your field service engineer.

SAV—*FATAL* VOLUME STRUCTURE LEVEL NOT SUPPORTED

Explanation: The disk has been written with a Files-11 structure level which is not supported by IAS. The disk is not usable.

SAV—INDEX FILE HEADER CHECKSUM ERROR

Explanation: Self-explanatory.

SAV—INDEX FILE HEADER READ ERROR

Explanation: Self-explanatory.

SAV—PARTITION parname EXPANDED BY nnnn*32 (Digital) WORD BLOCKS

Explanation: SAV found more memory than was specified at the last save of the system or at System Generation if no previous save occurred. The extra amount of memory is nnnn (Digitalimal) 32-word blocks. The total memory size has been printed in a previous message. The named partition has been expanded to use the extra memory.

SAVE—PARTITION parname TRUNCATED BY nnnn*32 (Digital) WORD BLOCKS

Explanation: SAV found less memory than was specified at the last save or at System Generation if no previous save occurred. The difference in memory size is indicated by nnnn (Digitalimal) 32-word blocks. The total memory size has been printed in a previous message. The named partition has been truncated to fit the memory available.

SAV—SHAREABLE GLOBAL AREA sganame FILE HEADER CHECKSUM ERROR

Explanation: Self-explanatory.

SAV—SHAREABLE GLOBAL AREA sganame FILE HEADER READ ERROR

Explanation: A hardware error has occurred while reading a file header.

SAV—SHAREABLE GLOBAL AREA sganame FILE NUMBER ERROR

Explanation: The information in the file header is incorrect.

SAV—SHAREABLE GLOBAL AREA sganame FILE SEQUENCE NUMBER ERROR

Explanation: The information in the file header is incorrect.

SAV—SHAREABLE GLOBAL AREA sganame LABEL BLOCK CHECK ERROR

Explanation: Self-explanatory.

SAV—SHAREABLE GLOBAL AREA sganame LABEL BLOCK READ ERROR

Explanation: Self-explanatory.

SAV—SHAREABLE GLOBAL AREA sganame REMOVED

Explanation: If an error occurs while an inactive global area is being re-installed, it is automatically removed and this error message is output.

SAV—SHAREABLE GLOBAL AREA sganame VBN ERROR

Explanation: Virtual block 1 of the file does not exist, or there is an error in the file header.

SAV—SHAREABLE GLOBAL AREA sganame LABEL BLOCK READ ERROR

Explanation: Self-explanatory.

SAV—TASK taskname FILE HEADER CHECKSUM ERROR

Explanation: Self-explanatory.

SAV—TASK taskname HAS ILLEGALLY REINSTALLED REGIONS

Explanation: A task with a pure area, or one of the task's regions, was removed and reinstalled without saving the system. SAV has removed the task. Install the task and resave the system.

SAV—TASK taskname FILE HEADER READ ERROR

Explanation: A hardware error has occurred while reading a file header.

SAV—TASK taskname FILE NUMBER ERROR

Explanation: The information in the file header is incorrect.

SAV—TASK taskname FILE SEQUENCE NUMBER ERROR

Explanation: The information in the file header is incorrect.

SAV—TASK taskname HEADER CHECK FAILURE ERROR

Explanation: The task header checkword is incorrect.

SAV—TASK taskname HEADER READ ERROR

Explanation: A hardware error has occurred in reading the task header.

SAV—TASK taskname LABEL BLOCK CHECK ERROR

Explanation: Self-explanatory.

SAV—TASK taskname LABEL BLOCK READ ERROR

Explanation: Self-explanatory.

SAV—TASK taskname REFERENCES UNKNOWN SHAREABLE GLOBAL AREA

Explanation: This occurs if an error has occurred while reinstalling a shareable global area to which the specified task is bound.

SAV—TASK taskname REMOVED

Explanation: If an error occurs while an inactive task is being re-installed, the task is automatically removed and this error message is output.

SAV—TASK taskname TASK HEADER READ ERROR

Explanation: Self-explanatory.

SAV—TASK taskname TASK HEADER WRITE ERROR

Explanation: Self-explanatory.

SAV—TASK taskname VBN ERROR

Explanation: Virtual block 1 of the file does not exist, or there is an error in the file header.

SAV—THE FOLLOWING EMPTY PARTITION(S) TRUNCATED TO ZERO SIZE: parname1 parname2

Explanation: The system has been successfully reduced in size but all of the named partitions are completely beyond the available memory. Any attempt to run a task already installed in one of these partitions causes the task to be queued on that partition's memory required list (MRL). The task remains on the MRL until it is aborted. Any attempt to install a task in such a partition results in an INS error message to the effect that the task is larger than the partition.

SAV—VOLUME HOME BLOCK READ ERROR

Explanation: The home block for the volume could not be found.

SAV—*WARNING* ACTIVE SGA sganame NOT CORRECTLY RE-INSTALLED

Explanation: If an error occurs while reinstalling an active SGA, this error message is output and the SGA is not removed. However any subsequent attempt to load it will fail.

SAV—*WARNING* ACTIVE TASK taskname NOT CORRECTLY RE-INSTALLED

Explanation: If an error occurs while re-installing an active task this error message is output and the task is not removed. However any subsequent attempt to load it will fail.

SAV—*WARNING* COMMON AREA sganame CONVERTED TO LIBRARY

Explanation: If an error occurs while re-installing an active common area SGA, it is automatically converted to a library SGA so that it is not written back to the disk.

SAV—*WARNING* FIXED TASK taskname NOT CORRECTLY RE-INSTALLED

Explanation: If an error occurs while re-installing a fixed task, this error message is output and the fixed task is not removed. However, any subsequent attempt to load it will fail.

SET—CAN'T READ MCR COMMAND BUFFER

Explanation: SET failed to obtain the command line with which it was invoked. This will occur if SET is invoked using the RUN command, as opposed to the prescribed method.

Retry the command using the prescribed format. If the failure recurs, notify the Digital software specialist.

SET—ILLEGAL CACHE VALUE

Explanation: A number that was not in the range 1 through 32767 (Digitalimal) was specified in the /CAC=n switch.

Retry the command using a number within the specified range.

SET—ILLEGAL FUNCTION FOR MACHINE TYPE

Explanation: The /CAC switch was specified while running on a computer other than a PDP-11/44 or a PDP-11/70.

SET—ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

or

SET—PARAMETER ERROR - BAD DEVICE NAME

or

SET—SPECIFIED DEVICE IS NOT A TERMINAL TYPE DEVICE

or

SET—SPOOLING NOT SUPPORTED IN THIS RELEASE FOR THIS DEVICE

Explanation: Self-explanatory.

SET—SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

SPR—DEVICE dev: NOT FOUND IN PUD TABLE

Explanation: A file has been queued via a PRINT$ macro to a nonexistent system device. The queue entry is automatically dequeued.

SPR—ERROR ISSUING SEND REQUEST TO SPR2.. -nn

Explanation: An error occurred when SPR issued a Send/Request directive to the output spooler task (SPR2..). The reason for the error is determined by -nn, where -nn is the directive status word (DSW) error code.

SPR—ERROR OR RECEIVE -nn

Explanation: An error occurred while SPR was receiving a Send/Request directive. The reason for the error is determined by nn, where -nn is the directive status word (DSW) error code. The error codes are described in the *IAS I/O Operations Reference Manual*.

SPR—ERROR READING SPRQUEUE.SYS FILE -nn

or

ERROR WRITING SPRQUEUE.SYS FILE -nn

or

ERROR OPENING SPRQUEUE.SYS FILE -nn

or

ERROR CLOSING SPRQUEUE.SYS FILE -nn

Explanation: An error occurred when SPR attempted to access the queue file. The reason for the error is determined by -nn, where -nn is the directive status word (DSW) error code. The error codes are described in the *IAS I/O Operations Reference Maunual.*

NOTE: If the error is persistent, the queue file is corrupted. Perform the queue error procedure documented in the *IAS System Management Guide.*

SPR—INVALID SEQUENCE NUMBER

Explanation: The sequence number for a kill or modify command does not correspond to any sequence number in the queue.

Retry the command with the correct sequence number specified.

SPR—SPRQUEUE.SYS FILE CAPACITY EXCEEDED

Explanation: The SPRQUEUE.SYS file has a capacity of 2047 queue entries. This capacity has been exceeded, and the last entry has not been queued.

Avoid adding more entries to be spooled. Allow the output spooler task to process sufficient entries to create space in the queue. Then retry the command.

SPR2.. —DEVICE dev:, FAILED TO OPEN FILE

Explanation: The output spooler was unable to open an output file on device (dev:). The open request is automatically aborted by the output spooler. Either the file to be opened was deleted after a request to print it was queued, or the volume on which the file resides was dismounted.

SPR2.. —DEVICE dev: TIMED OUT

Explanation: Three minutes have elapsed since a QIO output request was issued to device (dev:). The multiple device output spooler has received no notification from the handler that the request has completed. This error normally occurs when the line printer is not in a ready state, and will be accompanied by a not ready message issued by the device handler. This message requires no user action.

SPR2.. —GET$ ERROR - DEVICE dev: ERROR CODE -nn

Explanation: An input error occurred on the device that contains the file being transferred to device dev:. The reason for the error is determined by -nn, where -nn is a system standard error code. See Appendix D. This message does not require user action.

SPR2.. —GETS$ ERROR - DEVICE dev: - ERROR CODE 40

Explanation: The output spooler encountered an input record size, on device dev:, that is greater than 256 bytes. Printing of the input file is automatically aborted.

SPR2.. —INTERNAL ERROR - BAD SEND/REQUEST FROM ...OPR

Explanation: OPR, as a result of an OPR command issued by the user, issued a send/request directive to the output spooler. The request code does not match the appropriate send/request assigned codes.

Document the occurrence along with the OPR command issued, and notify the Digital software specialist.

SPR2.. —INTERNAL ERROR—DUPLICATE DEVICES ON DEVICE dev:

Explanation: The queue manager task directed the output spooler task to transfer data to a device (dev:) that is currently being used.

Perform the corrective action detailed in the *IAS System Management Guide*.

SPR2.. —INTERNAL ERROR—FAILED TO ASSIGN LUN

Explanation: The output spooler task was unable to assign the LUN for a given device. This message will be issued as a result of a system generation in which the required device was deleted.

Perform the corrective action detailed in the *IAS System Management Guide*.

SPR2.. —INTERNAL ERROR—TOO MANY SIMULTANEOUS DEVICES

Explanation: The output spooler task notified the queue manager task that it is capable of processing n devices simultaneously. The queue manager task, however, delivered n+1 process requests.

Perform the corrective action detailed in the *IAS System Management Guide*.

SPR2.. —INTERNAL ERROR - ZERO COPIES SPECIFIED

Explanation: The queue manager task requested an output spooler operation on a file, but it also requested that no copies are to be printed. The queue file might be corrupt.

Perform the corrective action detailed in the *IAS System Management Guide*.

SPR2.. —I/O ERROR ON DEVICE dev: - ERROR CODE -nn

Explanation: An I/O error occurred during an output operation on device (dev:). The reason for the error is determined by -nn, where -nn is a system standard error code. See Appendix D. This message does not require user action.

SPR2.. —QIOW$ FAILURE ON DEVICE dev: DSW=nn - DEVICE IN WAIT

Explanation: A QIO directive error occurred on device dev:. The reason for the error is determined by -nn, where -nn is the directive status word (DSW) error code. The error codes are described in the *IAS I/O Operations Reference Manual*.

Correct the error condition, then issue the appropriate OPR command:

OPR dev:/RS to continue

OPR dev:/AB to abort the operation

SPR2.. - SENT/REQUEST TO SPR... FAILED

Explanation: Task SPR... is not installed; install the task.

SWA—ALL BAD-BLOCK BLOCKS ARE UNREADABLE

Explanation: None of the blocks on the volume reserved for bad block information can be read. This probably indicates a serious hardware problem.

SWA—WARNING—BAD BLOCK DATA IS INCORRECT, IGNORED

Explanation: The format of the bad block information on the volume is incorrect.

Run the BAD command on the volume (see Chapter 7).

SWA—CAN'T ALLOCATE - SWAP FILE number IS ACTIVE

Explanation: When a swap file is created with a file number specified, SWA checks to see if any higher numbered swap files have space allocated in them. If so, the file cannot be created because internal swap file control data cannot be made correct.

Create a swap file with a number higher than that specified in the error message.

SWA—COMMAND FILE OPEN FAILURE

Explanation: The specified indirect command file either does not exist, or cannot be opened for another reason.

Check the indirect command file.

SWA—COMMAND SYNTAX ERROR

Explanation: Self-explanatory.

Reenter the command with the correct syntax.

SWA—DEDICATED SWAP VOLUME volumename MUST NOT BE MOUNTED

Explanation: A dedicated swap file can only be created on an unmounted volume, since any file structure existing on the volume will be destroyed.

Dismount the volume and reenter the command.

SWA—DEVICE devicename IS NOT A DIRECTORY DEVICE

Explanation: A swap file can only be created on a directory-structured device, normally a disk.

Reenter the command specifying a different device.

SWA—DEVICE devicename IS NOT MOUNTED

Explanation: When a swap file is created (except on a dedicated volume) the volume must be mounted.

Mount the volume and reenter the command.

SWA—DEVICE NOT ALLOWED FOR THIS OPERATON

Explanation: A device specification can only occur when a swap file is being created.

Reenter the command without a device specification.

SWA—WARNING—ERROR DELETING FILE filespec - ERR=err-code

Explanation: When a swap file is removed from the system, the corresponding file must be deleted using the file system. This error message indicates that the file could not be deleted, although it will no longer be used for swapping.

