IAS System Management Guide

Order Number: AA–2520F–TC

This manual describes fundamental system concepts and explains how the system can be established and dynamically controlled.

Software 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.

Copyright ©1990 by Digital Equipment Corporation

All Rights Reserved.
Printed in U.S.A.

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.

The following are trademarks of Digital Equipment Corporation:

DDIF	IAS	VAX C
DEC	MASSBUS	VAXcluster
DEC/CMS	PDP	VAXstation
DEC/MMS	PDT	VMS
DECnet	RSTS	VR150/160
DECUS	RSX	VT
DECwindows	ULTRIX	
DECwrite	UNIBUS	digital™
DIBOL	VAX	

This document was prepared using VAX DOCUMENT, Version 1.2

Contents

Part II ERROR HANDLING

Part III COMMAND SPECIFICATIONS

TABLES

Preface

Purpose of the Manual

The *IAS System Management Guide* is designed for the system manager or system operator. In conjunction with both the *IAS Installation and System Generation Guide* and the *IAS Performance and Tuning Guide*, this manual describes fundamental system concepts and explains how the system can be established and dynamically controlled.

The reader must have an understanding of the IAS system. In particular the following documents should be considered prerequisites:

- *IAS System Release Notes*
- *IAS PDS User's Guide*
- *IAS MCR User's Guide*
- *IAS Executive Facilities Reference Manual*

Document Structure

This manual is divided into three parts:

1 Fundamental system concepts.

This section describes the following information:

- How to execute tasks.
- How to set up the system parameters.
- How to operate the system.
- How to manage devices, volumes, and files.
- How to authorize and control user privileges.

2 Brief summary of PDS and SCI commands (and the corresponding MCR commands).

This section provides a detailed description of each SCI command.

NOTE: PDS commands are described in the *IAS PDS User's Guide* and MCR commands are described in the *IAS MCR User's Guide*.

3 Detailed description of the following information:

- Core dump analyzer
- Error logging
- Memory parity support

Each chapter contains the following information:

1 Tutorial

- Chapter 1:
  - System management introduction
  - CLI task concepts
  - User interface concepts
  - Types of systems that can be generated
- Chapter 2:
  - Timesharing
  - Real-time
  - CLIs
  - TCP
  - Devices and volumes
  - Device handlers
  - ACPs
- Chapter 3:
  - Timesharing and real-time system requirements for running tasks
  - Privileges that can be assigned to tasks
- Chapter 4:
  - Controlling and modifying a system to suit a particular installation
  - System parameter description
  - Introduction to the system startup process
- Chapter 5:
  - System startup and initialization
  - Runtime system control
  - Peripheral device handling
  - System recovery
  - File preservation
  - Batch processing control
- Chapter 6:
  - Device, volume, and file management
  - Spooling concepts
  - SGA concepts
- Chapter 7:
  - PDS user authorization and privilege

  - PDS accounting
  - User's command

2 Error Handling

- Chapter 8:
  - Details of the core dump analyzer (CDA)
  - Description of the function of each switch
- Chapter 9:
  - Logging of memory
  - Logging of device errors
  - Description of the function of each switch for the input file specification
- Chapter 10—Description of memory parity support.
- Chapter 11—Description of shadow recording.
- Command Specifications
  - Chapter 12:
    - Brief introduction to each PDS and SCI command
    - Definition of the privilege required to issue each command
    - Detailed description of each SCI command, in alphabetical order
- Appendix A—Illustration of the user profile file (UPF).
- Appendix B—Sample CDA listings.
- Appendix C—Sample error logging output listings.
- Appendix D—Illustration of PDP-11/70 parity error response.
- Appendix E—Description of the I/O exerciser.

Associated Documents

The following manuals are referred to in this manual:

- *IAS Device Handlers Reference Manual*
- *IAS Executive Facilities Reference Manual*
- *IAS Guide to Writing Command Language Interpreters*
- *IAS Guide to Writing a Device Handler Task*
- *IAS MCR User's Guide*
- *IAS PDS User's Guide*
- *IAS Performance and Tuning Guide*
- *IAS System Directives Reference Manual*
- *IAS Installation and System Generation Guide*
- *IAS Task Builder Reference Manual*

- *IAS Utilities Manual*
- *IAS I/O Operations Reference Manual*
- *IAS MACRO-11 Reference Manual*
- *PDP-11 Peripherals Handbook*
- *PDP-11 Processor Handbook*

Refer to the *IAS Master Index and Documentation Directory* for a description of all the available documents associated with IAS.

Documentation Conventions

Throughout this manual, the text enclosed in a box indicates information applicable only to timesharing systems.

Part I Tutorial

1 Introduction

IAS is a multifunction operating system designed for the PDP-11 family of computers. This manual describes how the system manager can regulate the system to suit local installation requirements. In conjunction with the *IAS Installation and System Generation Guide* and the *IAS Performance and Tuning Guide,* this manual also describes how to start up and operate the system.

1.1 Types of IAS Systems

You can generate IAS to provide three different types of systems:

- Timesharing

A timesharing system provides all the facilities for a program development environment with many simultaneous users. This system supports real-time tasks and batch processing. It includes the IAS Scheduler to control tasks and provide optimum service to all users, and the Timesharing Control Primitives (TCP). TCP provides a number of protection and privilege control features typically required in a timesharing environment. For example, it enables users to protect mountable magnetic media against other system users. TCP also provides a method to enable timesharing tasks to invoke and communicate with other timesharing tasks. The timesharing control services (TCS) provide a task level interface to TCP. For further details on TCS, see the *IAS Guide to Writing Command Language Interpreters.*

NOTE: Throughout this manual, boxed text indicates those areas applicable only to timesharing systems.

- Multiuser

  A multiuser system is used in a mixed real-time and program development environment with multiple users. Program development activities run under the control of the IAS scheduler. A multiuser system does not include the timesharing control primitives (TCP) and therefore does not offer the protection facilities of a timesharing system. This is the major difference between a multiuser and a timesharing system.

- Real-time

  A real-time system is used in a real-time environment and will support a small number of terminals on which there should be little or no program development activity. A real-time system includes neither the IAS scheduler nor the timesharing control primitives (TCP) and therefore is not suitable for multiple users.

You must decide during system generation which type of system best suits your installation. Changing to another type of system entails repeating the system generation process. For details of how to specify the required type of system, see the *IAS Installation and System Generation Guide.*

1.2 User/System Interfaces

1.2.1 Digital-Supplied Interfaces

The user/system interface on IAS consists of tasks called command language interpreters (CLI).

IAS supplies two standard interfaces:

- Digital Command Language (DCL)

  DCL consists of English language commands. DCL is implemented on IAS by the program development system (PDS or, on a multiuser system, PDX).

- Monitor Console Routine (MCR)

  MCR is not as easy to use as DCL for the beginner, but MCR is more flexible for the experienced user.

> On a timesharing system, DCL (PDS) is the only available user interface. However, PDS supports MCR Mode, which simulates the MCR interface while running under PDS. See the *IAS MCR User's Guide* for further details on MCR mode.

On a multiuser system, you can specify either DCL or MCR as the default user interface. You can override the default user interface for an individual terminal if necessary. If you specify DCL, it will be implemented by a special version of PDS called PDX. PDX is the same as PDS at the user level.

On a real-time system, you can specify either DCL or MCR as you do on a multiuser system. However, to avoid space constraints it is recommended that you use MCR on a real-time system.

1.2.2 Console Interface

On IAS, in response to the hardware bootstrap, the console terminal displays the following message:

```
nnnk (word) IAS mm.m
```

You cannot redirect the console to any other terminal.

If DCL is the user interface, the PDS version that runs for the console terminal has extra commands available for the console user (system manager or operator). This PDS version prompts SCI> (instead of PDS>). Throughout the documentation set, this system manager interface is referred to as the *System Control Interface (SCI).*

1.2.3 User-Written Interfaces

On a timesharing system, you can write your own CLIs to satisfy the needs of your installation (for example, to implement a specific application). You can write CLIs in any programming language. For the CLI to use the Timesharing Control Services (TCS), either of the following requirements must be met:

- The language must be MACRO, or
- The language must have the ability to call MACRO subroutines (see the *IAS Guide to Writing Command Language Interpreters*).

All CLI tasks run as timesharing tasks and are serviced in identical fashion to the other timesharing tasks in the system.

Using the console interface, you can designate a task as the CLI subsystem to service one or more specified terminals. For example, you might set aside one terminal for an application task required by your installation. If you designate this task as the CLI subsystem to service that terminal, it is automatically started when someone presses a CTRL/C. At that point, a user at that terminal cannot input commands to anything except the applications task.

2 Fundamental System Concepts

2.1 The IAS Scheduler

The *IAS scheduler* is supported only on timesharing or multiuser systems.

The prime objective of the IAS scheduler is to reduce as much as possible the average response time to all user demands. The IAS scheduler controls tasks running in a timesharing partition at a priority equal to or lower than the scheduler priority. tasks run under the control of the scheduler are called *scheduler-controlled tasks*.

On a timesharing system, this includes all tasks that run under the control of TCP (that is, all *timesharing tasks*).

The scheduler executes in the system when tasks of higher priority (that is, real-time tasks or other system components) do not require processor time. The scheduling mechanism is governed by a predetermined scheduling algorithm that automatically allocates resources to tasks under its control, depending on CPU usage and mode of processing (interactive or batch). The scheduler keeps a number of round-robin queues, or levels, of tasks to be scheduled and maintained.

Scheduling of tasks is described more fully in Chapter 4.

2.1.1 Swapping

Swapping is the writing of a task to disk, so as to leave space in memory for another task to execute. Only scheduler-controlled tasks are swapped. If insufficient contiguous space exists to load such a task when required, the system first moves resident tasks within memory to create more contiguous space. This process is known as *shuffling*. If insufficient space still exists, the system writes one or more tasks to the swap space on disk.