The err-code is a standard file system error code described in the *IAS I/O Operations Reference Manual.*

SWA—ERROR READING COMMAND LINE

Explanation: This error message normally indicates that a command line is too long, or that an indirect command file cannot be read.

Reenter the command, or check the indirect command file.

SWA—FAILED TO ALLOCATE SFL NODE

Explanation: The SCOM node required to control swap space allocation cannot be obtained, indicating that there is insufficient space available in the node pool.

Reenter the command when there is less system activity.

SWA—FAILED TO ALLOCATE SWAP BITMAP

Explanation: The swap file bitmap, which is used to control swap file allocation, is held in SCOM. This message indicates that there is insufficient space available in the node pool to accommodate it.

Create a smaller swap file, which will use less space for the bitmap.

SWA—FAILED TO CREATE SWAP FILE ON devicename -ERR=err-code

Explanation: An error (other than insufficient space) has occurred when creating a swap file. The err-code is a standard file system error code described in the *IAS I/O Operations Reference Manual.*

SWA—FAILED TO PICK NEW BITMAP

Explanation: The new, smaller bitmap cannot be created when a swap file is being deleted.

Reenter the command when there is less system activity.

SWA—FATAL DISK ERROR READING BAD BLOCK INFORMATION

Explanation: An irrecoverable disk error has occurred when reading bad block information from a dedicated swap volume. This indicates a serious hardware problem.

SWA—FILE ACTIVE - MARKED FOR DELETE

Explanation: This error may occur when a file is being deleted. It indicates that either this file or one with a higher number has space allocated from it. The file will be deleted automatically when the space is freed.

SWA—FILE LENGTH MUST BE SPECIFIED (/LE:n)

Explanation: Self-explanatory. See the SWA command in Chapter 7.

SWA—ILLEGAL DEVICE devicename -ERR=err-code

Explanation: The device name specified is incorrect; for example, it may not exist in the system. The err-code is a standard file system error code described in the *IAS I/O Operations Reference Manual*.

Reenter the command with the correct device name.

SWA—ILLEGAL DEVICE devicename FOR DEDICATED SWAP VOLUME

Explanation: A dedicated swap volume must be a non-removable type disk, for example, an RS04.

Do not use the dedicated volume facility with any other type of disk.

SWA—ILLEGAL OR INCONSISTENT SWITCH

Explanation: Self-explanatory.

Reenter the command with the correct switches specified.

SWA—MAXIMUM TOTAL SWAP SIZE EXCEEDED, MAX=n

Explanation: The total swap space limit (four million words) has been exceeded.

SWA—NO ROOM ON DEVICE devicename FOR SWAP FILE

Explanation: Self-explanatory.

SWA—NOT ENOUGH CONTIGUOUS SPACE FOR SWAP FILE ON devicename

Explanation: Self-explanatory.

Obtain the required space using several smaller files.

SWA—REALTIME-ONLY SWAP FILES MUST COME FIRST

Explanation: The number of the swap file must be less than that for any other swap file that does not have the real-time attribute.

SWA—SWAP DEVICE MUST BE SPECIFIED

Explanation: Swap device has not been specified.

Reenter the command with the device specified.

SWA—WARNING—SWAP FILE CONTAINS BAD BLOCK(S)

Explanation: The part of the dedicated swap volume used for this swap file contains some bad blocks. Therefore not all of the area may be used for swapping.

This is only a warning message.

SWA—SWAP FILE n DELETED

Explanation: This message is for information only. It indicates that a swap file that was marked for delete has actually been deleted.

SWA—SWAP FILE n DOES NOT EXIST

Explanation: Self-explanatory.

SWA—TOO MANY SWAP FILES, MAX=n

Explanation: The maximum number of swap files (64) has been exceeded.

SWA—T-S SWAP FILES MUST COME AFTER R-T SWAP FILES

Explanation: The number of a swap file available for timesharing (T-S) tasks must be greater than for any swap files reserved for real-time (R-T) tasks.

SYS—ILLEGAL SWITCH

Explanation: The user specified a switch that SYS does not recognize. Either the switch is not a SYS switch, or it was specified in the wrong format.

SYS—SYNTAX ERROR

Explanation: The user entered a SYS command in a format that does not conform to syntax rules.

TIM—CAN'T READ MCR COMMAND BUFFER

Explanation: TIM failed to obtain the command line with which it was invoked. This will occur if TIM is invoked using the RUN command, as opposed to the prescribed method.

Retry the command using the prescribed format. If the failue recurs, notify the Digital software specialist.

TIM—ILLEGAL FUNCTION FOR NON-PRIVILEGED USER

Explanation: A nonprivileged user is not allowed to modify the time or date.

TIM—SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

TER—AMBIGUOUS KEYWORD

Explanation: For example, /P could be /PRINTER or /PARITY.

Reissue complete command with sufficient number of characters for uniqueness.

TER—CHARACTERISTIC NOT IN THIS SYSTEM

Explanation: Handler has been built without support for this characteristic, for example, /ESCAPESEQ when E$$SEQ in PARAMS.MAC has been set to zero.

TER—DIALUP LINE NOT CONNECTED

Explanation: Self-explanatory.

TER—ERROR READING COMMAND LINE

Explanation: @ file does not exist or has bad syntax, or other Get Command Line error.

TER—FUNCTION NOT VALID FOR THIS TERMINAL

Explanation: Handler does not have set characteristics function, or /HANGUP has been issued for non-dialup line.

TER—ILLEGAL LINE SPEED FOR INTERFACE

Explanation: For example, /SPEED:7200 for DH11.

TER—ILLEGAL LINE (SPEED OR OTHER) PARAMETERS

Explanation: Self-explanatory.

TER—ILLEGAL VALUE

Explanation: Illegal value for option, for example, /FILL:364

TER—INTERFACE DOES NOT HAVE VARIABLE PARAMETERS

Explanation: For example, /SPEED is fixed for a DL11.

TER—INVALID CHARACTERISTIC NAME

Explanation: Self-explanatory.

TER—INVALID LINE SPEED

Explanation: Illegal argument to /SPEED.

TER—INVALID PARITY TYPE

Explanation: Illegal argument to /PARITY.

TER—INVALID READ-AHEAD TYPE

Explanation: Illegal argument to /READAHEAD.

TER—INVALID TERMINAL NUMBER

Explanation: A terminal name includes a unit number greater than 77 (octal).

TER—INVALID TERMINAL TYPE

Explanation: Illegal argument to /TERMINAL, for example, /TERMINAL:LA30X.

TER—MORE THAN ONE TERMINAL TYPE SPECIFIED

Explanation: Self-explanatory, for example, /LA30S/VT05.

TER—NO DYNAMIC SPACE IN HANDLER - TELL SYSTEM MANAGER

Explanation: No space in handler to save the default characteristics. Probably the handler has been built with too small a node pool.

TER—'NO' PREFIX NOT ALLOWED FOR THIS CHARACTERISTIC

Explanation: Self-explanatory.

TER—NO TERMINAL TYPE OR CHARACTERISTIC SPECIFIED

Explanation: Self-explanatory.

TER—NUMBER REQUIRED

Explanation: Number required as argument to option.

TER—ONLY [1,1] USERS ALLOWED TO CHANGE DEFAULT

Explanation: Only [1,1] users may issue /DEFAULT

TER—ONLY [1,1] USERS ALLOWED TO CHANGE EXPLICIT TERMINAL

Explanation: Refers to the format MCR>TER TTn:/.

TER—PRIVILEGE VIOLATION

Explanation: The user is not privileged to perform the function involved.

TER—SPLIT SPEED NOT AVAILABLE ON THIS INTERFACE

Explanation: For example, DZ11 must receive and send at same speed.

TER—SYNTAX ERROR

Explanation: Syntax error in command, that is, illegal character or invalid terminal name.

TER—TERMINAL NOT IN SYSTEM

Explanation: Self-explanatory.

TER—TERMINAL NOT IN THIS CONFIGURATION

Explanation: Self-explanatory.

TER—TOO MANY CHARACTERISTICS IN SINGLE COMMAND

Explanation: Self-explanatory.

TER—UNRECOGNIZED HANDLER ERROR CODE - FATAL SOFTWARE ERROR

Explanation: Unrecognized error return code from handler.

UFD—DEVICE NOT IN SYSTEM

Explanation: You specified a device mnemonic that has not been generated into the system.

UFD—DIRECTORY ALREADY EXISTS

Explanation: Self-explanatory.

UFD—FAILED TO CREATE DIRECTORY

Explanation: The device is write locked; write enable it and retry the command.

UFD—FAILED TO ENTER IN MFD

Explanation: The UFD command task is unable to establish a pointer in the master file directory to the newly created UFD.

Either the hardware has failed, or the master file directory is corrupted.

UFD—ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

or

UFD—NOT FILES-11 DEVICE

Explanation: Self-explanatory.

UFD—PRIVILEGE VIOLATION ON MFD

Explanation: The UFD command task, by virtue of the UIC under which it is running, is not allowed access to the volume MFD.

UFD—SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

UFD—VOLUME NOT MOUNTED

Explanation: Self-explanatory.

UFD—WRITE ATTRIBUTES FAILED

Explanation: The hardware has malfunctioned; notify field service.

UNL—CAN'T READ MCR COMMAND BUFFER

Explanation: UNL failed to obtain the command line with which it was invoked. This will occur if UNL is invoked using the RUN command, as opposed to the prescribed method.

Retry the command using the prescribed format. If the failure reoccurs, notify the Digital software specialist.

UNL—HANDLER NOT RESIDENT

or

UNL—ILLEGAL FUNCTION FOR NON-PRIVILEGED TERMINAL

Explanation: Self-explanatory.

UNL—NODE FOR EXIT UNAVAILABLE

Explanation: A required pool node is unavailable; the system is saturated with active tasks.

Either wait for a node to become available or notify the system manager. The system manager may want to reconfigure the system.

UNL—TASK NOT IN SYSTEM

Explanation: Self-explanatory.

UTL—SYNTAX ERROR

Explanation: Self-explanatory; refer to the command description for the correct syntax.

UTL—CSI SYNTAX ERROR

Explanation: Self-explanatory.

Reenter the command with the correct syntax.

UTL—GET COMMAND LINE ERROR

Explanation: One of the following conditions exists:

1 An indirect command file does not exist.

2 An indirect command file cannot be opened for another reason.

3 The command line is too long.

Reenter the command line or check that the indirect command file can be read.

UTL—ILLEGAL NUMBER OF LEVELS

Explanation: The number of levels specified with the /LV switch is too large or is zero.

Reenter the command with fewer levels.

UTL—ILLEGAL OPERATION FROM NON-PRIVILEGED TERMINAL

Explanation: Self-explanatory.

UTL—ILLEGAL SWITCH OR SWITCH VALUE

Explanation: Self-explanatory.

Reenter the command line with the correct switch or switch value specified.

UTL—ILLEGAL TIME SLICE VALUE

Explanation: The value specified with the /TS switch is too large, or is zero.

UTL—INVALID LEVEL NUMBER

Explanation: The level specified with the /TF switch does not exist; that is, it is greater than th number of levels.

Reenter the command line with a valid level number.

UTL—WARNING—LESS THAN 50K OF SWAP SPACE AVAILABLE

Explanation: Self-explanatory. This message may occur when timesharing is started. If there is too little swap space, timesharing tasks cannot be run correctly.

UTL—NODE PICK ERROR

Explanation: There are too few nodes in the system pool to create the specified number of levels or a promotion table of the specified size. (For further information on the promotion table, see th *IAS System Management Guide*.)

Reenter the command specifying fewer levels or a smaller promotion table.

UTL—NO SWAP SPACE

Explanation: Self-explanatory.

Create some swap space and Reenter the command.

UTL—OPERATION ILLEGAL—TIMESHARING ACTIVE

Explanation: Certain parameters cannot be altered while the scheduler is enabled.

Disable timesharing then Reenter the command.

UTL—SPECIFIED PARTITION IS NOT 'T'-TYPE

Explanation: The partition specified with the /EN switch (or GEN if no partition was specified) i not a timesharing partition.

Reenter the command specifying the correct partition, or perform a SYSGEN to create a timesharing partition (see the *IAS Installation and System Generation Guide*).

UTL—SPECIFIED PARTITION NOT IN SYSTEM

Explanation: Self-explanatory.

UTL—TIMESHARING ALREADY ACTIVE

Explanation: Self-explanatory.

UTL—TIMESHARING NOT ACTIVE

Explanation: Self-explanatory.

UTL—TIMESHARING TASKS ACTIVE

Explanation: Timesharing cannot be disabled while there are tasks running under the control of the scheduler.

Ensure that all timesharing tasks have terminated, then Reenter the command.

B.2 Messages from Task MCRERR

Nine of the MCR functions are performed by a single multi-function task named MFT:

ABO (Abort task)
ALT (Alter priority)
CAN (Cancel request)
FIX (Fix task in memory)
UNF (Unfix task)
DIS (Disable task)
ENA (Enable task)
LOA (Load handler task)
RES (Resume task execution)

MCRERR is the error handling task for MFT. The advantage of MCRERR is that it needs to be resident only when error handling is required. The messages issued by MCRERR are described below.

Each message consists of two lines. The first line printed is the command that caused the error. The second line is the actual error message.

Message descriptions are followed in the manual by a message number. If MCRERR is not installed, this number is printed in the following format on the terminal instead of the message.

```
xxx -- ERRnn
```

xxx = the function called, for example, LOA.
nn = the message number.

CAN'T READ MCR COMMAND BUFFER 9

Explanation: The requested function failed to obtain the command line with which it was invoked. This occurs when the function is requested using the RUN command as opposed to the prescribed method.