The swap space consists of files on one or more physical volumes that can coexist with normal user and system files, if required.

All tasks under IAS scheduler control can be swapped.

2.2 Execution of Real-Time Tasks

Real-time and certain system tasks run on a priority basis. space and time resources are allocated to real-time tasks on demand according to their priorities. each task is allocated a *run priority* that can range from 1 to 250 (priority 250 is the highest). High-priority tasks are executed in preference to low-priority tasks. Tasks are selected to run in strict priority order from a single real-time scheduling queue.

Several system components (notably device handlers) run as high priority, real-time tasks. When necessary, a critical real-time task with a higher priority than such system components can execute before these system components.

Use the following MCR or PDS commands to set the priority of a real-time task:

- When the task is activated—MCR or PDS RUN command.

- When the task is installed—MCR INS or PDS INSTALL command.
- When the task is task built—PRI option to MCR TKB or PDS LINK.

You can also set the priority of a real-time task using various system directives. (See the *IAS System Directives Reference Manual.*)

To change a real-time task priority while it is running, use the MCR ALT or PDS SET PRIORITY command, or the ALTP$ system directive.

Control of running real-time tasks is further described in the *IAS Executive Facilities Reference Manual.*

2.2.1 Checkpointing

Checkpointing is a mechanism to aid real-time tasks to obtain memory to execute as soon as possible after they are requested. If sufficient memory is already available when a task is requested, the task is loaded. If memory is not available, the system checkpoints out of memory, executing task(s) of lower priority than the requesting task to make the desired space available. The system can only checkpoint tasks that have the attribute of being checkpointable. You specify whether or not a task is checkpointable at task build time, using the /CHECKPOINT qualifier to the PDS LINK command or the /CP switch to MCR TKB. The default is that the task is checkpointable.

Checkpointing real-time tasks differs from swapping scheduler-controlled tasks (see Section 2.1.1) in that checkpointing is controlled by the priority of the real-time task. Swapping maximizes the use of the processor and memory, and provides a fair service to all tasks under IAS scheduler control. If you build a real-time task non-checkpointable, but it runs under IAS scheduler control (see Section 2.1), the IAS scheduler ignores the fact that it is non-checkpointable and can swap the task if necessary.

Checkpointing is further described in the *IAS Executive Facilities Reference Manual.*

2.3 Command Language Interpreters

A *command language interpreter (CLI)* is a task that interprets and services the commands entered by users at terminals. A (system-dependent) number of CLIs can be installed in the IAS system and each terminal in the system can be allocated to be serviced by one of these installed CLIs. All CLI tasks run as normal tasks and are serviced concurrently with the other tasks in the system.

In addition to the CLIs supplied with IAS (see Section 1.2), other command language interpreters can be written specifically for the requirements of a particular installation. See the *IAS Guide To Writing Command Language Interpreters* for further details.

2.4 Timesharing Control Primitives

The *timesharing control primitives (TCPs)* serve three major functions in the IAS system:

1. They provide an external interface through which executing timesharing tasks can use the timesharing facilities provided by the system.
2. They provide internal service and control functions for the timesharing components of the system.
3. They regulate access to system facilities and resources, according to user privileges.

You can give any task the privilege to use the TCP facilities at task initiation time. When a CLI is installed, it is granted this privilege by the /PRIV qualifier to the INSTALL/CLI command (see Chapter 12). Allocation of privilege to PDS users is described in Chapter 7.

The range of facilities available to a TCP-privileged task depends on the task type. PDS has access to all TCP facilities, although some facilities can only be requested by the version of PDS running on the console (SCI). User-written CLIs have access to a subset of the TCP facilities, and timesharing tasks have access to a further subset of those facilities available to user-written CLIs.

2.5 Devices and Volumes

IAS supports two types of devices:

1. Devices that are potentially shareable (for example, disk or DECtape).
2. Devices that are normally non-shareable (for example, card reader, line printer, teletypewriter).

A non-shareable device must be allocated to a user before that user can access it. Similarly, the user must mount a volume on a mountable device before I/O operations can be performed on the device for that user.

So that users do not have to wait for a serial processing, non-shareable device to be free, you can make such devices *spooled*. For example, several users might want to use the line printer concurrently. In this event, output directed to the line printer is automatically written to disk files, and any requests to output the files are queued. The queued requests are serviced in priority order, then first-in first-out. The files are output to the specified device as soon as possible after the request for output is made (or after the time specified for time-tagged entries). This process is called *output spooling*. IAS also supports *input spooling* for card readers that operate as batch job input devices.

All volumes must be mounted before they can be accessed. Three types of volumes are available:

1. Disk and DECtape volumes
2. Magnetic tape volumes
3. Foreign volumes

Files-11 magnetic tape volumes and foreign volumes are non-shareable only.

On a timesharing system, you can exclusively allocate a mountable device. When you mount a volume on a timesharing system, you must specify a volume identifier.

Once you have mounted a Files-11 disk or DECtape volume, you can access it only if the access protection permits. Access protection exists at three levels:

1 Volume protection
2 Directory protection
3 File protection

You are responsible for ensuring that the appropriate protection is applied at each level.

On a timesharing system, each user who wants to access a mountable volume must issue the appropriate mount command.

A distinction is made between the first user to mount a disk (or DECtape) and subsequent users who mount the same volume, in that only the first user can specify certain command qualifications. Note that only one user can mount a magnetic tape or a foreign volume.

If a Files-11 disk or DECtape volume is already mounted for at least one user, subsequent users who mount the same volume are granted access to it. Note that whether data can be transferred to or from the volume depends on the protection on the volume as well as access to the device.

Exclusive allocation of a device prevents any other user from mounting a volume on that device. If a user wants to mount several volumes consecutively on the same device (most likely in batch), the user would perform the following steps:

1 Allocate a drive (explicitly).
2 Mount and use the first volume.
3 Dismount the first volume keeping the drive.
4 Mount and use the second volume.
5 Dismount the second volume and deallocate the drive.

Without TCP, neither is it possible for you to allocate a device for exclusive use, nor is it necessary for each user to mount a volume before being able to use that volume. Once a volume has been mounted for the first time, any user can access the volume. Exclusive allocation of a device is provided for individual tasks by the *attach* mechanism, described in the *IAS Device Handlers Reference Manual*.

2.5.1 System Pseudo-Devices

The system employs *pseudo-devices* for the following reasons:

- To provide physical device independence for user tasks and system tasks.

- To provide a convenient means of communication between system and user tasks. Pseudo-devices, as their name implies, exist as a software mechanism only and do not have a physical form. Two main classes of pseudo-devices exist:
  - Those that represent physical devices. Requests directed to these are forwarded to the corresponding physical device. For some pseudo-devices, the correspondence to a particular physical device is fixed by the system manager. For others, it is user-dependent.
  - Those that provide a means for communicating with system tasks. This can be communication between system tasks or communication between user tasks and system tasks. These pseudo-devices do not represent physical devices.

IAS also provides the pseudo-device NL (null device), which does not fit into either class. To the software, the null device represents a physical device, although no actual hardware exists. You use NL mainly for testing purposes. For example, instead of specifying an actual device in a program, you specify NL. This allows you to test the rest of your program without actually writing to a physical device. For more information on NL, see the *IAS Device Handlers Reference Manual.*

The following pseudo-devices are in the first category described above:

1 Console input (CI)

   Use this pseudo-device for input from the system operator console. It corresponds to the same physical device as console output (CO).

2 Console output (CO)

   Use this pseudo-device to communicate with the system operator. It is associated with the system operator console. Any task can output to CO.

3 Default listing device (CL)

   This pseudo-device is usually associated with the system line printer.

4 System library device (LB)

   Use this pseudo-device for all system library files (for example, the default object module library, SYSLIB.OLB). It is usually associated with the bootstrapped system disk (that is, the disk from which the system was bootstrapped).

5 Spooler temporary files device (SP)

   All temporary files created for input and output spooling are directed to this pseudo-device. It can be associated with any disk having a suitably large amount of free space, but is normally the bootstrapped system disk. This pseudo-device is also used for all temporary files that do not require rapid random access.

6 User default device/system disk (SY)

   Use this pseudo-device for two different purposes:

- First:

On real-time and multiuser systems, SY is the disk from which the system was bootstrapped. To redirect the system disk on a multiuser or real-time system, see Section 7.2.4.

> On a timesharing system, SY is defined as the user default device, specified in the user profile file (see Section 7.2). You can change your default device by means of the PDS SET DEFAULT command (see the *IAS PDS User's Guide*).
>
> SY is used for input and output for timesharing tasks. Requests to SY are directed to the user default device.

- Second:

To refer to the bootstrapped system disk.

When SY is not used for input or output from timesharing tasks (that is, when requests to SY are used in a different context), the requests are directed to the disk from which the system was bootstrapped.

7 Terminal input (TI)

This pseudo-device is used for input from the user terminal (either interactive or batch). It is associated with an actual terminal, or for a batch job, with one of the batch pseudo-devices.

8 Terminal output (TO)

This pseudo-device is used for output to the user terminal. It is associated with the same physical device as TI.

9 Workfile device (WK)

This pseudo-device is used for temporary files to which rapid random access is required (for example, task builder work files). It is normally associated with the system disk, but might be associated with a rapid-access, fixed-head disk.

The following pseudo-devices are in the second category described above:

1 Message output (MO)

This pseudo-device is associated with the message output device handler. The handler is used by system tasks to expand error messages they produce into a more readable form. The *IAS Device Handlers Reference Manual* describes the use of MO.

2 Timesharing control primitives interface (PI).

> The PI pseudo-device is used for communication with the timesharing control primitives task (TCP).

3 Batch input/output (BA).

Each batch stream in the system (up to a maximum of eight) requires its own pseudo-device. These are called BA0, BA1 ... BA7. All batch stream input is obtained from these pseudo-devices and all batch log output is directed to them.

The following device does not fit into either category:

- Null device (NL).

  Input can be directed to the null device, instead of a physical device, for testing purposes.

NOTE: Optional software such as DECNET or DBMS might require other pseudo-devices. See the appropriate manuals for further details.

2.5.2 PDS User Logical Device Names

PDS users can associate *logical names* with physical devices, independently of the pseudo-device names. This is particularly useful for a batch user who might be unaware of the physical unit on which a volume is to be mounted. See the *IAS PDS User's Guide* for further details about logical device names.

2.6 Device Handlers

In IAS, support for PDP-11 devices is provided by device handlers. Device handler tasks perform the necessary functions that enable the physical I/O operations to be performed. A handler must be resident before a device can be accessed.

Handler tasks can be written and included in the IAS system to service non-standard devices. See the *IAS Guide To Writing a Device Handler Task*. Device handlers can be dynamically loaded into memory; infrequently used handlers must be resident only when required and can be removed after use.

NOTE: Loading handlers only when required might cause fragmentation of the partition. If a handler that connects to interrupts is loaded in the timesharing partition, the handler cannot be shuffled. This applies to all supplied handlers with the exception of the system pseudo-device handlers PI,MO,NL, and BA.

See the *IAS Device Handlers Reference Manual* for a detailed description of handlers.

2.7 Ancillary Control Processors

All file processing in IAS is channelled through tasks called ancillary control processors (ACPs). Basic operations, such as read and write, are directly controlled by the device handler. However, all file processing functions, such as create, delete, and file access, are performed by an ACP task associated with that device. An ACP is associated with a volume when that volume is mounted as a FILES-11 device.. Not all volumes have ACPs associated with them.

IAS supplies three ACPs:

1 F11ACP—For Files-11 volumes (disk and DECtape).

2 MTAACP—For ANSI magnetic tape handling.

3 DTAACP—For installations that require significant amounts of DECtape processing.

F11ACP and MTAACP are default ACPs for Files-11 disk volumes and magnetic tape, respectively. DTAACP can be used for DECtape instead of F11ACP, if so specified during system generation or when a DECtape volume is mounted. It is advisable to use DTAACP for DECtapes, because all processing in an ACP is serial. If F11ACP is used for processing DECtapes, it can reduce the file processing throughput for other Files-11 devices.

Only a single copy of each ACP can be run on one volume, because file functions on a given unit must be executed serially to prevent interference by other users. Concurrent execution of several file functions on one unit will destroy the data held on the volume. However, a different ACP can be assigned to handle file accesses for different device units. In large systems with a heavy processing load, you can install multiple ACPs to achieve some parallelism in file processing. See the *IAS Performance and Tuning Guide* for details.

You can specify a user-written ACP instead of a standard IAS ACP. Also, you can use a user-written ACP to perform special functions on a volume that is not associated with a standard ACP. You specify ACPs at system generation (to name the defaults) or when a volume is mounted on a device (to override temporarily the defaults). You can associate only one ACP at a time with each volume.

3 Task Execution

3.1 Introduction

IAS schedules tasks to execute in defined areas of memory called partitions. A partition is a contiguous area of real memory. It can contain both executing tasks, and tasks that are permanently resident in memory (fixed) regardless of whether or not they are executing.

You specify the name, base address, size, and type of each partition at system generation. You can also specify a default partition; real-time tasks for which no explicit partition is declared will execute in this default partition. If you do not specify a default partition, the default partition is called GEN. On timesharing and multiuser systems this is a T (timesharing) type partition, and is also the default partition for scheduler-controlled tasks. On real-time systems, GEN is an S (system-controlled) type partition.

Unless a task is fixed in memory, it is automatically removed from memory upon termination.

3.2 Timesharing and Multiuser System Requirements

Whether or not a scheduler-controlled task can run at a particular time depends on the following criteria:

1 Maximum scheduler-controlled tasks:

The maximum number of scheduler-controlled tasks that can run concurrently.

2 The maximum scheduler-controlled task size.

3 The size of swap space allocated.

These values are initially specified during system generation.

3.2.1 Memory Requirements

All scheduler-controlled tasks run in a single timesharing partition.

More than one timesharing (T) type partition can be defined at system generation, but only one partition is actually used for execution of scheduler-controlled tasks when the system is active. On a timesharing system, you specify this partition at system startup using the SET PARTITION timesharing start-up command.

On a multiuser system, you specify this partition when you enable the IAS scheduler, using the MCR UTL command as follows:

```
MCR>UTL /EN[:par]
```

If you do not specify this partition, on both timesharing and multiuser systems, the default is GEN.

This partition is primarily used for the execution of scheduler-controlled tasks, but real-time tasks can also execute in the partition. Scheduler-controlled tasks are swapped out of memory in favor of higher priority real-time tasks.

In a timesharing partition, all tasks can be shuffled (including fixed tasks), that is, they are moved in order to create contiguous areas of free memory. If an area of free space cannot be found, resident tasks and shareable global areas are moved towards the bottom of the partition to try to create a free memory area of the required size. A task segment will not be moved, however, if there is active I/O to or from that segment.

3.2.2 Swap Space Requirements

The swap space for scheduler-controlled tasks consists of a number of disk files that are used to record temporarily the task images of scheduler-controlled tasks not currently resident in memory. The size of this swap space limits the number of scheduler-controlled tasks that can execute in the system at any one time. A scheduler-controlled task is not run unless sufficient swap space exists to record the task. Therefore, the swap space must be of sufficient size, such that space can always be allocated for tasks or task segments that need to be loaded. A task is never split between two swap files; if it cannot fit into one swap file, it is placed in another swap file.

Swap space is also used for checkpointing real-time tasks, so make allowance for this.

Swap space is normally allocated during system generation, but you can redefine it either when you start up the system or during system operation, using the SCI command CREATE/SWAPFILE or the MCR command SWA.

3.2.3 Scheduling Parameters

Digital supplies default values for the parameters that control the scheduling of tasks running under control of the IAS scheduler. See the *IAS Performance and Tuning Guide* for details about these default values. You can change these parameters to suit the needs of your installation.

> On a timesharing system, specify the SET BATCH and SET QUANTUM timesharing start-up commands (see the *IAS Installation and System Generation Guide*).

Scheduling parameters can also be changed dynamically with the MCR UTL command (see the *IAS MCR User's Guide* or the SCI SET command detailed in Chapter 12).

The tasks to be scheduled are divided into a number of scheduling levels. The default number of levels is four on a timesharing system with an exclusive batch level and three on multiuser systems, but you can change the numbers as described above. The way in which the IAS scheduler uses these levels to optimize system performance is described in Chapter 4.

For guidance on tuning the scheduling parameters to suit your installation, see the *IAS Performance and Tuning Guide*.

3.2.4 Real-Time Considerations for Timesharing Users

The response time for a given number of users is affected by the amount of real-time processing and the nature of the real-time activity. Real-time tasks are normally allocated priorities higher than scheduler-controlled tasks. Therefore, if there is much real-time activity, interactive and batch processing can suffer. For example, a real-time program that accesses the system disk 100 times per second could seriously affect the scheduler service. On the other hand, a program that writes a short record on magnetic tape or a nonshared disk once every second should cause very slight interference.

Real-time tasks can be run in the same partition as scheduler-controlled tasks. In this case scheduler-controlled tasks will be shuffled or swapped out to make room for higher priority real-time tasks, while low priority real-time tasks are checkpointed or swapped to make room for scheduler-controlled tasks.

For most purposes, it is sufficient to have a single general-purpose partition (normally called GEN). However, if you need rapid response for some real-time tasks, for example, if they are controlling critical processes, it is advisable to use a separate partition for them. This is because swapping out scheduler-controlled tasks to make room for a real-time task can take several seconds, whereas checkpointing of real-time tasks is much faster.

3.3 Real-Time System Requirements

Real-time tasks can run in any partition. Real-time tasks in the timesharing partition that have a priority less than or equal to the scheduler run under the control of the scheduler.

A system-controlled partition can contain one or more tasks at a time. The operations within this partition are similar to those in timesharing partitions, except that no shuffling of tasks in memory will occur in order to create space for tasks that require memory. In this case, a real-time task cannot be loaded until there is sufficient contiguous memory available within the partition. However, if the task to be loaded is of a higher priority than currently executing checkpointable tasks, it can be loaded into memory if the required contiguous memory area can be made available by checkpointing lower priority tasks. If sufficient memory is already available without checkpointing any tasks out of memory, additional tasks can be loaded into the partition.

A user-controlled partition can contain only one real-time task at a time. Although checkpointing can still occur, running a task in a user-controlled partition prevents memory fragmentation in another partition for a long-resident task, and ensures memory availability for real-time tasks.

3.4 User Requirements

All users require privileges to issue appropriate commands and access devices.

3.4.1 Command Language Interpreters

Timesharing users need a Command Language Interpreter (CLI) to provide access to the system facilities. The user interface supplied with an IAS timesharing system is DCL, implemented by the Program Development System (PDS) that can execute in interactive or batch mode.

Users can write their own CLIs in any programming language that provides the necessary facilities. Any number of CLIs, up to a system dependent maximum, can be installed in the system. The number and type of CLIs installed will depend on the user requirement (for example, a number of application dependent CLIs can well be established for certain groups of users). See the *IAS Guide to Writing Command Language Interpreters* for further details about CLIs.

3.4.2 PDS Timesharing Privileges

If a user is operating the PDS interface supplied with IAS, that user needs to be authorized with command privileges. Command privileges allow a set of commands to be issued that satisfy the individual requirements.

PDS commands are described in detail in the *IAS PDS User's Guide*. For control purposes, these commands are subdivided into groups, so that you can make each group available to, or withold each group from an individual user of the system.

You can make a different set of commands available for use in batch mode from those available for interactive use.