Retry the command using the prescribed format. If the failure recurs, notify the Digital software specialist.

DEVICE NOT IN SYSTEM 3

Explanation: The specified device was not defined during System Generation. The user probably typed the device mnemonic incorrectly.

HANDLER NOT RESIDENT 17

Explanation: The user specified a device for which the required handler is not resident.

Load the required device handler task and retry the command.

ILLEGAL FUNCTION OR SWITCH FOR NON-PRIVILEGED TERMINAL 2

Explanation: Self-explanatory.

ILLEGAL SWITCH OR SWITCH COMBINATION FOR THIS FUNCTION 24

Explanation: Self-explanatory; refer to the description of the command.

INSUFFICIENT POOL NODES TO PERFORM RQST 25

Explanation: Required pool nodes are unavailable; the system is saturated with active tasks.

Either wait for nodes to become available, or notify the system manager. The system manager may want to reconfigure the system.

INVALID KEYWORD SPECIFIED 32

Explanation: Self-explanatory.

INVALID PRIORITY 29

Explanation: Self-explanatory.

INVALID TIME PARAMETER SPECIFIED 31

Explanation: Self-explanatory.

INVALID UIC SPECIFIED 27

Explanation: Self-explanatory.

MEMORY UNAVAILABLE 16

Explanation: Self explanatory. Remove unnecessary tasks, or wait for memory to become available and then retry the command.

NODE FOR FIXING UNAVAILABLE 20

Explanation: A required pool node is unavailable; the system is saturated with active tasks.

Either wait for a node to become available, or notify the system manager. The system manager might want to reconfigure the system.

NOT ALLOWED IN TIME-SCHEDULED PARTITION 33

Explanation: Self-explanatory.

NUMERIC CONVERSION ERROR 30

Explanation: Self-explanatory.

PARTITION TOO SMALL FOR TASK 26

Explanation: Self-explanatory; install the task in a larger partition or reduce the size of the task.

PLEASE LOGON WITH "[UIC]" 8

Explanation: You have not logged into the system using the HEL command.

Log in and retry the command.

SPECIFIED PARTITION DOES NOT EXIST 28

Explanation: Self-explanatory; if the error was not caused by mistyping the partition name, notify the system manager.

SYNTAX ERROR 1

Explanation: Self-explanatory.

TASK ALREADY FIXED 21

Explanation: Self-explanatory.

TASK ALREADY (OR STILL) ACTIVE 18

Explanation: Self-explanatory.

TASK CAN'T BE DISABLED 12

Explanation: The task is not enabled.

TASK CHECKPOINTABLE 15

Explanation: The task is checkpointable and, therefore, cannot be fixed in memory.

TASK DISABLED 14

Explanation: The specified task must be enabled before it can be fixed.

TASK EXITING 10

Explanation: The requested function cannot be performed because the specified task is in the process of exiting from the system.

TASK LOADING OR EXITING 5

Explanation: The requested function cannot be performed because the specified task is either being loaded or is exiting.

TASK NOT ABORTABLE 6

Explanation: An attempt was made to abort a task that was task built as not abortable using the /-AB switch. This task cannot be aborted.

TASK NOT ACTIVE 7

Explanation: The requested function cannot be performed because the specified task is not active.

TASK NOT DISABLED 13

Explanation: Self-explanatory.

TASK NOT ENABLED 11

Explanation: Self-explanatory.

TASK NOT FIXABLE 19

Explanation: The specified task was not fixable at task build time. The task must be built with the /FX switch specified.

TASK NOT FIXED 23

Explanation: Self-explanatory.

TASK NOT IN SYSTEM 4

Explanation: Self-explanatory.

TASK NOT SUSPENDED 22

Explanation: Self-explanatory.

UNKNOWN DIRECTIVE ERROR 0

Explanation: An unidentifiable error occurred during execution of the requested MFT function.

Retry the command. If the error recurs, notify the Digital software specialist.

C TKB Command Summary and Error Messages

C.1 TKB (Task Builder) Command Summary

The TKB command string format is as follows:

```
task-image[/switch(es)],mem-allocation,[/switch(es)]
symbol-definition[/switch(es)]=input/switch(es)
```

where:

Parameter	Definition
task-image	The relocatable executable task file (.TSK file) in the format: dev:[UIC]filename.typ;ver
mem-allocation	Memory allocation file containing information about the size and location of components within the task (.MAP)
symbol-definition	STB file that contains all the global symbols defined in this task. This file has the format: dev:[UIC]filename.typ;ver
input	.OBJ file to be task built in the format: dev:[UIC]filename.type;ver
/switch(es)	One or more of the TKB control switches.

C.1.1 TKB File Switches

Table C–1 Task Builder Switches

Switch	Function
	Task Image File Switches
/AB	The task is abortable. /AB is assumed.
/CP	The task is checkpointable. /-CP (not checkpointable) is assumed.
/DA	A debugging aid is to be loaded with the task. This switch causes ODT.OBJ, which resides in the system area (LB:[1,1]), to be automatically included in the task image. /-DA is assumed.
/DS	The task can be disabled. /DS is assumed.
/FP	The task uses floating point. /FP is assumed.
/FU	Search all co-tree overlay segments for matching definition or reference when processing modules from the default object module library. /-FU is assumed.
/FX	The task can be fixed in memory. /FX is assumed.
/HD	The task image includes a header. /HD is assumed.
/MU	The task is multi-user. /-MU is assumed.
/OR	Include overlay run-time system. /OR is assumed.

Table C–1 (Cont.) Task Builder Switches

Switch	Function
	Task Image File Switches
/PI	The task contains only position-independent code. /-PI is assumed.
/PR	The task is privileged. /-PR is assumed.
/RO	Memory resident overlay operator (!) is enabled so that task can be built with memory resident overlays. /-RO is assumed.
/RW	Task has read-write access to read-only code. /-RW is assumed.
/SE	Task can receive data.
/SQ	The segments in the task image file are constructed of P-sections in the order in which they were encountered during input file processing. The default, /-SQ, causes the segments to be constructed alphabetically by P-section name.
/SR	The task accepts send and request/resume data from nonprivileged users. /-SR is assumed.
/TR	The processor T bit is to be set in the initial PS wprd of the task. /-TR is assumed.
/WN	System waits a certain period of time for nodes to become available. /WN is assumed.
/XT:n	Exit after any n diagnostics are produced. /-XT is assumed.
	If no switches are specified, the resultant task image file has the following default attributes: /AB,/CP,/-DA,/DS,/FP,/-FX,/HD,/-MU,/-PI, /-PR,/-SQ,/-SR,/-TA,/-TR,/-XT
	Memory Allocation Map File Switches
/CR	Produce a cross reference listing. /-CR is assumed.
/MA	Include all modules in the memory allocation file. /MA is assumed.
/-MA	Exclude all modules in this input file from the memory allocation file. /MA is assumed.
/SH	Produce a short map. /-SH is assumed.
/SP	Spool the memory allocation file. /SP is assumed.
/UR	Print undefined references on initiating terminal. /UR is assumed.
/WI	Produce a wide memory allocation file. /WI is assumed.
	Symbol Definition File Switches
/PI	The task image contains only position-independent code. Therefore, the symbol definition file is writte as a relocatable file, that is, the symbol definitions may be relocated. /-PI is assumed.
/UN	Include references to undefined symbols in symbol table file. /UN is assumed.

Table C–1 (Cont.) Task Builder Switches

Switch	Function
	Input Object File Switches
/CC	The specified file contains concatenated object files. Note that when /CC is used, it results in an additional pass of the object file and, therefore, should be used judiciously.
/DA	The specified file contains a debugging aid (for example, ODT). This switch must be included to specify a debugging aid with a transfer address as an input file. It is included so that experimental or private copies of debugging aids other than ODT may be included in the task image.
/DL	Specified library file is a replacement for the default system object module library. /-DL is assumed.
/LB	The specified file contains a user relocatable library (as built by the Librarian). This library will be searched to resolve undefined global symbols.
/MP	The specified file contains an overlay description of the Task to be built. If this switch is present, it must appear on the first input file specification.
/SS	Perform selective symbol search. /-SS is assumed.

For example:

```
         TKB>FRED,FRED=

         TKB>FRED/MP

         ENTER OPTIONS:

         TKB>

         .

         . OPTIONS

         .

         TKB>/

     OPTIONS:

     ABSPAT = SGNAM:PADDR:VALUE:...:VALUE

     ACTFIL = NUM

     ASG = devnam:unit:...:unit

ASG default assignments:
Symbolic    LUN

     SY0    1
     SY0    2
     SY0    3
     SY0    4
     TI     5
     CL     6

     ARTG = NUMBER OF ATTACHMENT DESCRIPTOR BLOCKS IN TASK HEADER

     BASE = BOUND

     EXTSCT = CNAME:LENGTH

     EXTTSK = TASK EXTENSION

     FMTBUF = BUFFER EXTENSION

     GBLDEF = SNAME:VALUE
```

```
GBLPAT = SGNAM:SNAME+OFFSET:VALUE:VALUE:...:VALUE

GBLREF = SNAME

MAXBUF = MAXIMUM RECORD BUFFER SIZE

MAXEXT = MAXIMUM EXTENSION SIZE

ODTV = SNAME:LENGTH

PAR = PNAME

POOL = PNUM

PRI = PNUM

RESAPR = APR1[:APR2...]

RESSGA = FNAME/ACCESS[:APR]

SGA =  LNAME:ACCESS;APR

STACK = SNUM

SYMPAT = SGNAME:SNAME[+/-OFFSET]:VALUE:VALUE:...:VALUE

TASK = TASKNAME

TOP =  BOUND

TSKV = ADDRESS OF TASK SST VECTOR

UIC = [PROJ,PROG]

UNITS = NUNIT

VSECT = P-SECTNAME:BASE:WINDOW:[PHYSICAL - LENGTH]
```

C.2 TKB Error Messages

The Task Builder produces diagnostic and fatal error messages. Error messages are printed in the following forms:

```
TKB -- *DIAG*-error-message
```

or

```
TKB -- *FATAL*-error-message
```

Some errors depend on correction from the terminal. If the user is entering text at the terminal, a diagnostic error message can be printed, the error corrected, and the task building sequence continued. If the same error is detected by the Task Builder in an indirect file, the Task Builder cannot request correction and thus the error is termed fatal and the task build is aborted.

Some diagnostic error messages are simply informative and advise the user of an unusual condition. If you consider the condition normal to the task, you can install and run the task image. This appendix lists the error messages produced by the Task Builder. Most of the error messages are self-explanatory. The Task Builder prints the text shown in this manual in upper case letters. In some cases, the Task Builder prints the line in which the error occurred, so that the user can examine the line which caused the problem and correct it.

0. ILLEGAL GET COMMAND LINE ERROR CODE

   Explanation: System error. (No recovery.)

1. COMMAND SYNTAX ERROR

   Explanation: invalid-line

   The invalid-line printed has incorrect syntax.

2. REQUIRED INPUT FILE MISSING

   Explanation: At least one file is required for a task build.

3. ILLEGAL SWITCH

   Explanation: invalid-line

   The invalid line printed contains an illegal switch or switch value.

4. NO DYNAMIC STORAGE AVAILABLE

   Explanation: The Task Builder needs additional symbol table storage and cannot obtain it. The input has exceeded the Task Builder's capability.

5. ILLEGAL ERROR/SEVERITY CODE

   Explanation: System error. (No recovery.)

6. COMMAND I/O ERROR

   Explanation: I/O error on command input device. (Device may not be on-line or possible hardware error.)

7. INDIRECT FILE OPEN FAILURE

   Explanation: invalid-line

   The invalid-line contains a reference to a command input file which could not be located.

8. INDIRECT COMMAND SYNTAX ERROR

   Explanation: invalid-line

   The invalid-line printed contains a syntactically incorrect indirect file specification.

9. INDIRECT FILE DEPTH EXCEEDED

   Explanation: invalid-line

   The invalid-line printed gives the file reference that exceeded the permissible indirect file depth (3).

10. I/O ERROR ON INPUT FILE file-name

    Explanation:

11. OPEN FAILURE ON FILE file-name

    Explanation:

12. SEARCH STACK OVERFLOW ON SEGMENT segment-name

    Explanation: The segment segment-name is more than 16 branch segments from the root segment.

13. PASS CONTROL OVERFLOW AT SEGMENT segment-name

    Explanation: The segment segment-name is more than 16 branch segments from the root segment.

14. FILE file-name HAS ILLEGAL FORMAT

    Explanation: The file file-name contains an object module whose format is not valid.

15. MODULE module-name AMBIGUOUSLY DEFINES P-SECTION p-sect-name

    Explanation: The p-section p-sect-name has been defined in two modules not on a column path and referenced ambiguously.

16. MODULE module-name MULTIPLY DEFINES P-SECTION p-sect-name

    Explanation:

    1 The p-section p-sect-name has been defined in the same segment with different attributes.

    2 A global p-section has been defined in more than one segment along a common path with different attributes.

17. MODULE module-name MULTIPLY DEFINES XFR ADDR IN SEG

    Explanation: segment-name

    This error occurs when more than one module comprising the root has a start address.

18. MODULE module-name ILLEGALLY DEFINES XFR ADDRESS p-sect-name

    Explanation: addr

    The module module-name is in an overlay segment and has a start address. The start address must be in the root segment of the main tree.

19. P-SECTION p-sect-name HAS OVERFLOWED

    Explanation: A section greater than 32K has been created.

20. MODULE module-name AMBIGUOUSLY DEFINES SYMBOL sym-name

    Explanation: Module module-name references or defines a symbol sym-name whose definition cannot be uniquely resolved.