You can also give the user certain task privileges. Task privileges enable the user to run timesharing tasks that use real-time privileged system directives, memory management directives, the timesharing control primitives, or to run auto-installed timesharing tasks that are executive privileged (linked with the /PR option and thus mapped onto areas of the system Executive).

Real-time directive privilege and memory management directive privilege are described in the *IAS System Directives Reference Manual*. Executive privileged tasks are described in the *IAS Executive Facilities Reference Manual* and the *IAS System Directives Reference Manual*. The allocation of user command privileges is described in Chapter 7.

3.4.3 PDS Real-time Privileges

A user who wants to run real-time tasks requires the privileges to issue the PDS commands that enable execution and control of real-time tasks.

Real-time tasks are automatically given access to all system directives including real-time privileged directives, and can be executive privileged. This is independent of the initiating user's timesharing task privileges.

Task privileges as described in Section 3.4.2 have no effect on a multiuser system.

3.4.4 Access to Devices

Certain devices are designated to be available to timesharing tasks, while all devices in the system are potentially accessible to real-time tasks. Thus, appropriate devices must be specified as available for use by timesharing tasks while those devices required for real-time can be withheld from timesharing use.

In order that certain *single-stream* devices can be used by all system users without having to wait for device availability, such devices can be set as spooled devices. This enables all users of the system to access the device while the system spools the input or output automatically.

See Chapter 6 for a description of supported devices.

4 System Control

4.1 Introduction

This chapter describes how the IAS scheduler operates and how you can modify it to suit your installation. The chapter also describes the aspects of the system initially established at system generation that can, in many cases, be modified as follows:

- On a timesharing system:

At timesharing system startup.

- Dynamically:

  Using SCI or MCR commands.

The timesharing startup commands are described in the *IAS Installation and System Generation Guide*.

This chapter also introduces the run time system control functions available at the operator console.

NOTE: The SCI commands are described in Chapter 12, while the MCR commands are described in the *IAS MCR User's Guide*. Table 12–1 lists the SCI commands and their MCR equivalents.

If you need a full description of how to schedule real-time tasks, see the *IAS Executive Facilities Reference Manual*.

You can generate your IAS system to reflect the general needs of your installation, then tailor it to suit the needs of the current session. You can tailor it either at timesharing system start-up (on a timesharing system), or at run time using SCI or MCR commands.

At run time, you can do the following things:

• On a timesharing system, initiate a controlled closedown of timesharing activities.

- Use various commands to control the interactive, batch and real-time activity in the system.
- Control the availability of CLIs and designate terminals from which CLIs are run.
- Control device and volume availability for all system users.
- Abort any abortable system or user task, if necessary. Thus you have ultimate control over tasks in the system.

4.2 System Parameters

You set up the majority of system parameters initially at system generation time. Some of these cannot be modified, but you can change the values of others to suit a particular system session. For example, you can change the swapping device or alter the number of batch streams or timesharing terminals. Generally, in a stable installation where fluctuations in hardware configuration and workload are infrequent, you can set up the system characteristics at system generation and little further involvement is necessary. This section describes those areas of the system that you can modify after system generation at run time, or

during timesharing system start-up.

4.2.1 Scheduling Parameters

You can modify a number of IAS scheduler parameters to suit the needs of your particular installation.

The tasks to be scheduled are divided into a number of *scheduling levels*. Tasks in each level are serviced in *round-robin* fashion in priority order of levels, with level one the highest.

On a timesharing system only, you can choose to have a separate level for batch jobs, called the *batch level*. Batch level is always the lowest, (usually level 4), and all batch run tasks run in this level.

Since tasks in the high levels (Levels 1 and 2, for example) are given short and frequent amounts of time, while tasks in the lower levels (Levels 3 and 4, for example) are given longer time periods less frequently, it follows that running highly interactive tasks in the high levels provides good response times. Additionally, running CPU-bound tasks in the lower levels provides adequate but less continuous service for their throughput.

You cannot always decide in advance the type of execution characteristics a task will exhibit, and these characteristics might change during the execution of the task. For this reason, the system automatically moves tasks within the interactive scheduling levels on the basis of their recent past execution characteristics.

On timesharing systems, no promotion to nor demotion from the batch level occurs.

The mechanism of promotion and demotion between levels is based on the system parameters of *quantum* and *promotion time*. quantum parameters are described in Section 4.2.2.

A task executing at a certain level is demoted to the next lower level if it uses up all its quantum (CPU time) without relinquishing control. In this case, the task tends to be relatively more CPU-bound. The system strives for the opposite occurrence (a CPU-bound task becoming relatively interactive) by using a promotion mechanism. Setting the *promotion time parameter* (number of ticks between scheduler promotions) ensures that tasks at lower levels are not starved of CPU time while waiting for the next schedule. For each level in which no task has been scheduled during the time specified in the promotion time parameter, the scheduler promotes one task to the next higher level.

Certain activities indicate the immediate interactive nature of a task. For example, the scheduler automatically promotes tasks to Level 1 whenever users complete terminal I/O. This ensures that terminal interactive tasks obtain the best possible response from the system.

> Whenever you type CTRL/C on a terminal (timesharing systems only), the scheduler promotes your CLI to level 1.

The greater the number of interactive tasks in the system, the more likely it is that tasks at lower levels may get starved of CPU time.

It is possible to draw the following conclusions from these points:

- The use of more than one scheduling level becomes effective if the execution characteristics of currently executing tasks are different (this is nearly always true in a non-specialist environment, and at least three levels are recommended).
- Quantum values (see Section 4.2.2) should increase with level number for optimum working of the scheduling mechanism.
- Specification of the promotion time parameter ensures that tasks in lower levels are not starved of time.

IAS is distributed with three scheduling levels, but you can optionally define a fourth level during QASGN (see the *IAS Installation and System Generation Guide*), to be used exclusively for batch.

- Level 1—Terminal interactive tasks
- Level 2—I/O-bound tasks
- Level 3—CPU-bound tasks

> - Level 4—Batch tasks (timesharing systems only)

Level 1 is used exclusively for tasks that are performing terminal I/O. A task is always promoted to Level 1 when a terminal read or write is completed or when you type CTRL/C. Tasks are demoted from Level 1 each time they wait for I/O and when they use their full quantum of CPU time. This ensures that terminal users receive the best possible response from the system.

An I/O-bound task is initially demoted to Level 2. If it never uses its full quantum, it remains in this level, where it receives preferential treatment to CPU-bound tasks. Since I/O-bound tasks normally require short bursts of CPU time before waiting again for a transfer operation, this arrangement enables the scheduler to make full use of the speed of I/O devices.

Level 3 is for CPU-bound tasks, which are given all the time left after other tasks have been serviced.

> Level 4 batch tasks are guaranteed a certain amount of processor time. The operation of the batch level is described fully in Section 4.2.3.
>
> **NOTE: The batch level (usually Level 4) is available only on timesharing systems running PDS BATCH. Tasks are neither promoted to nor demoted from the batch level. The batch level is always the lowest level (that is, the level with the highest number).**

4.2.2 Quantum Parameters

A task's *quantum* determines the amount of CPU time a task has before it becomes eligible to be swapped out of main memory. The scheduler allocates each task its quantum in a series of *time slices*. A time slice is the maximum continuous CPU time for which a task can execute before the scheduler services another task.

Each scheduling level also has an associated *time factor value* (t). This value increases with the level number so that tasks at higher levels (Levels 1 and 2), which are scheduled more often, receive shorter amounts of CPU time, while tasks at lower levels receive longer amounts of CPU time when scheduled.

The following formula determines a task quantum (q):

```
q = ast+c
```

where:

- q = Task quantum
- a = *Allocation factor* (number of ticks per memory size)
- s = Task size in 32-word blocks
- t = Time factor associated with each scheduling level.
- c = *Quantum constant* (minimum guaranteed quantum for the system)

NOTE:

- **The allocation factor is used to determine the amount of CPU time allocated to a task. The value is given in ticks per memory size, so that the amount of CPU time allocated increases with task size.**
- **The value of "a" multiplied by "s" is assigned to a task when it begins execution or when it issues an extend-task directive.**
- **"t" increases with level number. In other words, the lower the level, the higher the value of "t." The quantum for a task in a low (high-numbered) scheduling level might be quite large.**

In order not to block other higher priority tasks awaiting service, the scheduler calculates the quantum for the task, then allocates the quantum to the task in a series of time slices. At the end of each time slice, or if the task waits before the time slice is complete, the scheduler performs a reschedule. Although this can cause a higher priority task to be executed, the task is not swapped until its quantum has expired or it relinquishes control (for example, by entering a wait state).

You can specify a maximum time slice. Do not, however, specify a maximum time slice smaller than the maximum quantum for a Level 1 task. To calculate the quantum for a Level 1 task, use the average size for tasks at that level.

You can reset the scheduling parameters at timesharing system startup (on a timesharing system) by using the SET QUANTUM command.

See the *IAS Installation and System Generation Guide* for details on the SET QUANTUM command. You can also reset the parameters dynamically using the SCI commands SET ALLOCATION, SET QUANTUM, and SET SERVICE, or the MCR command UTL. See the *IAS MCR User's Guide* for details on the UTL command.

4.2.3 Batch Scheduling Parameters

The IAS batch facilities enable batch tasks to execute in a background scheduling level, which is the lowest scheduling level. Although tasks initiated at this level timeshare with tasks at higher levels, they are not subject to the promotion and demotion mechanism of the interactive levels. Batch level is supported only for tasks run under PDS batch in a timesharing system. Batch level is not available on a multiuser system.

Two parameters are associated with batch level:

- Batch quantum
- Time between batch schedules

The defaults are those values distributed with the system. See the *IAS Performance and Tuning Guide* for a list of default values for the scheduling parameters. You can alter these defaults at timesharing system start-up by issuing the SET BATCH command. See the *IAS Installation and System Generation Guide* for details on the SET BATCH command. You can also alter the values dynamically using the SCI commands SET QUANTUM/BATCH and SET SERVICE/BATCH (see Chapter 12).

Batch level processing can be used as desired by the local installation. It can be used as a soak for any time not needed for real-time or interactive activities, for tasks whose turn-around requirements are non-critical. In this case, specify the parameter that allocates time between schedules as 0 (zero). Alternatively, you can obtain a minimum level of service as required by specifying the maximum time after which a batch schedule is to occur.

WARNING: Do not set the time between batch schedules to 0 on systems where scheduler-controlled tasks are likely to be CPU-bound for long periods of time. Under these conditions, the scheduler does not run any non-memory-resident tasks while there is a CPU-bound job running and a non-memory-resident job to run. This situation persists until the CPU-bound job becomes non-CPU-bound. This means your batch job might be CPU-starved.

To ensure that batch tasks are guaranteed time, set the batch quantum and the time between batch schedules to non-zero figures.

You can use batch level for classical batch processing for input from slow input devices. In this case, you can specify a CLI that is allocated to service-queued input to run in the batch level. The queued input might have been spooled from a card reader or submitted to the queue from a timesharing terminal. All user tasks initiated by a CLI task running at the batch level are also executed at the batch level and therefore processed in background mode.

A CLI servicing queued input can also run in the normal interactive levels, causing it to receive the same level of service that strictly interactive CLIs would receive.

By limiting the amount of time specified for tasks which are executing in the batch level, the effects of batch level execution on the interactive execution can be controlled and varied to meet any changing day-to-day requirements.

4.2.4 Swapping Parameters

Scheduler-controlled tasks are temporarily *swapped* out of memory when there is not sufficient free memory to load a task which requires to execute. This requirement may be for a new task, or one which has been previously swapped out of memory. When a task needs to be loaded, and sufficient space is not already available, the memory allocation routine first checks to see if enough contiguous space would be created by shuffling task areas in allocated memory. If enough space cannot be created by shuffling, the scheduler selects one or more tasks to be removed from memory to create the required space. The tasks selected to be swapped out of memory are written to one or more swap files. Only the read/write segment of a task and dynamic regions are written to the swap file. Read/write SGAs are written to the file from which they were loaded. Read-only task segments simply have their memory space freed.

The swapping performance of a system is not only dependent on the hardware specifications of the swapping disks (for example, transfer speed, rotational delay, head positioning time) but also on the use of the disks within the system. The significant factors are whether or not the swapping disk has a dedicated controller or units. Consider the following three cases:

Dedicated Controller	Dedicated Units	Example
No	No	Swapping onto system disk
No	Yes	Swapping onto RK disk unit, but other RK disk units used for file storage
Yes	Yes	Dedicated device and units

In the first case, the swapping performance of the system is poor. Executive swap requests to the disk handler can be delayed while you wait for other transfers on the same or other units to finish. The IAS scheduler runs at the timesharing priority. Swap requests from the scheduler to the disk handler are placed after any realtime tasks running at a higher priority, but compete on a first-in first-out basis with other tasks running at equal to or less than the timesharing priority. Also, the total time to perform a swap transfer might increase if the disk head is frequently moved away from the cylinders occupied by the swap files.

In the second case, if swapping is performed on a dedicated unit, the swapping performance improves. Delays still occur if the controller is performing transfers on other units, but can be reduced if the handler for the device supports overlapped seeks.

The third case gives the best swapping performance. No delays occur because no other activity is taking place.

The total swap space can be formed from any number of swap files on any number of volumes. A single volume can have more than one swap file. Whenever the system must swap a task, the swap files are searched for a free area of sufficient size to form a swap area to store the task. The order in which the swap files are specified normally determines the order in which the files are searched for free areas. Therefore, disks that are more suitable for swapping (for example, fixed head disks) should be specified first. A task is never split between two swap files; if it cannot fit anywhere in one swap file it is placed somewhere in the next file.

The size of a swap file is specified in terms of *logical swap blocks*. The allocation of the variable sized swap areas within a swap file is also performed in terms of logical swap blocks. A logical swap block comprises four 256 (decimal) word disk blocks. For example, a 200-block swap file uses 800 blocks of disk space.

You specify the swap space during system generation. However, you can modify the swap space during system operation by using the MCR SWA command, or the SCI commands CREATE/SWAP file, DELETE/SWAP file, and SHOW/SWAP FILE. See the *IAS MCR User's Guide* and Chapter 12 of this manual respectively for details on these commands.

4.2.5 Memory Allocation Parameters

The allocation of memory into contiguous areas, or partitions, can occur only at system generation. Each of these partitions might have the characteristics of user, system, or timesharing as described in Chapter 3. At timesharing system start-up or when you enable the IAS scheduler, you can specify the required timesharing type partition in which the current session's timesharing activity is to occur. This parameter defaults to the partition GEN.

In most systems, there is only one timesharing type partition (GEN) that is always used for the timesharing activity in the system. However, it might be useful to be able to vary the timesharing partition for a current session. This is so if a large partition normally used for timesharing is occasionally required to execute a number of concurrent real-time tasks. In such a case, a smaller timesharing type partition can be used to provide reduced timesharing capabilities as well as real-time activity.

4.3 Device Usage Parameters

NOTE: On a system without TCP, all devices are automatically available to all users and this section therefore does not apply.

Initially, you declare the full complement of devices and their characteristics at system generation.

Additionally, you can specify the following parameters:

- Maximum number of interactive terminals.
- Maximum number of batch streams.
- Maximum number of devices available to timesharing users at timesharing system startup.
- Maximum number of concurrently active timesharing terminals. (You can limit this to a subset of the actual number of timesharing terminals.)
- Maximum number of devices concurrently accessible by any single timesharing user.

You can specify devices to be automatically available to timesharing users on a system wide basis. As a result of this, the devices are automatically allocated and accessible for each timesharing user when the CLI task for the user terminal is made active. All spooled devices and the bootstrapped system device are automatically set as such devices at timesharing system startup.

See Section 4.4 for a description of timesharing system startup. For more details and for a description of how to configure the timesharing executive (TCP), see the *IAS Installation and System Generation Guide*.

4.4 System Startup Procedures

System start-up procedures perform the following functions:

- Start up a newly generated system.
- Override or modify parameters specified at a previous system generation without repeating the SYSGEN process.

The startup procedures are different for timesharing systems, as opposed to multiuser or real-time systems.

On multiuser and real-time systems, the system is normally available for use as soon as the generated system has been bootstrapped. You can specify additional commands (for example, to create swap files) if needed.

A timesharing system has a special timesharing startup procedure. Timesharing system startup specifies the following:

- Devices to be automatically allocated for timesharing users.
- Terminals to be designated as timesharing terminals.

Timesharing system startup consists of a series of commands. You can either enter the timesharing startup commands directly at a terminal or you can create a startup command file to contain them.

If necessary, you can reedit the startup command file before timesharing system startup, if the system must be reconfigured for that session. Otherwise, the command file can remain fixed to ensure that a system with identical characteristics is initiated each time you perform timesharing system startup.

See the *IAS Installation and System Generation Guide* for details on the system startup procedures, including descriptions of the timesharing startup commands.

4.5 Runtime System Control

The special version of PDS that runs on the console terminal (SCI) enables the interactive and batch activity in the system to be dynamically controlled from the operator console.

The system manager or operator can use SCI to perform privileged functions that control the IAS system. For example, you can use SCI to abort any task executing in the system or to install or remove a device handler from the system. Further, you can use SCI to tune the system dynamically, by setting the various system parameters for scheduling, quantum, batch, swapping, memory, and device usage, as described above. Similar facilities are available using the monitor console routine (MCR) on a multiuser system with MCR as the default CLI.

Chapter 12 contains descriptions of all the SCI commands available to the system manager. Table 12–1 lists these commands and their MCR equivalents. For details of the MCR commands, see the *IAS MCR User's Guide*.

5 Operating an IAS System

This chapter contains the information required for the day-to-day operation of the IAS system. The topics covered are:

1 System start-up and initialization procedures

2 Run-time system control

3 Peripheral device handling

4 System recovery and file preservation

5 Control of batch processing

5.1 Introduction

Operating system requests, messages and operator responses provide the means of communication between the operator and the system. This communication is performed on the operator terminal, which is the physical device with which the pseudo-device CO is associated. By default, this is the console terminal.

Operating system messages inform the operator of system activities that affect the system operation. These activities may reflect error or abnormal conditions that need operator intervention. For example, re-running a task which could not execute because of lack of resources.

Operating system requests solicit action from the operator (for example, a request to load a disk pack onto a disk drive).

The system manager/operator can communicate with all users by means of the message facility and the notice facility. With the notice facility, the system displays the contents of the file LB0:[1,1]NOTICE.TXT on user terminals when they log in. Similarly, the system prints the contents of the file LB0:[1,1]BATNOTICE.TXT when a batch job is started. (If BATNOTICE.TXT does not exist, the system prints NOTICE.TXT instead.) The system manager/operator must use EDIT or CREATE to create these files (see the *IAS PDS User's Guide*). If these files are created as fixed length record files, certain PDS commands will not work correctly after the system prints the notice.

The message facility enables you to send messages to other user terminals, by specifying the PDS MESSAGE command (see the *IAS PDS User's Guide*). You can also force messages to a terminal that has requested not to receive messages by specifying the SCI MESSAGE/FORCE command (see Chapter 12).

In a timesharing system, the communication between the operator and the system initially follows timesharing system start-up. After timesharing system start-up, the communication is through the CLI that runs for the console terminal (see Section 5.3). Once the system has been started up, that CLI is available on the operator terminal, although you can log out if you have to leave the terminal.

The console CLI is:

- On a real-time system, usually MCR.

- On a multiuser system, either MCR or the special version of PDS (known as SCI).
- On a timesharing system, the special version of PDS (known as SCI).