21. MODULE module-name MULTIPLY DEFINES SYMBOL sym-name

    Explanation: Two definitions for the relocatable symbol sym-name have occurred on a common path. Or two definitions for an absolute symbol with the same name but different values have occurred.

22. INSUFFICIENT APRS AVAILABLE TO MAP READ ONLY ROOT

    Explanation: No initial address space can be found to map the read-only portion of a task.

23. SEGMENT seg-name HAS ADDR OVERFLOW: ALLOCATION DELETED

    Explanation: Within a segment, the program has attempted to allocate more than 32K. A map file is produced, but no task image file is produced.

24. ALLOCATION FAILURE ON FILE filename

    Explanation:

    1 The Task Builder could not acquire sufficient contiguous disk space to store the task image file. (If possible, delete unecessary files on disk to make more room available.) Or,

    2 An attempt has been made to write the task file into a directory for which the user does not have write access.

25. I/O ERROR ON OUTPUT FILE file-name

    Explanation: This error may occur on any of the three output files.

26. LOAD ADDR OUT OF RANGE IN MODULE module-name

    Explanation: An attempt has been made to store data in the task image outside the address limits of the segment. This usually indicates incorrect use of an absolute p-section.

27. TRUNCATION ERROR IN MODULE module-name

    Explanation: An attempt has been made to load a global value greater than +127 or less that -128 into a byte. The low-order eight bits are loaded.

28. Number UNDEFINED SYMBOLS SEGMENT seg-name

    Explanation: The Memory Allocation File lists each undefined symbol by segment.

29. INVALID KEYWORD IDENTIFIER

    Explanation: invalid-line

    The invalid-line printed contains an unrecognizable keyword.

30. OPTION SYNTAX ERROR

    Explanation: invalid-line

    The invalid-line printed contains unrecognizable syntax.

31. TOO MANY PARAMETERS

    Explanation: invalid-line

    The invalid-line printed contains a keyword with more parameters than required.

32. ILLEGAL MULTIPLE PARAMETER SETS

    Explanation: The invalid line printed contains multiple parameters for an option keyword which only allows a single parameter.

33. INSUFFICIENT PARAMETERS

    Explanation: invalid-line

    The invalid-line contains a keyword with an insufficient number of parameters to complete the keyword meaning.

34. TASK HAS ILLEGAL MEMORY LIMITS

    Explanation: The highest virtual address of the task is greater than 32K words. Relink the task without a task image file to trace the cause.

35. OVERLAY DIRECTIVE HAS NO OPERANDS

    Explanation: invalid-line

    All overlay directives except .END require operands.

36. ILLEGAL OVERLAY DIRECTIVE

    Explanation: invalid-line

    The invalid-line printed contains an unrecognizable overlay directive.

37. OVERLAY DIRECTIVE SYNTAX ERROR

    Explanation: invalid-line

    The invalid-line printed contains a syntax error.

38. ROOT SEGMENT MULTIPLY DEFINED

    Explanation: invalid-line

    The invalid-line printed contains the second .ROOT directive encountered. Only one .ROOT directive is allowed.

39. LABEL OR NAME IS MULTIPLY DEFINED

    Explanation: invalid-line

    The invalid-line printed contains a name that has already appeared on a .FCTR, .NAME, or .PSECT directive.

40. NO ROOT SEGMENT SPECIFIED

    Explanation: The overlay description did not contain a .ROOT directive.

41. BLANK P-SECTION NAME IS ILLEGAL

    Explanation: invalid-line

    The invalid-line printed contains a .OSECT directive that does not have a p-section name.

42. ILLEGAL P-SECTION/SEGMENT ATTRIBUTE

    Explanation: invalid-line

    The invalid-line printed contains a p-section or segment attribute that is not recognized.

43. ILLEGAL OVERLAY DESCRIPTION OPERATOR

    Explanation: invalid-line

    The invalid-line printed contains an unrecognizable operator in an overlay description.

44. TOO MANY NESTED .ROOT/.FCTR DIRECTIVES

    Explanation: invalid-line

    The invalid-line printed contains a .FCTR directive that exceeds the maximum nesting level (32).

45. TOO MANY PARENTHESES LEVELS

    Explanation: invalid-line

    The invalid-line printed contains a parenthesis that exceeds the maximum nesting level (32).

46. UNBALANCED PARENTHESES

    Explanation: invalid-line

    The invalid-line printed contains unbalanced parentheses.

47. ILLEGAL BASE OR TOP ADDRESS OFFSET

    Explanation: The task is too large to fit into the space allowed by BASE= or TOP= keywords.

48. ILLEGAL LOGICAL UNIT NUMBER

    Explanation: invalid-line

    The invalid-line printed contains a device assignment to a unit number larger than the number of logical units specified by the UNITS keyword or assumed by default if the UNITS keyword is not used.

49. ILLEGAL NUMBER OF LOGICAL UNITS

    Explanation: invalid-line

    The invalid-line printed contains a number of logical units greater that 250.

50. ILLEGAL MAXIMUM EXTENSION

    Explanation: invalid line

    The argument to the MAXEXT option is outside the range 0-2000 (octal).

51. ILLEGAL BASE OR TOP BOUNDARY VALUE

    Explanation: invalid line

52. ILLEGAL POOL USAGE NUMBER SPECIFIED

    Explanation: invalid line

    The pool request is greater than 255 or it is zero.

53. ILLEGAL DEFAULT PRIORITY SPECIFIED

    Explanation: invalid-line

    The invalid-line printed contains a priority greater than 250.

54. ILLEGAL ODT OR TASK VECTOR SIZE

    Explanation: SST vector size specified greater than 32 words.

55. ILLEGAL FILENAME

    Explanation: invalid-line

    The invalid-line printed contains a wildcard (*) in a file specification. The use of wildcards is prohibited.

56. ILLEGAL DEVICE/VOLUME

    Explanation: invalid line

    The device/volume string is too long.

57. LOOKUP FAILURE ON FILE file-name

    Explanation: invalid-line

    The invalid-line printed contains a file-name which cannot be located in the directory.

58. ILLEGAL DIRECTORY

    Explanation: invalid-line

    The invalid-line printed contains an illegal UFD.

59. INCOMPATIBLE REFERENCE TO A LIBRARY P-SECTION p-sect-name

    Explanation: A task has attempted to reference more storage in a shareable global area than exists in the shareable global area definition.

60. ILLEGAL REFERENCE TO LIBRARY P-SECTION p-sect-name

    Explanation: A task attempted to reference a p-sect-name existing in a shareable global area but has not named the library in an SGA keyword.

61. RESIDENT LIBRARY MEMORY ALLOCATION CONFLICT

    Explanation: keyword-string

    One of the following problems has occurred:

    1 More than three shareable global areas have been specified.

    2 The same shareable global area has been specified more than once.

    3 Shareable global areas whose memory allocations overlap have been specified.

    4 BASE or TOP specifications conflict.

62. LOOKUP FAILURE RESIDENT LIBRARY FILE

    Explanation: invalid-line

    No symbol table or task image file found for the shareable global area on SYO under UFD [1,1].

63. Not used.

    Explanation:

64. ILLEGAL PARTITION/COMMON BLOCK SPECIFIED

    Explanation: invalid-line

    User defined base or length not on 32 word bound or user defined length = 0.

65. NO MEMORY AVAILABLE FOR LIBRARY library-name

    Explanation: Insufficient virtual memory available to cover total memory needed by referenced shareable global areas.

66. PIC LIBRARIES MAY NOT REFERENCE OTHER LIBRARIES

    Explanation: invalid-line

67. ILLEGAL APR RESERVATION

    Explanation: APR specified in an SGA keyword that is outside the range 0-7.

68. I/O ERROR LIBRARY IMAGE FILE

    Explanation: An I/O error has occurred during an attempt to open or read the Task Image File of a shareable global area.

69. Not used.

    Explanation:

70. Not used.

    Explanation:

71. INVALID APR RESERVATION

    Explanation: APR specified in an SGA keyword for an absolute shareable global area.

72. COMPLEX RELOCATION ERROR - DIVIDE BY ZERO: MODULE

    Explanation: module-name

    A divisor having the value zero was detected in a complex expression. The result of the divide was set to zero. (Probable cause - division by an undefined global symbol.)

73. WORK FILE I/O ERROR

    Explanation: I/O error during an attempt to reference data stored by the Task Builder in a work file. Possibly an attempt to extend the file when no more space is available on the volume.

74. LOOKUP FAILURE ON SYSTEM LIBRARY FILE

    Explanation: The Task Builder cannot find the system Library (SYO:[1,1]SYSLIB.OLB) to resolve undefined symbols.

75. UNABLE TO OPEN WORK FILE

    Explanation: Work file device is not mounted or has not been initialized as Files-11 or there is no space on the volume.

76. NO VIRTUAL MEMORY STORAGE AVAILABLE

    Explanation: Maximum permissible size of the work file exceeded (no recovery).

77. MODULE module-name NOT IN LIBRARY

    Explanation: The Task Builder could not find the module named on the LB switch in the library.

78. INCORRECT LIBRARY MODULE SPECIFICATION

    Explanation: invalid-line

    The invalid-line contains a module with a non-Radix-50 character.

79. LIBRARY FILE filename HAS INCORRECT FORMAT

    Explanation: A module has been requested from a library file that has an empty module name table.

80. RESIDENT LIBRARY IMAGE HAS INCORRECT FORMAT

    Explanation: invalid-line

    The invalid-line specifies a shareable global area that has one of the following problems:

    1 The shareable global area image has a header.

    2 The shareable global area references another shareable global area with invalid address bounds, that is, not on 4K boundary.

    3 The shareable global area has invalid address bounds.

81. PARTITION partition-name HAS ILLEGAL MEMORY LIMITS

    Explanation: The user has attempted to build a privileged task whose length exceeds 16K.

82. Not used.

    Explanation:

83. ABORTED VIA REQUEST

    Explanation: input-line

    The input-line contains a request from the user to abort the task build.

    84-87. Not used.

88. LIBRARY REFERENCES OVERLAID LIBRARY

    Explanation: It is illegal to build a shareable global area that references another overlaid shareable global area.

89. TASK IMAGE FILE file-name IS NON-CONTIGUOUS

    Explanation: Not enough contiguous disk space could be found to create the task image file. The task image is placed in a non-contiguous file, which must be copied with the COPY/CONTIGUOUS PDS command (or using the PIP utility under MCR) before it can be installed or run.

90. VIRTUAL SECTION HAS ILLEGAL ADDRESS LIMITS

    Explanation: option-line

    The option-line printed contains a VSECT keyword whose base address plus window size exceeds 177777.

91. FILE file-name ATTEMPTED TO STORE DATA IN VIRTUAL SECTION

    Explanation: The file contains a module that has attempted to initialize a virtual section with data.

92. SGA MAPPED ARRAY ALLOCATION TOO LARGE

    Explanation: invalid-line

    The invalid-line printed contains a reference to a shareable global area that has allocated too muck memory in the task's mapped array area. The total allocation exceeds 2.2 million bytes.

93. INVALID REFERENCE TO MAPPED ARRAY BY MODULE module-name

    Explanation: The module has attempted to initialize the mapped array with data. An SPR shoulc be submitted if this problem is caused by Digital-supplied software.

94. END OF FILE REACHED BEFORE .END DIRECTIVE IN file-name

    Explanation: The overlay description file named in this message does not contain an .END directive as required.

95. DUPLICATE LIBRARY NAME

    Explanation: invalid-line

    The SGA name specified has already appeared.

96. SYMBOL sym-name NOT FOUND FOR PATCH

    Explanation: A global symbol specified in a GBLPAT or SYMPAT option cannot be found.

97. SEGMENT seg-name NOT FOUND FOR PATCH

    Explanation: The segment name specified in an ABSPAT, GBLPAT or SYMPAT option cannot be found.

98. ILLEGAL NUMBER OF REGIONS

    Explanation: The argument to the ATRG option is greater than 240

99. INSUFFICIENT APRS TO MAP TASK

    Explanation: There is not enough virtual address space, after allocating libraries, common areas, the task pure area and resident overlays, to map the task root.