With SCI, as with PDS, you must log in before you can enter commands. SCI differs from PDS in the following ways:

1 The operator terminal is automatically logged in when the system is started up or bootstrapped, with the user name SCITERMINAL. It is not possible to log in explicitly with this user name.

2 Logging in and out:

It is possible to log out and log in at the operator console, using the LOGOUT and <CTRL/C> commands. However, you need special privilege (PR.SCI) to log in at this terminal because of the extra facilities automatically granted.

2 Whichever user is logged in, the operator console can always be used for SCI commands (described in Chapter 12).

3 Certain commands entered at the operator terminal are executed at a priority of 220, above the normal priority for real-time tasks. This makes it possible, for example, to abort a real-time task that is looping, preventing normal timesharing tasks from running. Note that only control-type commands receive high priority. Program development commands receive only the usual priority and do not pre-empt real-time tasks.

4 The SCI terminal is not automatically logged out when timeout occurs. Instead, the terminal prints TIMEOUT followed by a carriage return. Typing Ctrl/C causes the prompt to be repeated. The timeout interval for SCI is normally much shorter than for PDS (30 seconds unless altered by rebuilding PDS).

5 Once the SHUTDOWN command has been issued to initiate system closedown, you cannot log out from the SCI terminal.

On a multiuser or real-time system, any user with the privilege PR.SCI can issue SCI commands from any terminal. However, messages directed to the operator, by means of the device CO:, appear only at the operator console.

5.2 Initializing and Starting Up The System

You must begin by loading the system volume on the appropriate device. The system volume is a disk pack containing the system generated for that specific installation (see the *IAS Installation and System Generation Guide*). You must then use the appropriate bootstrapping procedure.

When you have successfully bootstrapped the system, the sequence of events is as follows:

1 A message of the following form is printed on the operator console:

```
nnnk (WORD) IAS mm.m
```

Where:

- nnk = Amount of memory available to the system.
- mm.m = Version of the system being run.

At this point you might see displayed warning messages from SAV. Take appropriate action; for example, reinstall any tasks that SAV has removed and, if necessary, save the system.

NOTE: If a task that maps onto a shareable global area (SGA) is removed by SAV, it will not be possible to remove the SGA and its reference count will never be zero. Perform a system generation to rectify this.

2 The system disk is automatically mounted, and the mount information is printed. This does not occur if the system disk was dismounted before the system was saved.

3 The current time and date, as known to the system, are printed. (This is the time and date at which the system was last saved.)

4 The prompt DATE AND TIME? is printed. In response to this, you type the date and time in the form:

```
dd-MMM-yy hh:mm
```

dd, yy, hh, and mm are decimal integers indicating day of the month, year, hour, and minutes past the hour, respectively. MMM is an abbreviation for the name of the month, consisting of its first three letters.

5 A command prompt, either SCI> or MCR>, appears. On a multiuser or real-time system, start-up is complete. Further commands might be required for a particular installation to get the system fully active (for example, loading non-resident device handlers). These can be MCR or SCI commands. You can create an indirect command file to contain these commands, if you are going to issue the same sequence of commands every time the system is started.

6 On a timesharing system, you might have to perform other steps before timesharing can start (for example, mounting other disks). Specify the necessary commands using the MCR interface, which is fully described in the *IAS MCR User's Guide*. The following commands should be particularly useful:

- EDI—Edit the start-up command file.
- INI—Initialize a disk volume.
- LOAD—Load a device handler.
- MOUNT—Mount disks.
- SET—Set various system parameters.
- SWAP—Alter the swap file configuration.
- UFD—Create directories on newly initialized volumes.
- UTL—Alter scheduler parameters, which you cannot change once timesharing is active.

If you are going to issue the same sequence of commands every time the system is started, create an indirect command file and invoke it using the @ symbol. For example:

```
MCR>@MCRSTART
```

Where the file MCRSTART.CMD contains the required command sequence.

7 You can now start timesharing by using the IAS command in either of the following ways:

```
MCR>IAS @startupfile
```

Where startupfile is the name of a file containing timesharing system start-up commands, or:

```
MCR>IAS

IAS> start-up commands

IAS> ...

IAS> SET START
```

You must not issue this command from an indirect MCR command file. Startup commands are described in the *IAS Installation and System Generation Guide*.

It is not possible to use a single indirect command file to start up timesharing and issue DCL commands, because two command processors are involved (MCR for invoking start-up and PDS for DCL commands).

8 The operator terminal is automatically logged in with a username of SCITERMINAL, and the usual login information appears. The terminal prompts SCI, indicating that timesharing has been started.

5.3 Runtime System Control

5.3.1 Console Interface

When the IAS system is operational on a multiuser system, the operator interface to the system is either the SCI or MCR interface. On a timesharing system, the operator interface is SCI. On a real-time system, the operator interface is MCR, and although SCI is possible, it is not recommended.

If there is no prompt on the operator terminal, type [Ctrl/C] to cause either SCI or MCR to prompt. If you do not reply within a certain period, a timeout occurs and you must type [Ctrl/C] to repeat the prompt.

5.3.2 Overall System and Task Execution Control

By means of SCI or MCR, the operator can control the system such that its workload can be satisfactorily serviced. This includes:

1 Loading and unloading of device handlers that are not permanently resident (for example, only needed by a single task during the course of a session).

NOTE: Loading handlers only when required can cause fragmentation of the partition. If a handler that connects to interrupts is loaded in the timesharing partition then the handler cannot be shuffled. This applies to all supplied handlers with the exception of the system pseudo-device handlers PI,MO,NL, and BA.

2 Setting system parameters to suit the current workload.

3 Closing:

Closing down the system.

SCI and MCR also provide the operator with ultimate control over task execution in the system. Facilities for the termination of any abortable task in the system are provided. For example, you might want to stop a development batch task which, due to internal errors, loops indefinitely.

In a situation where the system workload is fairly constant and the likelihood of "runaway" programs is slight, the operator interaction in this area should be minimal.

See Chapter 12 for details on the SCI commands, and see the *IAS MCR User's Guide* for details on MCR commands. Table 12–1 lists the SCI commands with their MCR equivalents.

5.3.3 Device Control

During the course of system operation, system users might require access to disk packs, magnetic tapes, or other removable media. This requires the operator to load the volumes physically onto the device units. Users can request a device or volume without operator intervention by specifying the /NOOP[ERATOR] qualifier to the MOUNT command (see Chapter 12).

Users might want output to be printed on different types of stationery. Different types of stationery are referred to as forms types. The operator can change the type of stationery at any time by setting the forms types for output spooled devices. See Section 5.4.5.2 for further details.

5.3.4 System Information

You can display specific information about system operation on the operator terminal by means of the SHOW command (SCI) or the corresponding MCR commands. The SHOW command is described in Chapter 12. See the *IAS MCR User's Guide* for details on the MCR commands.

5.3.5 Command Language Interpreter Control

The installation of CLI subsystems and the allocation of CLIs to the timesharing terminals in the system is normally performed only once by the operator during initialization operations. You can do this by means of a command file if day-to-day variations of CLI allocation are not required. You can dynamically modify the CLIs running in the system and the CLI/terminal relationships while the system is running. If necessary, you can abort a CLI.

> You can request any CLI task running for an input device to terminate in a specified time (except for the console terminal) or, for a batch CLI, to stop at the end of the current batch job. On a timesharing system, either the console alone or all terminals can be automatically allocated to the default CLI, PDS.

On a multiuser system, the default CLI can be either PDS or MCR, and is specified through the DCLI directive at system generation (see the *IAS Installation and System Generation Guide*).

5.4 Peripheral Device Handling

5.4.1 Device Errors

Device error messages can occur when a device unit or volume on that unit cannot be accessed initially, or when an error condition is encountered during the accessing of the device or volume.

Initial device errors are indicated by a message of the form:

```
ddnn NOT READY
```

or, for magnetic tapes or DECtapes:

```
SELECT ERROR ON ddnn
```

Initial device errors normally occur because of some omission in setting up the device unit (for example, not switching the unit on-line, volume not loaded on a device, or a disk pack not up to speed). Rectify such device errors by setting the device operational according to the operational instructions for that device (see the *PDP-11 Peripherals Handbook*). On some occasions, the device might fail to become operational in spite of using correct procedures. In this case, the hardware might be at fault and require the attention of the Field Service Engineer.

The system can also display error messages relating to device operations (for example, mounting a volume). These error messages are self-explanatory.

If error conditions occur during the accessing of a device, the system tries the operation again a number of times, according to the handler servicing the device. If the failure persists, it could be due to a bad block on the physical medium being processed (for example, a bent or mispunched card or a damaged disk pack). However, it could be a true hardware error, in which case you must call the Field Service Engineer. These types of errors are not logged on the operator console but will be logged if error logging (ERRLOG or ERRDSA) is running. See Chapter 9 for details about ERRLOG and ERRDSA.

The *not ready* condition for the card reader always occurs when all cards in the hopper have been read. This occurs until more cards or an end-of-file card (rows 12, 11, 0, 1, 6, 7, 8, 9 punched in column 1 of a card) are placed in the hopper.

5.4.2 Device Loading Requests

On a timesharing system, the operator controls physical loading of volumes onto devices. However, if users do not require operator intervention, they can mount a volume using the /NOOP[ERATOR] qualifier (see Chapter 12). Alternatively, you can specify the /NOOP qualifier as the default when you build PDS at system generation (see the *IAS Installation and System Generation Guide*). When the system requires the physical loading or unloading of volumes, it makes an appropriate request to the operator. A request for volume load specifies the volume identification and device type, and either the unit number requested or an indication that any unit of the requested device type may be used. For example:

```
LOAD volumeid ON ddnn or LOAD volumeid ON ANY dd
```

The request can also specify that write protection is required. For example:

```
LOAD volumeid ON ddnn WRITE PROTECTED
```

In this case the appropriate write protection procedures according to device type must be executed, for example, omitting a write permit ring for magnetic tape or setting the write protect switch for a disk drive. If the operator can satisfy the request (that is, an appropriate device is free and the requested volume has been physically loaded), the operator must indicate to the system that this is the case. The response is of the form:

```
LOAD ddnn: [volumeid]
```

If the operator cannot satisfy the request (for example, because the requested volume is unavailable) the operator must enter the following command:

```
NOLOAD ddnn: [volumeid]
```

This informs the user that the load request could not be satisfied. In this way, users need not wait for device availability but can continue their processing and attempt to obtain a free drive again at a later time.