D System Standard Error Codes

```
;
; DEFINE THE ERROR CODES RETURNED BY DEVICE HANDLER AND FILE PRIMITIVES
; IN THE FIRST WORD OF THE I/O STATUS BLOCK
; THESE CODES ARE ALSO RETURNED BY FILE CONTROL SERVICES (FCS) IN THE
; BYTE F.ERR IN THE FILE DESCRIPTOR BLOCK (FDB)
; THE BYTE F.ERR+1 IS 0 IF F.ERR CONTAINS A HANDLER OR FCP ERROR CODE.
;

 .ENABL LC

 .MACRO IOERR$ $$$GBL
 .MCALL .IOER.,DEFIN$
 .IF IDN,<$$$GBL>,<DEF$G>
 ...GBL=1
 .IFF
 ...GBL=0
 .ENDC
 .IIF NDF,$$MSG,$$MSG=0

;
; SYSTEM STANDARD CODES, USED BY EXECUTIVE AND DRIVERS
;

 .IOER. IE.BAD,-01.,<Bad parameters>
 .IOER. IE.IFC,-02.,<Invalid function code>
 .IOER. IE.DNR,-03.,<Device not ready>
 .IOER. IE.VER,-04.,<Parity error on device>
 .IOER. IE.ONP,-05.,<Hardware option not present>
 .IOER. IE.SPC,-06.,<Illegal user buffer>
 .IOER. IE.DNA,-07.,<Device not attached>
 .IOER. IE.DAA,-08.,<Device already attached>
 .IOER. IE.DUN,-09.,<Device not attachable>
 .IOER. IE.EOF,-10.,<End of file detected>
 .IOER. IE.EOV,-11.,<End of volume detected>
 .IOER. IE.WLK,-12.,<Write attempted to locked unit>
 .IOER. IE.DAO,-13.,<Data overrun>
 .IOER. IE.SRE,-14.,<Send/receive failure>
 .IOER. IE.ABO,-15.,<Request terminated>
 .IOER. IE.PRI,-16.,<Privilege violation>
 .IOER. IE.RSU,-17.,<Sharable resource in use>
 .IOER. IE.OVR,-18.,<Illegal overlay request>
 .IOER. IE.BYT,-19.,<Odd byte count (or virtual address)>
 .IOER. IE.BLK,-20.,<Logical block number too large>
 .IOER. IE.MOD,-21.,<Invalid UDC module #>
 .IOER. IE.CON,-22.,<UDC connect error>
 .IOER. IE.BBE,-56.,<Bad block on device>
 .IOER. IE.STK,-58.,<Not enough stack space (FCS or FCP)>
 .IOER. IE.FHE,-59.,<Fatal hardware error on device>
 .IOER. IE.EOT,-62.,<End of tape detected>
 .IOER. IE.OFL,-65.,<Device off line>
 .IOER. IE.BCC,-66.,<Block check, CRC, or framing error>
 .IOER. IE.NFW,-69.,<Path lost to partner> ;THIS CODE MUST BE ODD
 .IOER. IE.DIS,-69.,<Path lost to partner> ;DISCONNECTED (SAME AS NFW)
 .IOER. IE.NDR,-72.,<No dynamic space available> ; SEE ALSO IE.UPN
 .IOER. IE.MFE,-91.,<Media Format Error>
 .IOER. IE.TMO,-95.,<Timeout on request> ; see also IS.TMO
 .IOER. IE.CNR,-96.,<Connection rejected>
 .IOER. IE.MII,-99.,<Media inserted incorrectly>
 .IOER. IE.SPI,-100.,<Spindown ignored>

;
; FILE PRIMITIVE CODES
;

 .IOER. IE.NOD,-23.,<Caller's nodes exhausted>
 .IOER. IE.DFU,-24.,<Device full>
 .IOER. IE.IFU,-25.,<Index file full>
 .IOER. IE.NSF,-26.,<No such file>
 .IOER. IE.LCK,-27.,<Locked from read/write access>
 .IOER. IE.HFU,-28.,<File header full>
 .IOER. IE.WAC,-29.,<Accessed for write>
 .IOER. IE.CKS,-30.,<File header checksum failure>
 .IOER. IE.WAT,-31.,<Attribute control list format error>
 .IOER. IE.RER,-32.,<File processor device read error>
 .IOER. IE.WER,-33.,<File processor device write error>
 .IOER. IE.ALN,-34.,<File already accessed on LUN>
 .IOER. IE.SNC,-35.,<File ID, file number check>
 .IOER. IE.SQC,-36.,<File ID, sequence number check>
 .IOER. IE.NLN,-37.,<No file accessed on LUN>
 .IOER. IE.CLO,-38.,<File was not properly closed>
 .IOER. IE.DUP,-57.,<ENTER - duplicate entry in directory>
 .IOER. IE.BVR,-63.,<Bad version number>
 .IOER. IE.BHD,-64.,<Bad file header>
 .IOER. IE.EXP,-75.,<File expiration date not reached>
 .IOER. IE.BTF,-76.,<Bad tape format>
 .IOER. IE.ALC,-84.,<Allocation failure>
 .IOER. IE.ULK,-85.,<Unlock error>
 .IOER. IE.WCK,-86.,<Write check failure>
 .IOER. IE.DSQ,-90.,<Disk quota exceeded>

;
; FILE CONTROL SERVICES CODES
;

 .IOER. IE.NBF,-39.,<OPEN - no buffer space available for file>
 .IOER. IE.RBG,-40.,<Illegal record size>
 .IOER. IE.NBK,-41.,<File exceeds space allocated, no blocks>
 .IOER. IE.ILL,-42.,<Illegal operation on file descriptor block>
 .IOER. IE.BTP,-43.,<Bad record type>
 .IOER. IE.RAC,-44.,<Illegal record access bits set>
 .IOER. IE.RAT,-45.,<Illegal record attributes bits set>
 .IOER. IE.RCN,-46.,<Illegal record number - too large>
 .IOER. IE.2DV,-48.,<Rename - 2 different devices>
 .IOER. IE.FEX,-49.,<Rename - new file name already in use>
 .IOER. IE.BDR,-50.,<Bad directory file>
 .IOER. IE.RNM,-51.,<Can't rename old file system>
 .IOER. IE.BDI,-52.,<Bad directory syntax>
 .IOER. IE.FOP,-53.,<File already open>
 .IOER. IE.BNM,-54.,<Bad file name>
 .IOER. IE.BDV,-55.,<Bad device name>
 .IOER. IE.NFI,-60.,<File ID was not specified>
 .IOER. IE.ISQ,-61.,<Illegal sequential operation>
 .IOER. IE.NNC,-77.,<Not ANSI 'D' format byte count>

;
; NETWORK ACP, PSI, AND DECDATAWAY CODES
;
```

```
 .IOER. IE.NNN,-68.,<No such node>
 .IOER. IE.BLB,-70.,<Bad logical buffer>
 .IOER. IE.URJ,-73.,<Connection rejected by user>
 .IOER. IE.NRJ,-74.,<Connection rejected by network>
 .IOER. IE.NDA,-78.,<No data available>
 .IOER. IE.IQU,-91.,<Inconsistent qualifier usage>
 .IOER. IE.RES,-92.,<Circuit reset during operation>
 .IOER. IE.TML,-93.,<Too many links to task>
 .IOER. IE.NNT,-94.,<Not a network task>
 .IOER. IE.UKN,-97.,<Unknown name>

;
; ICS/ICR ERROR CODES
;
 .IOER. IE.NLK,-79.,<Task not linked to specified ICS/ICR interrupts>
 .IOER. IE.NST,-80.,<Specified task not installed>
 .IOER. IE.FLN,-81.,<Device offline when offline request was issued>


;
; TTY ERROR CODES
;

 .IOER. IE.IES,-82.,<Invalid escape sequence>
 .IOER. IE.PES,-83.,<Partial escape sequence>


;
; RECONFIGURATION CODES
;

 .IOER. IE.ICE,-47.,<Internal consistancy error>
 .IOER. IE.ONL,-67.,<Device online>
 .IOER. IE.SZE,-98.,<Unable to size device>

;
; PCL ERROR CODES
;

 .IOER. IE.NTR,-87.,<Task not triggered>
 .IOER. IE.REJ,-88.,<Transfer rejected by receiving CPU>
 .IOER. IE.FLG,-89.,<Event flag already specified>


;
; SUCCESSFUL RETURN CODES---
;

 DEFIN$ IS.PND,+00. ;OPERATION PENDING
 DEFIN$ IS.SUC,+01. ;OPERATION COMPLETE, SUCCESS
 DEFIN$ IS.RDD,+02. ;FLOPPY DISK SUCCESSFUL COMPLETION
    ;OF A READ PHYSICAL, AND DELETED
    ;DATA MARK WAS SEEN IN SECTOR HEADER
 DEFIN$ IS.TNC,+02. ;(PCL) SUCCESSFUL TRANSFER BUT MESSAGE
    ;TRUNCATED (RECEIVE BUFFER TOO SMALL).
 DEFIN$ IS.CHW,+04. ;(IBM COMM) DATA READ WAS RESULT OF
    ;IBM HOST CHAINED WRITE OPERATION
 DEFIN$ IS.BV,+05. ;(A/D READ) AT LEAST ONE BAD VALUE
    ;WAS READ (REMAINDER MAY BE GOOD).
    ;BAD CHANNEL IS INDICATED BY A
    ;NEGATIVE VALUE IN THE BUFFER.
 DEFIN$ IS.DAO,+02. ;SUCCESSFUL BUT WITH DATA OVERRUN
    ;(NOT TO BE CONFUSED WITH IE.DAO)
 DEFIN$ IS.BBR,+04 ; MSCP success status with BBR flag asserted

;
; TTY SUCCESS CODES
;
```

```
 DEFIN$ IS.CR,<15*400+1>  ;CARRIAGE RETURN WAS TERMINATOR
 DEFIN$ IS.ESC,<33*400+1> ;ESCAPE (ALTMODE) WAS TERMINATOR
 DEFIN$ IS.CC,<3*400+1> ;CONTROL-C WAS TERMINATOR
 DEFIN$ IS.ESQ,<233*400+1>  ;ESCAPE SEQUENCE WAS TERMINATOR
 DEFIN$ IS.PES,<200*400+1>  ;PARTIAL ESCAPE SEQUENCE TERMINATOR
 DEFIN$ IS.EOT,<4*400+1>  ;EOT WAS TERMINATOR (BLOCK MODE INPUT)
 DEFIN$ IS.TAB,<11*400+1> ;TAB WAS TERMINATOR (FORMS MODE INPUT)
 DEFIN$ IS.TMO,+2. ;REQUEST TIMED OUT

;
; Professional Bisync Success Codes
;
 DEFIN$ IS.RVI,+2. ; DATA SUCC. XMITTED; HOST ACKED W/RVI
 DEFIN$ IS.CNV,+3. ; DATA SUCC. XMITTED; HOST ACKED W/CONVERSATION
 DEFIN$ IS.XPT,+5. ; DATA SUCC. RECVD IN TRANSPARENT MODE
;
; Professional Bisync Abort Codes
;
; These codes are returned in the high byte of the first word of the IOSB
; when the low byte contains IE.ABO.
;
 DEFIN$ SB.KIL,-1. ; ABORTED BY IO.KIL
 DEFIN$ SB.ACK,-2. ; ABORTED BECAUSE TOO MANY ACKS RECD OUT OF SEQ
 DEFIN$ SB.NAK,-3. ; ABORTED BECAUSE NAK THRESHOLD EXCEEDED
 DEFIN$ SB.ENQ,-4. ; ABORTED BECAUSE ENQ THRESHOLD EXCEEDED
 DEFIN$ SB.BOF,-5. ; ABORTED BECAUSE OF IO.RLB BUFFER OVERFLOW
 DEFIN$ SB.TMO,-6. ; ABORTED BECAUSE OF TIMEOUT
 DEFIN$ SB.DIS,-7. ; ABORTED BECAUSE HOST DISCONNECTED W/ DLE, EOT

;+
; The following bit values are used with the IO.STC queue  I/O function.
; The MSCP disk class  handler will  expect a value in sixth word of the
; requestor's queue I/O parameter list.   A value of zero is will always
; default to a volume valid function.  Any other value outside the range
; defined below will produce a bad function error to be returned in word
; zero of the requestor's I/O status block (IOSB+0).
;-
 DEFIN$ VV.ONL,0 ; Set volume valid ("ONLINE")
 DEFIN$ VV.AVL,020000 ; Reset volume valid ("AVAILABLE")
 DEFIN$ VV.SPD,020001 ; Reset volume valid, spin unit down