Normally the system checks that the specified unit or a specified device of the required type is available before making a LOAD request to the operator. However, the system cannot detect that a drive is malfunctioning and not in use; in this case, LOAD requests might occur when there is no usable free drive. Users can request a device or volume indicating that no operator intervention is required by specifying the /NOOP[ERATOR] qualifier to the MOUNT command (see Chapter 12). In this case, a LOAD request is not made to the operator.

The current state of all selected devices in the system can be displayed at any time by means of the SHOW DEVICES command (see Chapter 12).

Whenever a volume is mounted, a message of the following form displays:

```
MOUNT -- ddn: *** MOUNT COMPLETE ***
```

This enables the operator to keep track of which volumes are currently known to the system. You can display unsatisfied LOAD requests by specifying the SHOW VOLUMES command, to inform you of outstanding LOAD requests. See Chapter 12 for details about the SHOW VOLUMES command.

5.4.3 Device Unloading Requests

When all users of a removable volume have finished using it, the system requests the operator to unload the volume. Where the volume label is known to the system, the device name, unit number, and volume label are all stated in the following form:

```
UNLOAD volumeid FROM ddnn
```

Otherwise, the device name and unit number only are specified in the following form:

```
UNLOAD ddnn
```

The unload request is followed by this message:

```
ddnn--DISMOUNT COMPLETE
```

The UNLOAD request also occurs if for some reason the software fails to identify the volume as a result of a LOAD request. This might be as a result of a user supplying incorrect parameters, and the request is frequently followed immediately by another LOAD request. The UNLOAD request in this case is accompanied by a second message prefixed MOUNT that gives the reason for the inability to recognize the volume and can thus be identified. The operator unloads the indicated device, but unloading never needs an SCI command.

5.4.4 Multivolume Magnetic Tape

When the data required on a Files-11 magnetic tape cannot fit onto a single magnetic tape volume, if the tape is loaded for multivolume tapes the system automatically requests another volume to be loaded. The following conditions must then be met:

1 You must load the correct magnetic tape handler and install the ACP task. MTAACP is the default ACP for Files-11 magnetic tape volumes.

2 You must initialize all volumes (see Section 6.2.1).

3 You must mount the volumes in sequence. The system prompts:

```
MOUNT NEXT VOLUME ON MMnn
```

until all volumes have been mounted. You must load each volume on the appropriate device and set the device on-line. The system checks to see that each label name is correct and automatically loads each volume. If the volume is Files-11 and if the label is not correct, the system sends the following message:

```
MTAACP -- WRONG VOLUME MOUNTED ON DRIVE MMnn
```

If you do not take any action, the system starts reprompting for the next volume, with the following three messages:

```
MTAACP -- REQUESTED ACTION FOR DRIVE MMnn NOT COMPLETED
MTAACP -- PLACE DRIVE MMnn ON-LINE
MTAACP -- PLEASE COMPLETE OR ABORT TASK
```

5.4.5 Spooled Device Control

The need to wait for the direct availability of a slow serial device can be eliminated by setting the device in question as a spooled device (see Section 6.3). The system manager normally decides which input and output devices to set as spooled devices.

5.4.5.1 Running Batch from Input Spooled Device

If card input is required for batch input, you must set the card reader spooled (see Chapter 12, the SET SPOOL SCI command, or the *IAS MCR User's Guide*, the SET /SP MCR command and install the input spooler in the system.

The input spooler runs as a real-time task. You must first select the priority at which the input spooler is to run (for example, a priority either above or below timesharing). If the priority is set below timesharing and is run in the timesharing partition, the input spooler competes for time like any other scheduler-controlled task. Secondly, you must select the partition in which the input spooler is to run. If it runs in the timesharing partition and is of a lower priority than timesharing, it can be checkpointed to make space for scheduler-controlled tasks.

> For batch processing, you must specify the pseudo-devices BA0, BA1...BAn (where n is the number of batch streams) as timesharing terminals (not devices) during timesharing system start-up, and you must make a CLI active on those terminals (see Chapter 12, the RUN command). When a batch terminal is assigned to a CLI, you must initiate the CLI to run on that terminal, since initiation by typing [Ctrl/C] is not possible. When batch runs from queued input, it continually looks for jobs in the queue. Batch only terminates in response to a STOP, SHUTDOWN or ABORT command (see Chapter 12).

On multiuser or real-time systems, you initiate batch CLIs with the DCL command RUN/REAL or the MCR command RUN. You can terminate batch CLIs with the DCL command ABORT/REAL or the MCR command ABO (see Section 5.6.2).

5.4.5.2 Setting Form Types on Output Spooled Devices

Output might require printing on different types of stationery. The IAS system provides for seven different forms types in an installation.
The form types are referred to in the system by a number in the range 0 to 6 inclusive. Each forms type can be associated by the local installation with a specific type of output stationery (for example, 3-part paper or pre-printed forms).

Most files are queued with a default form type of zero. When a different form type has been specified by the user, the operator receives a message to load the line printer with a different stationery. You can then, at any time after the request, change the stationery on the appropriate output device by performing the following actions:

1 Issue a SPOOL/TEST (SCI) or OPR /TE (MCR) command to stop output on the specified device at the end of the next file. If any test files have been queued (QUEUE/TEST or QUE /TE) these are printed and can be used to align the new stationery by stopping (SPOOL/STOP or OPR /ST) the output and retrying it (SPOOL/RESUME or OPR /RS) until it is correct. The device form type is then set to the number associated with the required stationery (SET FORMS or OPR /FO). See the *IAS MCR User's Guide* for details on the OPR command, and see Chapter 12 for details of the SPOOL command.

2 You can queue test files with the QUEUE/TEST or QUE /TE command, in which case they will be immediately printed. Once the file (or files) for the special stationery has been output, replace the original stationery and reset the forms type for the device to its previous value (usually 0).

3 You can stop output on a spooled device at any time. For short periods, merely switching the device off-line may be sufficient, for example, to realign the stationery. However, if the device is off-line for more than a few minutes, the device will timeout and you must restart it with a SPOOL/RESUME or OPR /RS command. You can change the timeout period by rebuilding SPR2.

4 You can issue the command SPOOL/STOP or OPR /ST to cause an output spooled device to stop printing. Restore output using the SPOOL/RESUME or OPR /RS command.

You can abort the file currently being processed on a spooled device by the SPOOL/ABORT or OPR /AB command, in which case the rest of the file is not printed and the next queued file is selected for printing.

NOTE: If a user has queued a file with the /DELETE or /DE qualifiers (see the *IAS PDS User's Guide* or the *IAS MCR User's Guide*), and you abort it, the file is preserved.

5.4.5.3 Spool Device Errors

If device errors persistently occur for a spooled device, suggesting that the hardware is in error, you can redirect the output queued for the unusable device to a different device. To achieve this, you must terminate the output on the failing device as follows:

1 Set the device as if a forms change were required (to prevent any further queued files being output).

2 Either abort the current output file (if the output would be useless) or wait until the current output file has terminated.

You must then abort the output spooler task, SPR2.., if it is active (ABORT/REALTIME SPR2.. ddn or ABO SPR2../TI=ddn, where ddn is the failing device). You can then redirect the device, using the ASSIGN/REDIRECT (SCI) or RED (MCR) command. You can start the spooled output using the SPOOL/START or OPR /ddn:ST command (where ddn: is the device name).

While the output spooled device is redirected, any files queued to the original device are automatically printed on the new device. Inspection of the queue shows the device for which each file is queued. When a queued file becomes the current file being printed, its device specification field will show the actual device on which the output is taking place.

Internal errors in the spooling software occasionally occur when the spooling environment has been corrupted. A message from task SPR2.. can indicate such an error. In this event, obtain a list of all spooled output devices if you do not already know which ones they are (using SHOW DEVICES (SCI) or DEV (MCR)), and terminate the output on them by a SPOOL/TEST (SCI) or OPR /TE (MCR) command. When all spooled output devices have become inactive, abort all copies of the task SPR2.. (ABORT/REALTIME SPR2.. ddn or ABO SPR2../TI=ddn). Obtain the values for ddn by typing the command SHOW TASK SPR2.. ALL (SCI) or ACT SPR2../AL (MCR). Reset the output spooled device forms types to their original values. The spooled output then automatically resumes.

A message from the task SPR... indicates persistent errors in the queuing mechanism. Use the following error recovery procedures:

1 Abort the queue manager if it is active (ABORT/REALTIME SPR... or ABO SPR...).

2 List the queue file to determine which files are queued to which spooled devices (QUEUE/ALL).

3 Delete the queue file, SP0:[1,4]SPRQUEUE.SYS;1.

If any batch streams are allocated, stop them (STOP/CLI (BA0...BAn)) before deleting the queue file, then restart them (RUN/CLI (BA0...BAn)) after deleting the queue file.

4 Obtain a directory of SP0:[1,4] and if it contains any files of type .SPR, requeue these files to the appropriate device as shown in the listing of the queue file.