 DEFIN$ VV.SWP,040000 ; Software Write Protect
 DEFIN$ VV.GUS,100000 ; Size unit ("GET UNIT STATUS")

;  ******
;
; THE NEXT AVAILABLE ERROR NUMBER IS: -101.
; NUMBER -71. IS AVAILABLE AND MAY BE REASSIGNED
; AFTER 11M V4.0 AND M-PLUS V2.0
;
;  *****
 .IF EQ,$$MSG
 .MACRO IOERR$ A
 .ENDM IOERR$
 .ENDC
 .ENDM IOERR$
;
; DEFINE THE DIRECTIVE ERROR CODES RETURNED IN THE DIRECTIVE STATUS WORD
;
; FILE CONTROL SERVICES (FCS) RETURNS THESE CODES IN THE BYTE F.ERR
; OF THE FILE DESCRIPTOR BLOCK (FDB).  TO DISTINGUISH THEM FROM THE
; OVERLAPPING CODES FROM HANDLER AND FILE PRIMITIVES, THE BYTE
; F.ERR+1 IN THE FDB WILL BE NEGATIVE FOR A DIRECTIVE ERROR CODE.
;
 .MACRO DRERR$ $$$GBL
 .MCALL .QIOE.,DEFIN$
 .IF IDN,<$$$GBL>,<DEF$G>
 ...GBL=1
 .IFF
 ...GBL=0
 .ENDC
 .IIF NDF,$$MSG,$$MSG=0
;
; STANDARD ERROR CODES RETURNED BY DIRECTIVES IN THE DIRECTIVE STATUS WORD
;
 .QIOE. IE.UPN,-01.,<Insufficient dynamic storage> ; SEE ALSO IE.NDR
 .QIOE. IE.INS,-02.,<Specified task not installed>
 .QIOE. IE.PTS,-03.,<Partition too small for task>
 .QIOE. IE.UNS,-04.,<Insufficient dynamic storage for send>
 .QIOE. IE.ULN,-05.,<Un-assigned LUN>
 .QIOE. IE.HWR,-06.,<Device handler not resident>
 .QIOE. IE.ACT,-07.,<Task not active>
 .QIOE. IE.ITS,-08.,<Directive inconsistent with task state>
 .QIOE. IE.FIX,-09.,<Task already fixed/unfixed>
 .QIOE. IE.CKP,-10.,<Issuing task not checkpointable>
 .QIOE. IE.TCH,-11.,<Task is checkpointable>
 .QIOE. IE.RBS,-15.,<Receive buffer is too small>
 .QIOE. IE.PRI,-16.,<Privilege violation>
 .QIOE. IE.RSU,-17.,<Resource in use>
 .QIOE. IE.NSW,-18.,<No swap space available>
 .QIOE. IE.ILV,-19.,<Illegal vector specified>
 .QIOE. IE.ITN,-20.,<Invalid table number>
 .QIOE. IE.LNF,-21.,<Logical name not found>
;
;
;
 .QIOE. IE.AST,-80.,<Directive issued/not issued from AST>
 .QIOE. IE.MAP,-81.,<Illegal mapping specified>
 .QIOE. IE.IOP,-83.,<Window has I/O in progress>
 .QIOE. IE.ALG,-84.,<Alignment error>
 .QIOE. IE.WOV,-85.,<Address window allocation overflow>
 .QIOE. IE.NVR,-86.,<Invalid region ID>
 .QIOE. IE.NVW,-87.,<Invalid address window ID>
 .QIOE. IE.ITP,-88.,<Invalid TI parameter>
 .QIOE. IE.IBS,-89.,<Invalid send buffer size ( .GT. 255.)>
 .QIOE. IE.LNL,-90.,<LUN locked in use>
 .QIOE. IE.IUI,-91.,<Invalid UIC>
 .QIOE. IE.IDU,-92.,<Invalid device or unit>
 .QIOE. IE.ITI,-93.,<Invalid time parameters>
 .QIOE. IE.PNS,-94.,<Partition/region not in system>
 .QIOE. IE.IPR,-95.,<Invalid priority ( .GT. 250.)>
 .QIOE. IE.ILU,-96.,<Invalid LUN>
 .QIOE. IE.IEF,-97.,<Invalid event flag ( .GT. 64.)>
 .QIOE. IE.ADP,-98.,<Part of DPB out of user's space>
 .QIOE. IE.SDP,-99.,<DIC or DPB size invalid>
;
; SUCCESS CODES FROM DIRECTIVES - PLACED IN THE DIRECTIVE STATUS WORD
;
 DEFIN$ IS.CLR,0 ;EVENT FLAG WAS CLEAR
    ;FROM CLEAR EVENT FLAG DIRECTIVE
 DEFIN$ IS.SET,2 ;EVENT FLAG WAS SET
    ;FROM SET EVENT FLAG DIRECTIVE
 DEFIN$ IS.SPD,2 ;TASK WAS SUSPENDED
    ;
 DEFIN$ IS.ACT,3 ; Data sent to non-suspended task
    ;
 DEFIN$ IS.SUP,3 ;LOGICAL NAME SUPERSEDED
    ;
```

```
;
;
 .IF EQ,$$MSG
 .MACRO DRERR$ A
 .ENDM DRERR$
 .ENDC
 .ENDM DRERR$
;
; DEFINE THE GENERAL I/O FUNCTION CODES - DEVICE INDEPENDENT
;
 .MACRO FILIO$ $$$GBL
 .MCALL .WORD.,DEFIN$
 .IF IDN,<$$$GBL>,<DEF$G>
 ...GBL=1
 .IFF
 ...GBL=0
 .ENDC
;
; GENERAL I/O QUALIFIER BYTE DEFINITIONS
;
 .WORD. IQ.X,001,000 ;NO ERROR RECOVERY
 .WORD. IQ.Q,002,000 ;QUEUE REQUEST IN EXPRESS QUEUE
 .WORD. IQ.S,004,000 ;SYNONYM FOR IQ.UMD
 .WORD. IQ.UMD,004,000 ;USER MODE DIAGNOSTIC STATUS REQUIRED
 .WORD. IQ.LCK,200,000 ;MODIFY IMPLIED LOCK FUNCTION
;
; EXPRESS QUEUE COMMANDS
;

 .WORD. IO.KIL,012,000 ;KILL CURRENT REQUEST
 .WORD. IO.RDN,022,000 ;I/O RUNDOWN
 .WORD. IO.UNL,042,000 ;UNLOAD I/O HANDLER TASK
 .WORD. IO.LTK,050,000 ;LOAD A TASK IMAGE FILE
 .WORD. IO.RTK,060,000 ;RECORD A TASK IMAGE FILE
 .WORD. IO.SET,030,000 ;SET CHARACTERISTICS FUNCTION
;
; GENERAL DEVICE HANDLER CODES
;
 .WORD. IO.WLB,000,001 ;WRITE LOGICAL BLOCK
 .WORD. IO.RLB,000,002 ;READ LOGICAL BLOCK
 .WORD. IO.LOV,010,002 ;LOAD OVERLAY (DISK DRIVER)
 .WORD. IO.LDO,110,002 ;LOAD D-SPACE OVERLAY (DISK)
 .WORD. IO.ATT,000,003 ;ATTACH A DEVICE TO A TASK
 .WORD. IO.DET,000,004 ;DETACH A DEVICE FROM A TASK
;
; DIRECTORY PRIMITIVE CODES
;
 .WORD. IO.FNA,000,011 ;FIND FILE NAME IN DIRECTORY
 .WORD. IO.RNA,000,013 ;REMOVE FILE NAME FROM DIRECTORY
 .WORD. IO.ENA,000,014 ;ENTER FILE NAME IN DIRECTORY
;
; FILE PRIMITIVE CODES
;
 .WORD. IO.CLN,000,007 ;CLOSE OUT LUN
 .WORD. IO.ULK,000,012 ;UNLOCK BLOCK
 .WORD. IO.ACR,000,015 ;ACCESS FOR READ
 .WORD. IO.ACW,000,016 ;ACCESS FOR WRITE
 .WORD. IO.ACE,000,017 ;ACCESS FOR EXTEND
 .WORD. IO.DAC,000,020 ;DE-ACCESS FILE
 .WORD. IO.RVB,000,021 ;READ VIRITUAL BLOCK
 .WORD. IO.WVB,000,022 ;WRITE VIRITUAL BLOCK
 .WORD. IO.EXT,000,023 ;EXTEND FILE
 .WORD. IO.CRE,000,024 ;CREATE FILE
 .WORD. IO.DEL,000,025 ;DELETE FILE
 .WORD. IO.RAT,000,026 ;READ FILE ATTRIBUTES
 .WORD. IO.WAT,000,027 ;WRITE FILE ATTRIBUTES
 .WORD. IO.APV,010,030 ;PRIVILEGED ACP CONTROL
 .WORD. IO.APC,000,030 ;ACP CONTROL
;
;
 .MACRO FILIO$ A
 .ENDM FILIO$
 .ENDM FILIO$
;
; DEFINE THE I/O FUNCTION CODES THAT ARE SPECIFIC TO INDIVIDUAL DEVICES
;
 .MACRO SPCIO$ $$$GBL
 .MCALL .WORD.,DEFIN$
 .IF IDN,<$$$GBL>,<DEF$G>
 ...GBL=1
 .IFF
 ...GBL=0
 .ENDC
;
; I/O FUNCTION CODES FOR SPECIFIC DEVICE DEPENDENT FUNCTIONS
;
 .WORD. IO.WLV,100,001 ;(DECTAPE) WRITE LOGICAL REVERSE
 .WORD. IO.WLS,010,001 ;(COMM.) WRITE PRECEDED BY SYNC TRAIN
 .WORD. IO.WNS,020,001 ;(COMM.) WRITE, NO SYNC TRAIN
 .WORD. IO.WAL,010,001 ;(TTY) WRITE PASSING ALL CHARACTERS
 .WORD. IO.WMS,020,001 ;(TTY) WRITE SUPPRESSIBLE MESSAGE
 .WORD. IO.CCO,040,001 ;(TTY) WRITE WITH CANCEL CONTROL-O
 .WORD. IO.WBT,100,001 ;(TTY) WRITE WITH BREAKTHROUGH
 .WORD. IO.WLT,010,001 ;(DISK) WRITE LAST TRACK
 .WORD. IO.WLC,020,001 ;(DISK) WRITE LOGICAL W/ WRITECHECK
 .WORD. IO.MSW,030,001 ;(MSCP) MULTICOPY STRUCTURE WRITE
 .WORD. IO.WPB,040,001 ;(DISK) WRITE PHYSICAL BLOCK
 .WORD. IO.WDD,140,001 ;(FLOPPY DISK) WRITE PHYSICAL W/ DELETED DATA
 .WORD. IO.RSN,140,002 ;(MSCP DISK) READ VOLUME SERIAL NUMBER
 .WORD. IO.RLV,100,002 ;(MAGTAPE,DECTAPE) READ REVERSE
 .WORD. IO.RST,001,002 ;(TTY) READ WITH SPECIAL TERMINATOR
 .WORD. IO.RAL,010,002 ;(TTY) READ PASSING ALL CHARACTERS
 .WORD. IO.RNE,020,002 ;(TTY) READ WITHOUT ECHO
 .WORD. IO.RNC,040,002 ;(TTY) READ - NO LOWER CASE CONVERT
 .WORD. IO.RTM,200,002 ;(TTY) READ WITH TIME OUT
 .WORD. IO.RDB,200,002 ;(CARD READER) READ BINARY MODE
 .WORD. IO.SCF,200,002 ;(DISK) SHADOW COPY FUNCTION
 .WORD. IO.RHD,010,002 ;(COMM.) READ, STRIP SYNC
 .WORD. IO.RNS,020,002 ;(COMM.) READ, DON'T STRIP SYNC
 .WORD. IO.CRC,040,002 ;(COMM.) READ, DON'T CLEAR CRC
 .WORD. IO.RPB,040,002 ;(DISK) READ PHYSICAL BLOCK
 .WORD. IO.RDF,240,002 ;(DISK) READ DISK FORMAT
 .WORD. IO.RLC,020,002 ;(DISK,MAGTAPE) READ LOGICAL W/ READCHECK
 .WORD. IO.MSR,030,002 ;(MSCP) MULTICOPY STRUCTURE READ
 .WORD. IO.ATA,010,003 ;(TTY) ATTACH WITH AST'S
 .WORD. IO.GTS,000,005 ;(TTY) GET TERMINAL SUPPORT CHARACTERISTICS
 .WORD. IO.R1C,000,005 ;(AFC,AD01,UDC) READ SINGLE CHANNEL
 .WORD. IO.INL,000,005 ;(COMM.) INITIALIZATION FUNCTION
 .WORD. IO.TRM,010,005 ;(COMM.) TERMINATION FUNCTION
 .WORD. IO.RWD,000,005 ;(MAGTAPE,DECTAPE) REWIND
 .WORD. IO.SPB,020,005 ;(MAGTAPE) SPACE "N" BLOCKS
 .WORD. IO.RPL,020,005 ;(DISK) REPLACE LOGICAL BLOCK (RESECTOR)
 .WORD. IO.SPF,040,005 ;(MAGTAPE) SPACE "N" EOF MARKS
 .WORD. IO.STC,100,005 ;SET CHARACTERISTIC
 .WORD. IO.SMD,110,005 ;(FLOPPY DISK) SET MEDIA DENSITY
 .WORD. IO.SEC,120,005 ;SENSE CHARACTERISTIC
 .WORD. IO.RWU,140,005 ;(MAGTAPE,DECTAPE) REWIND AND UNLOAD
```

```
.WORD. IO.SMO,160,005 ;(MAGTAPE) MOUNT & SET CHARACTERISTICS
.WORD. IO.HNG,000,006 ;(TTY) HANGUP DIAL-UP LINE
.WORD. IO.HLD,100,006 ;(TMS) HANGUP BUT LEAVE LINE ON HOLD
.WORD. IO.BRK,200,006 ;(PRO/TTY) SEND SHORT OR LONG BREAK
.WORD. IO.RBC,000,006 ;READ MULTICHANNELS (BUFFER DEFINES CHANNELS)
.WORD. IO.MOD,000,006 ;(COMM.) SETMODE FUNCTION FAMILY
.WORD. IO.HDX,010,006 ;(COMM.) SET UNIT HALF DUPLEX
.WORD. IO.FDX,020,006 ;(COMM.) SET UNIT FULL DUPLEX
.WORD. IO.SYN,040,006 ;(COMM.) SPECIFY SYNC CHARACTER
.WORD. IO.EOF,000,006 ;(MAGTAPE) WRITE EOF
.WORD. IO.ERS,020,006 ;(MAGTAPE) ERASE TAPE
.WORD. IO.DSE,040,006 ;(MAGTAPE) DATA SECURITY ERASE
.WORD. IO.RTC,000,007 ;READ CHANNEL - TIME BASED
.WORD. IO.SAO,000,010 ;(UDC) SINGLE CHANNEL ANALOG OUTPUT
.WORD. IO.SSO,000,011 ;(UDC) SINGLE SHOT, SINGLE POINT
.WORD. IO.RPR,000,011 ;(TTY) READ WITH PROMPT
.WORD. IO.MSO,000,012 ;(UDC) SINGLE SHOT, MULTI-POINT
.WORD. IO.RTT,001,012 ;(TTY) READ WITH TERMINATOR TABLE
.WORD. IO.SLO,000,013 ;(UDC) LATCHING, SINGLE POINT
.WORD. IO.MLO,000,014 ;(UDC) LATCHING, MULTI-POINT
.WORD. IO.LED,000,024 ;(LPS11) WRITE LED DISPLAY LIGHTS
.WORD. IO.SDO,000,025 ;(LPS11) WRITE DIGITAL OUTPUT REGISTER
.WORD. IO.SDI,000,026 ;(LPS11) READ DIGITAL INPUT REGISTER
.WORD. IO.SCS,000,026 ;(UDC) CONTACT SENSE, SINGLE POINT
.WORD. IO.REL,000,027 ;(LPS11) WRITE RELAY
.WORD. IO.MCS,000,027 ;(UDC) CONTACT SENSE, MULTI-POINT
.WORD. IO.ADS,000,030 ;(LPS11) SYNCHRONOUS A/D SAMPLING
.WORD. IO.CCI,000,030 ;(UDC) CONTACT INT - CONNECT
.WORD. IO.LOD,000,030 ;(LPA11) LOAD MICROCODE
.WORD. IO.MDI,000,031 ;(LPS11) SYNCHRONOUS DIGITAL INPUT
.WORD. IO.DCI,000,031 ;(UDC) CONTACT INT - DISCONNECT
.WORD. IO.PAD,000,031 ;(PSI) DIRECT CONTROL OF X.29 PAD
.WORD. HT.RPP,010,000 ;(PSI) RESET PAD PARAMETERS SUBFUNCTION
.WORD. IO.XMT,000,031 ;(COMM.) TRANSMIT SPECIFIED BLOCK WITH ACK
.WORD. IO.XNA,010,031 ;(COMM.) TRANSMIT WITHOUT ACK
.WORD. IO.INI,000,031 ;(LPA11) INITIALIZE
.WORD. IO.HIS,000,032 ;(LPS11) SYNCHRONOUS HISTOGRAM SAMPLING
.WORD. IO.RCI,000,032 ;(UDC) CONTACT INT - READ
.WORD. IO.RCV,000,032 ;(COMM.) RECEIVE DATA IN BUFFER SPECIFIED
.WORD. IO.CLK,000,032 ;(LPA11) START CLOCK
.WORD. IO.CSR,000,032 ;(BUS SWITCH) READ CSR REGISTER
.WORD. IO.MDO,000,033 ;(LPS11) SYNCHRONOUS DIGITAL OUTPUT
.WORD. IO.CTI,000,033 ;(UDC) TIMER - CONNECT
.WORD. IO.CON,000,033 ;(COMM.) CONNECT FUNCTION
   ;(VT11) - CONNECT TASK TO DISPLAY PROCESSOR
   ;(BUS SWITCH) CONNECT TO SPECIFIED BUS
   ;(COMM./PRO) DIAL TELEPHONE AND ORIGINATE
.WORD. IO.ORG,010,033 ;(COMM.) INITIATE CONNECTION IN ORIGINATE MODE
.WORD. IO.ANS,020,033 ;(COMM.) INITIATE CONNECTION IN ANSWER MODE
.WORD. IO.STA,000,033 ;(LPA11) START DATA TRANSFER
   ;(XJDRV) - SHOW STATE
.WORD. IO.DTI,000,034 ;(UDC) TIMER - DISCONNECT
.WORD. IO.DIS,000,034 ;(COMM.) DISCONNECT FUNCTION
   ;(VT11) - DISCONNECT TASK FROM DISPLAY PROCESSOR
   ;(BUS SWITCH) SWITCHED BUS DISCONNECT
.WORD. IO.MDA,000,034 ;(LPS11) SYNCHRONOUS D/A OUTPUT
.WORD. IO.DPT,010,034 ;(BUS SWITCH) DISCONNECT TO SPECIF PORT NO.
.WORD. IO.RTI,000,035 ;(UDC) TIMER - READ
.WORD. IO.CTL,000,035 ;(COMM.) NETWORK CONTROL FUNCTION
.WORD. IO.STP,000,035 ;(LPS11,LPA11) STOP IN PROGRESS FUNCTION
   ;(VT11) - STOP DISPLAY PROCESSOR
.WORD. IO.SWI,000,035 ;(BUS SWITCH) SWITCH BUSSES
.WORD. IO.CNT,000,036 ;(VT11) - CONTINUE DISPLAY PROCESSOR
```

```
   ;(XJDRV) - SHOW COUNTERS
 .WORD. IO.ITI,000,036 ;(UDC) TIMER - INITIALIZE
;
; PRO 300 SERIES BITMAP FUNCTIONS
;
; NOTE: THESE FUNCTIONS ARE FOR DEC USE ONLY AND ARE SUBJECT TO CHANGE
;

 .WORD. IO.RSD,030,014 ; READ SPECIAL DATA
 .WORD. IO.WSD,010,013 ; WRITE SPECIAL DATA
 DEFIN$ SD.TXT,0 ; TEXT DATA TYPE FOR SPECIAL DATA
 DEFIN$ SD.GDS,1 ; GIDIS DATA TYPE FOR SPECIAL DATA

;
; PROFESSIONAL 300 BISYNC DRIVER (XJDRV) FUNCTIONS
;
 .WORD. SB.PRT,020,003 ; ATTACH AS A PRINTER
 .WORD. SB.CLR,010,036 ; CLEAR COUNTERS (IO.CNT SUBFUNCTION)
 .WORD. SB.RDY,010,033 ; SET DEVICE STATE READY (IO.STA SUBFUNC)
 .WORD. SB.NRD,020,033 ; SET DEVICE STATE NOT READY

 .WORD. IO.LBK,000,035 ; PERFORM LOOPBACK TEST

 .WORD. SB.CBL,010,035 ; PERFORM CABLE LOOPBACK TEST
 .WORD. SB.CLK,020,035 ; DEVICE PERFORMS LINE CLOCKING
;
; COMMUNICATIONS FUNCTIONS
;

 .WORD. IO.CPR,010,033 ;CONNECT NO TIMEOUTS
 .WORD. IO.CAS,020,033 ;CONNECT WITH AST
 .WORD. IO.CRJ,040,033 ;CONNECT REJECT
 .WORD. IO.CBO,110,033 ;BOOT CONNECT
 .WORD. IO.CTR,210,033 ;TRANSPARENT CONNECT
 .WORD. IO.GNI,010,035 ;GET NODE INFORMATION
 .WORD. IO.GLI,020,035 ;GET LINK INFORMATION
 .WORD. IO.GLC,030,035 ;GET LINK INFO CLEAR COUNTERS
 .WORD. IO.GRI,040,035 ;GET REMOTE NODE INFORMATION
 .WORD. IO.GRC,050,035 ;GET REMOTE NODE ERROR COUNTS
 .WORD. IO.GRN,060,035 ;GET REMOTE NODE NAME
 .WORD. IO.CSM,070,035 ;CHANGE SOLO MODE
 .WORD. IO.CIN,100,035 ;CHANGE CONNECTION INHIBIT
 .WORD. IO.SPW,110,035 ;SPECIFY NETWORK PASSWORD
 .WORD. IO.CPW,120,035 ;CHECK NETWORK PASSWORD.
 .WORD. IO.NLB,130,035 ;NSP LOOPBACK
 .WORD. IO.DLB,140,035 ;DDCMP LOOPBACK

;
; ICS/ICR I/O FUNCTIONS
;
 .WORD. IO.CTY,000,007 ;CONNECT TO TERMINAL INTERRUPTS
 .WORD. IO.DTY,000,015 ;DISCONNECT FROM TERMINAL INTERRUPTS
 .WORD. IO.LDI,000,016 ;LINK TO DIGITAL INTERRUPTS
 .WORD. IO.UDI,010,023 ;UNLINK FROM DIGITAL INTERRUPTS
 .WORD. IO.LTI,000,017 ;LINK TO COUNTER MODULE INTERRUPTS
 .WORD. IO.UTI,020,023 ;UNLINK FROM COUNTER MODULE INTERRUPTS
 .WORD. IO.LTY,000,020 ;LINK TO REMOTE TERMINAL INTERRUPTS
 .WORD. IO.UTY,030,023 ;UNLINK FROM REMOTE TERMINAL INTERRUPTS
 .WORD. IO.LKE,000,024 ;LINK TO ERROR INTERRUPTS
 .WORD. IO.UER,040,023 ;UNLINK FROM ERROR INTERRUPTS
 .WORD. IO.NLK,000,023 ;UNLINK FROM ALL INTERRUPTS
 .WORD. IO.ONL,000,037 ;UNIT ONLINE
 .WORD. IO.FLN,000,025 ;UNIT OFFLINE
 .WORD. IO.RAD,000,021 ;READ ACTIVATING DATA
```

```
;
; IP11 I/O FUNCTIONS
;
 .WORD. IO.MAO,010,007 ;MULTIPLE ANALOG OUTPUTS
 .WORD. IO.LEI,010,017 ;LINK EVENT FLAGS TO INTERRUPT
 .WORD. IO.RDD,010,020 ;READ DIGITAL DATA
 .WORD. IO.RMT,020,020 ;READ MAPPING TABLE
 .WORD. IO.LSI,000,022 ;LINK TO DSI INTERRUPTS
 .WORD. IO.UEI,050,023 ;UNLINK EVENT FLAGS
 .WORD. IO.USI,060,023 ;UNLINK FROM DSI INTERRUPTS
 .WORD. IO.CSI,000,026 ;CONNECT TO DSI INTERRUPTS
 .WORD. IO.DSI,000,027 ;DISCONNECT FROM DSI INTERRUPTS
 .WORD. IO.RAM,000,032 ;READ ANALOG MAPPING TABLES

;
; PCL11 I/O FUNCTIONS
;

 .WORD. IO.ATX,000,001 ;ATTEMPT TRANSMISSION
 .WORD. IO.ATF,000,002 ;ACCEPT TRANSFER
 .WORD. IO.CRX,000,031 ;CONNECT FOR RECEPTION
 .WORD. IO.DRX,000,032 ;DISCONNECT FROM RECEPTION
 .WORD. IO.RTF,000,033 ;REJECT TRANSFER

 .MACRO SPCIO$ A
 .ENDM SPCIO$
 .ENDM SPCIO$
;
; DEFINE THE I/O CODES FOR USER-MODE DIAGNOSITCS.  ALL DIAGNOSTIC
; FUNCTION ARE IMPLEMENTED AS A SUBFUNCTION OF I/O CODE 10 (OCTAL).
;

 .MACRO UMDIO$ $$$GBL
 .MCALL .WORD.,DEFIN$
 .IF IDN <$$$GBL>,<DEF$G>
...GBL=1
 .IFF
...GBL=0
 .ENDC

;
; DEFINE THE GENERAL USER-MODE I/O QUALIFIER BIT.
;

 .WORD. IQ.UMD,004,000 ;USER MODE DIAGNOSTIC REQUEST

;
; DEFINE USER-MODE DIAGNOSTIC FUNCTIONS.
;
```

```
 .WORD. IO.HMS,000,010 ;(DISK) HOME SEEK OR RECALIBRATE
 .WORD. IO.BLS,010,010 ;(DISK) BLOCK SEEK
 .WORD. IO.OFF,020,010 ;(DISK) OFFSET POSITION
 .WORD. IO.RDH,030,010 ;(DISK) READ DISK HEADER
 .WORD. IO.WDH,040,010 ;(DISK) WRITE DISK HEADER
 .WORD. IO.WCK,050,010 ;(DISK) WRITECHECK (NON-TRANSFER)
 .WORD. IO.RNF,060,010 ;(DECTAPE) READ BLOCK NUMBER FORWARD
 .WORD. IO.RNR,070,010 ;(DECTAPE) READ BLOCK NUMBER REVERSE
 .WORD. IO.LPC,100,010 ;(MAGTAPE) READ LONGITUDINAL PARITY CHAR
 .WORD. IO.RTD,120,010 ;(DISK) READ TRACK DESCRIPTOR
 .WORD. IO.WTD,130,010 ;(DISK) WRITE TRACK DESCRIPTOR
 .WORD. IO.TDD,140,010 ;(DISK) WRITE TRACK DESCRIPTOR DISPLACED
 .WORD. IO.DGN,150,010 ;DIAGNOSE MICRO PROCESSOR FIRMWARE
 .WORD. IO.WPD,160,010 ;(DISK) WRITE PHYSICAL BLOCK
 .WORD. IO.RPD,170,010 ;(DISK) READ PHYSICAL BLOCK
 .WORD. IO.CER,200,010 ;(DISK) READ CE BLOCK
 .WORD. IO.CEW,210,010 ;(DISK) WRITE CE BLOCK

;
; MACRO REDEFINITION TO NULL
;

 .MACRO UMDIO$ A
 .ENDM

 .ENDM UMDIO$
;
; HANDLER ERROR CODES RETURNED IN I/O STATUS BLOCK ARE DEFINED THROUGH THIS
; MACRO WHICH THEN CONDITIONALLY INVOKES THE MESSAGE GENERATING MACRO
; FOR THE QIOSYM.MSG FILE
;
 .MACRO .IOER. SYM,LO,MSG
 DEFIN$ SYM,LO
 .IF GT,$$MSG
 .MCALL .IOMG.
 .IOMG. SYM,LO,<MSG>
 .ENDC
 .ENDM .IOER.
;
; I/O ERROR CODES ARE DEFINED THOUGH THIS MACRO WHICH THEN INVOKES THE
; ERROR MESSAGE GENERATING MACRO, ERROR CODES -129 THROUGH -256
; ARE USED IN THE QIOSYM.MSG FILE
;
 .MACRO .QIOE. SYM,LO,MSG
 DEFIN$ SYM,LO
 .IF GT,$$MSG
 .MCALL .IOMG.
 .IOMG. SYM,<LO-128.>,<MSG>
 .ENDC
 .ENDM .QIOE.
;
; CONDITIONALLY GENERATE DATA FOR WRITING A MESSAGE FILE
;
 .MACRO .IOMG. SYM,LO,MSG
 .WORD -^O<LO>
 .ENABL LC
 .ASCIZ ^MSG^
 .DSABL LC
 .EVEN
 .IIF LT,^O<$$$MAX+<LO>>,$$$MAX=-^O<LO>
 .ENDM .IOMG.
;
; DEFINE THE SYMBOL SYM WHERE LO IS IS THE LOW ORDER BYTE, HI IS THE HIGH BYTE
;
```

```
.MACRO .WORD. SYM,LO,HI
DEFIN$ SYM,<HI*400+LO>
.ENDM .WORD.

.DSABL LC
```

E Node Pool Status Program

The Node Pool Status Program (POOL) provides the privileged user with statistics about node pool usage, and prints these statistics on the terminal. Once the POOL program is initiated, it executes every two seconds and prints the number of nodes available (unused) in the pool along with the largest contiguous amount of node space. All numbers (decimal) are in terms of 8-word nodes.

Type the following command to cause POOL to execute:

```
MCR>RUN POOL ESC
```

The following example shows sample POOL execution and termination.

```
MCR>RUN POOL ESC
NODES = 745    LARGEST  HOLE = 113
NODES = 759    LARGEST  HOLE = 113
NODES = 751    LARGEST  HOLE = 113
NODES = 753    LARGEST  HOLE = 113
NODES = 746    LARGEST  HOLE = 113
NODES = 742    LARGEST  HOLE = 113
Ctrl/Z
NODES = 728  LARGEST  HOLE = 113
LARGEST NUMBER OF NODES = 759
SMALLEST NUMBER OF NODES = 728
LARGEST HOLE = 113
SMALLEST HOLE = 113
```

To terminate POOL execution, press Ctrl/Z.

This causes POOL to print summary information and exit. The summary information consists of the following:

1 The largest contiguous amount of node space.

2 The smallest amount of node space.

3 The largest number of available nodes.

4 The smallest number of available nodes.

This summary information is accumulated from the time POOL is initiated to the time it is terminated.

```
    ;(XJDRV) - SHOW COUNTERS
 .WORD. IO.ITI,000,036 ;(UDC) TIMER - INITIALIZE
;
; PRO 300 SERIES BITMAP FUNCTIONS
;
; NOTE: THESE FUNCTIONS ARE FOR DEC USE ONLY AND ARE SUBJECT TO CHANGE
;

 .WORD. IO.RSD,030,014 ; READ SPECIAL DATA
 .WORD. IO.WSD,010,013 ; WRITE SPECIAL DATA
 DEFIN$ SD.TXT,0 ; TEXT DATA TYPE FOR SPECIAL DATA
 DEFIN$ SD.GDS,1 ; GIDIS DATA TYPE FOR SPECIAL DATA

;
; PROFESSIONAL 300 BISYNC DRIVER (XJDRV) FUNCTIONS
;
 .WORD. SB.PRT,020,003 ; ATTACH AS A PRINTER
 .WORD. SB.CLR,010,036 ; CLEAR COUNTERS (IO.CNT SUBFUNCTION)
 .WORD. SB.RDY,010,033 ; SET DEVICE STATE READY (IO.STA SUBFUNC)
 .WORD. SB.NRD,020,033 ; SET DEVICE STATE NOT READY

 .WORD. IO.LBK,000,035 ; PERFORM LOOPBACK TEST

 .WORD. SB.CBL,010,035 ; PERFORM CABLE LOOPBACK TEST
 .WORD. SB.CLK,020,035 ; DEVICE PERFORMS LINE CLOCKING
;
; COMMUNICATIONS FUNCTIONS
;

 .WORD. IO.CPR,010,033 ;CONNECT NO TIMEOUTS
 .WORD. IO.CAS,020,033 ;CONNECT WITH AST
 .WORD. IO.CRJ,040,033 ;CONNECT REJECT
 .WORD. IO.CBO,110,033 ;BOOT CONNECT
 .WORD. IO.CTR,210,033 ;TRANSPARENT CONNECT
 .WORD. IO.GNI,010,035 ;GET NODE INFORMATION
 .WORD. IO.GLI,020,035 ;GET LINK INFORMATION
 .WORD. IO.GLC,030,035 ;GET LINK INFO CLEAR COUNTERS
 .WORD. IO.GRI,040,035 ;GET REMOTE NODE INFORMATION
 .WORD. IO.GRC,050,035 ;GET REMOTE NODE ERROR COUNTS
 .WORD. IO.GRN,060,035 ;GET REMOTE NODE NAME
 .WORD. IO.CSM,070,035 ;CHANGE SOLO MODE
 .WORD. IO.CIN,100,035 ;CHANGE CONNECTION INHIBIT
 .WORD. IO.SPW,110,035 ;SPECIFY NETWORK PASSWORD
 .WORD. IO.CPW,120,035 ;CHECK NETWORK PASSWORD.
 .WORD. IO.NLB,130,035 ;NSP LOOPBACK
 .WORD. IO.DLB,140,035 ;DDCMP LOOPBACK

;
; ICS/ICR I/O FUNCTIONS
;
 .WORD. IO.CTY,000,007 ;CONNECT TO TERMINAL INTERRUPTS
 .WORD. IO.DTY,000,015 ;DISCONNECT FROM TERMINAL INTERRUPTS
 .WORD. IO.LDI,000,016 ;LINK TO DIGITAL INTERRUPTS
 .WORD. IO.UDI,010,023 ;UNLINK FROM DIGITAL INTERRUPTS
 .WORD. IO.LTI,000,017 ;LINK TO COUNTER MODULE INTERRUPTS
 .WORD. IO.UTI,020,023 ;UNLINK FROM COUNTER MODULE INTERRUPTS
 .WORD. IO.LTY,000,020 ;LINK TO REMOTE TERMINAL INTERRUPTS
 .WORD. IO.UTY,030,023 ;UNLINK FROM REMOTE TERMINAL INTERRUPTS
 .WORD. IO.LKE,000,024 ;LINK TO ERROR INTERRUPTS
 .WORD. IO.UER,040,023 ;UNLINK FROM ERROR INTERRUPTS
 .WORD. IO.NLK,000,023 ;UNLINK FROM ALL INTERRUPTS
 .WORD. IO.ONL,000,037 ;UNIT ONLINE
 .WORD. IO.FLN,000,025 ;UNIT OFFLINE
 .WORD. IO.RAD,000,021 ;READ ACTIVATING DATA
```

```
;
; IP11 I/O FUNCTIONS
;
 .WORD. IO.MAO,010,007 ;MULTIPLE ANALOG OUTPUTS
 .WORD. IO.LEI,010,017 ;LINK EVENT FLAGS TO INTERRUPT
 .WORD. IO.RDD,010,020 ;READ DIGITAL DATA
 .WORD. IO.RMT,020,020 ;READ MAPPING TABLE
 .WORD. IO.LSI,000,022 ;LINK TO DSI INTERRUPTS
 .WORD. IO.UEI,050,023 ;UNLINK EVENT FLAGS
 .WORD. IO.USI,060,023 ;UNLINK FROM DSI INTERRUPTS
 .WORD. IO.CSI,000,026 ;CONNECT TO DSI INTERRUPTS
 .WORD. IO.DSI,000,027 ;DISCONNECT FROM DSI INTERRUPTS
 .WORD. IO.RAM,000,032 ;READ ANALOG MAPPING TABLES

;
; PCL11 I/O FUNCTIONS
;

 .WORD. IO.ATX,000,001 ;ATTEMPT TRANSMISSION
 .WORD. IO.ATF,000,002 ;ACCEPT TRANSFER
 .WORD. IO.CRX,000,031 ;CONNECT FOR RECEPTION
 .WORD. IO.DRX,000,032 ;DISCONNECT FROM RECEPTION
 .WORD. IO.RTF,000,033 ;REJECT TRANSFER

 .MACRO SPCIO$ A
 .ENDM SPCIO$
 .ENDM SPCIO$
;
; DEFINE THE I/O CODES FOR USER-MODE DIAGNOSITCS.  ALL DIAGNOSTIC
; FUNCTION ARE IMPLEMENTED AS A SUBFUNCTION OF I/O CODE 10 (OCTAL).
;

 .MACRO UMDIO$ $$$GBL
 .MCALL .WORD.,DEFIN$
 .IF IDN <$$$GBL>,<DEF$G>
...GBL=1
 .IFF
...GBL=0
 .ENDC

;
; DEFINE THE GENERAL USER-MODE I/O QUALIFIER BIT.
;

 .WORD. IQ.UMD,004,000 ;USER MODE DIAGNOSTIC REQUEST

;
; DEFINE USER-MODE DIAGNOSTIC FUNCTIONS.
;
```

```
 .WORD. IO.HMS,000,010 ;(DISK) HOME SEEK OR RECALIBRATE
 .WORD. IO.BLS,010,010 ;(DISK) BLOCK SEEK
 .WORD. IO.OFF,020,010 ;(DISK) OFFSET POSITION
 .WORD. IO.RDH,030,010 ;(DISK) READ DISK HEADER
 .WORD. IO.WDH,040,010 ;(DISK) WRITE DISK HEADER
 .WORD. IO.WCK,050,010 ;(DISK) WRITECHECK (NON-TRANSFER)
 .WORD. IO.RNF,060,010 ;(DECTAPE) READ BLOCK NUMBER FORWARD
 .WORD. IO.RNR,070,010 ;(DECTAPE) READ BLOCK NUMBER REVERSE
 .WORD. IO.LPC,100,010 ;(MAGTAPE) READ LONGITUDINAL PARITY CHAR
 .WORD. IO.RTD,120,010 ;(DISK) READ TRACK DESCRIPTOR
 .WORD. IO.WTD,130,010 ;(DISK) WRITE TRACK DESCRIPTOR
 .WORD. IO.TDD,140,010 ;(DISK) WRITE TRACK DESCRIPTOR DISPLACED
 .WORD. IO.DGN,150,010 ;DIAGNOSE MICRO PROCESSOR FIRMWARE
 .WORD. IO.WPD,160,010 ;(DISK) WRITE PHYSICAL BLOCK
 .WORD. IO.RPD,170,010 ;(DISK) READ PHYSICAL BLOCK
 .WORD. IO.CER,200,010 ;(DISK) READ CE BLOCK
 .WORD. IO.CEW,210,010 ;(DISK) WRITE CE BLOCK

;
; MACRO REDEFINITION TO NULL
;

 .MACRO UMDIO$ A
 .ENDM

 .ENDM UMDIO$
;
; HANDLER ERROR CODES RETURNED IN I/O STATUS BLOCK ARE DEFINED THROUGH THIS
; MACRO WHICH THEN CONDITIONALLY INVOKES THE MESSAGE GENERATING MACRO
; FOR THE QIOSYM.MSG FILE
;
 .MACRO .IOER. SYM,LO,MSG
 DEFIN$ SYM,LO
 .IF GT,$$MSG
 .MCALL .IOMG.
 .IOMG. SYM,LO,<MSG>
 .ENDC
 .ENDM .IOER.
;
; I/O ERROR CODES ARE DEFINED THOUGH THIS MACRO WHICH THEN INVOKES THE
; ERROR MESSAGE GENERATING MACRO, ERROR CODES -129 THROUGH -256
; ARE USED IN THE QIOSYM.MSG FILE
;
 .MACRO .QIOE. SYM,LO,MSG
 DEFIN$ SYM,LO
 .IF GT,$$MSG
 .MCALL .IOMG.
 .IOMG. SYM,<LO-128.>,<MSG>
 .ENDC
 .ENDM .QIOE.
;
; CONDITIONALLY GENERATE DATA FOR WRITING A MESSAGE FILE
;
 .MACRO .IOMG. SYM,LO,MSG
 .WORD -^O<LO>
 .ENABL LC
 .ASCIZ ^MSG^
 .DSABL LC
 .EVEN
 .IIF LT,^O<$$$MAX+<LO>>,$$$MAX=-^O<LO>
 .ENDM .IOMG.
;
; DEFINE THE SYMBOL SYM WHERE LO IS IS THE LOW ORDER BYTE, HI IS THE HIGH BYTE
;
```

```
.MACRO .WORD. SYM,LO,HI
DEFIN$ SYM,<HI*400+LO>
.ENDM .WORD.

.DSABL LC
```

E Node Pool Status Program

The Node Pool Status Program (POOL) provides the privileged user with statistics about node pool usage, and prints these statistics on the terminal. Once the POOL program is initiated, it executes every two seconds and prints the number of nodes available (unused) in the pool along with the largest contiguous amount of node space. All numbers (decimal) are in terms of 8-word nodes.

Type the following command to cause POOL to execute:

```
MCR>RUN POOL ESC
```

The following example shows sample POOL execution and termination.

```
MCR>RUN POOL ESC
NODES = 745    LARGEST  HOLE = 113
NODES = 759    LARGEST  HOLE = 113
NODES = 751    LARGEST  HOLE = 113
NODES = 753    LARGEST  HOLE = 113
NODES = 746    LARGEST  HOLE = 113
NODES = 742    LARGEST  HOLE = 113
Ctrl/Z
NODES = 728  LARGEST  HOLE = 113
LARGEST NUMBER OF NODES = 759
SMALLEST NUMBER OF NODES = 728
LARGEST HOLE = 113
SMALLEST HOLE = 113
```

To terminate POOL execution, press Ctrl/Z.

This causes POOL to print summary information and exit. The summary information consists of the following:

1 The largest contiguous amount of node space.

2 The smallest amount of node space.

3 The largest number of available nodes.

4 The smallest number of available nodes.

This summary information is accumulated from the time POOL is initiated to the time it is terminated.

Index

IAS
MCR User's Guide
AA-H006C-TC

Reader's Comments

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

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

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
- □ Other (please specify)________________

Name________________ Date________

Organization________________

Street________________

City________________ State________ Zip Code________
or Country

Do Not Tear - Fold Here and Tape

digital™

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO 33 MAYNARD MASS

POSTAGE WILL BE PAID BY ADDRESSEE

IAS Engineering/Documentation
Digital Equipment Corporation
5 Wentworth Drive GSF/L20
Hudson, NH 03051-4929

Do Not Tear - Fold Here