5.5 System Recovery and File Preservation

This section describes the following occurrences:

1 Automatic recovery (Section 5.5.1)

2 Total system failure (Section 5.5.2)

3 Update (Section 5.5.3)

4 Files and media backup (Section 5.5.4)

5.5.1 Automatic Recovery

IAS initiates automatic error recovery procedures whenever it encounters a device hardware error or a system power failure. On a device error, the system retries the failed operation a number of times, depending upon the device. If it is successful, the system operation continues uninterrupted although the occurrence of the error may be recorded in the error log file (see Chapter 9). If it is not successful, the system returns a device error.

If the power level drops below the minimum required for the operation of the hardware, the system enters its power fail recovery routine. Provided that the power is restored rapidly, the system users and operator wll be unaware that it has occurred. If power is not restored, the system will appear to be effectively switched off. On restoration of power, the system will resume. However, in this case, some devices (such as magnetic tape) must be manually restored and any active operation terminated. Many devices, (for example, lineprinters) must be manually turned on-line before they resume operation.

NOTE: Power fail recovery is available only on systems with battery backup facilities.

5.5.2 Total System Failure

On certain occasions, the system is unable to recover automatically from a system failure. A message of one of three types can be sent to the console to notify you that a system crash has occurred.

```
CRASH -- CONT WITH SCRATCH ON ddnn

CRASH -- reason for crash -- CONT WITH SCRATCH ON ddnn
```

Where:

ddnn = Device specification

or:

```
CRASH -- NO CRASH MODULE RESIDENT
```

If the system is generated not to produce a crash dump, the system simply halts at location 44 and must then be rebootstrapped and restarted.

In the first and second cases, if desired, a dump of the system memory can be made at this point in order to assist in tracing the cause of the system failure.

Load a scratch volume on the requested device and press the *continue* switch on the operating panel. (Some processors might not have an operating panel. In these cases, refer to the appropriate PDP-11 Processor Handbook for the action required to make the processor continue). A memory dump is then written to the volume. The processor halts at completion of the dump. The system should now be rebootstrapped. If a dump was taken, it can be analyzed during the normal running of the system (see Chapter 8).

You should back up your system periodically to minimize the effects of failures when they occur. See Section 5.5.4 for details. By following such procedures, you can always recover the system and its files in the event of the system or user file corruption.

In almost all cases, a system failure does not cause system or user file corruption and the system can be reactivated by rebootstrapping (see Section 5.2). In all cases of this type of system failure, open files are not closed and saved. This includes both system and user files. After rebootstrapping the system, the files open at the time of the crash must be attended to; some can be left locked and others left with blocks allocated. The PDS command SET END_OF_FILE or the MCR command PIP /EOF might be of some use as a file information recovery procedure. Also recommended is a message issued to all users of the system after the restart, informing them of the occurrence of a system crash. You can use either the PDS notice or message facility for this purpose (see the MESSAGE command in the *IAS PDS User's Guide* for details).

The User Profile File can become locked as a result of a crash, in which case timesharing cannot be started. You must then issue the following command to MCR before staring timesharing:

```
PIP LB0:[1,100]PDSUPF.DAT/UN
```

The input and output spooling operations are automatically re-activated when timesharing is restarted.

Items queued at the time of the crash are not lost and are processed once the system has been restarted. Any batch jobs active at the time of the crash are lost.

The system files are generally be intact, since write operations to these files are infrequent. However, if you have trouble performing file operations while running the system after a crash, inspection of the appropriate file might indicate that it is locked or of zero length. In this case, unlock or recreate the file. If the system disk is badly corrupted, you must recreate it from previously archived tapes. This is a radical step causing any files created on the disk (system or user) after the archive to be lost and you should do this only if there are no other means of getting the system restarted.

5.5.3 Update

Update enables the system manager to apply patches to the system automatically. Update updates the system disk directly, and contains corrected files to replace those on the system.

Update provides a short dialogue with the system manager, and then runs without intervention. Documentation for Update is provided by files on the Update kit itself.

When you generate a system with Update, see the *IAS Installation and System Generation Guide* for details.

5.5.4 Files and Media Backup

IAS files can be copied to secondary storage volumes in one of two ways. Either a snapshot of the complete volume contents can be written to magnetic tape, or its file contents (or selected files) can be written onto any other backup medium.

The first of the backup procedures is performed using the BRU (Backup and Restore) utility. See the *IAS Utilities Manual* for details on BRU.

The preservation of a volume performs a complete transfer of all data on the volume. Volume preservation is normally used for system volumes on which the rate of update of permanent data is infrequent. It is essential that the system be quiescent (no active users) when a system disk is preserved. If the system is not quiescent, there will normally be scratch pad areas in use on the system disk (for example, temporary user files, spool files). Copying the system disk in such a condition will not provide a suitable basis for creation of a clean system disk.

You can use BRU on any volume where a copy of the total contents is required for backup purposes. BRU compresses data while transferring files to the backup volume, thus freeing space on the volume.

User files are normally backed up onto private volumes by the users themselves. However, if an installation has specific common user volumes, the system manager may decide to implement some periodic preservation or selective copying of user files.

5.6 Batch

This section describes how the IAS operator can control the operation of batch processing under IAS. Because batch is handled quite differently on a timesharing system from a multiuser system, the two are described independently. Use of batch is not recommended on a real-time system.

5.6.1 Batch Processing

The queueing of batch jobs is handled similarly to the queueing of line printer output. When you submit a batch job to the system using the PDS command SUBMIT or the MCR command QUE, the job is placed in a queue for the pseudo-device called BA0. Whenever a batch CLI is ready to process another job, the next entry is taken from this queue.

It is possible to run multiple batch streams, so that more than one batch job can be processed at the same time. In this case, each stream has its own CLI and operates for a different unit of the BA pseudo device. However, all batch jobs are taken from the same queue, BA0. You cannot control the stream in which a job will run. In addition to the CLI, each batch stream must have a handler task for the BA pseudo-device. This is performed by a multiuser task called BA...., one copy of which runs for each batch unit.

For a particular batch stream (that is, unit of the BA pseudo-drive) to be able to run batch jobs, both the handler and the CLI must be active. Once activated, the CLI remains so until stopped or shut down (on a timesharing system) or aborted (on a multiuser system).

A batch log is created for every batch job that is run. The log consists of one or more concatenated files that resulted from the job, and is automatically spooled to CL under the file name LP.SPR.

NOTE: The batch handler for a stream must NEVER be unloaded (UNLOAD or STOP/HANDLER) while a CLI is active for that stream. The effect of doing so is unpredictable.

5.6.2 Controlling Batch on a Timesharing System

The following sections describe how to control batch on a timesharing system.

5.6.2.1 Starting Batch

To start a batch stream and enable it to run jobs, load the batch handler for the unit and run the CLI:

```
SCI> RUN/HANDLER BA[n]

SCI> RUN/CLI BAn
```

Usually, the batch handler is started during system generation. In this case, omit the RUN/HA BAn command.

Normally, batch jobs will run in the batch scheduling level. To run batch jobs as though they were interactive, you must deallocate the CLI, then allocate it and run it again as an interactive CLI:

```
SCI> DEALLOCATE/TERMINAL BAn

SCI> ALLOCATE/TERMINAL BAn PDS (RUN NOBATCH)
```

5.6.2.2 Stopping Batch

To stop batch processing, use the STOP/CLI command with the TIME qualifier, the END_OF_JOB qualifier, or both, as follows:

```
SCI> STOP/CLI PDS BAn TIME:m
```

or:

```
SCI> STOP/CLI PDS BAn END_OF_JOB
```

or:

```
SCI> STOP/CLI PDS BAn TIME:m END_OF_JOB
```

m is the time (in minutes) after which batch processing is to stop for this stream. To stop batch processing immediately, specify time as zero. To stop batch processing when the currently active job has completed, specify END_OF_JOB. Batch processing for that stream will stop immediately if there is no currently active job. Specifying both TIME and END_OF_JOB will cause the stream to terminate after the given time or at the end of the current job—whichever comes first.

You can also use the STOP/CLI command to terminate a particular batch job which is currently running on a stream. In this case, to restart the CLI, use the RUN/CLI command when the job has ceased to be active.

To determine which stream is running a particular batch job, scan the console output to find the start message for the job, for example:

```
10:35:36 START PDS username jobname BAn
```

5.6.3 Controlling Batch on a Multiuser System

5.6.3.1 Starting Batch

To start a batch stream and enable it to run jobs, you must load the batch handler for the unit, and you must run the CLI as follows:

```
SCI> RUN/HANDLER BAn
SCI> RUN/REALTIME ...PDX BAn
```

or:

```
MCR>LOA BAn
MCR>RUN ...PDX/TI=BAn
```

Usually, the batch handler is started during system generation. In this case, omit the RUN/HA or LOA command.

To use PDS commands on a multiuser system where MCR is the default CLI, the following tasks must be installed:

```
[11,1]PDX
[11,1]DCL
[11,1]SUIC
```

On a multiuser system there is no batch scheduling level. All batch tasks compete with other tasks which run under scheduler control.

5.6.3.2 Stopping Batch

You cannot stop batch easily on a non-timesharing system. You must abort the CLI using the command:

```
SCI> ABORT/REALTIME ...PDX BAn
```

or:

```
MCR>ABO ...PDX/TI=BAn
```

6 Management of Devices, Volumes and Files

6.1 Devices

The standard IAS system supports all the physical devices listed in Table 6–1. The two-character names and the unit number are used in the device field of file specifications.

Table 6–1 IAS Physical Device Types

Mnemonic	Device
ADn	AD01 A/D converter
AFn	AFC11 analog input
CRn	Card reader
CTn	Cassette tape
DBn	RP04/05/06 disk
DDn	TU58 tape cartridge
DFn	RF11 disk
DKn	RK05 disk
DLn	RL01/02 disk
DMn	RK06/07 disk
DPn	RP02/03 disk
DRn	RM02/03/05 disk
DSn	RS03/04 disk
DTn	DECtape
DUn	RA60/70/80/81/82/90, RD50/51/52/53/54, RX33/50, RD31/32 disks
DXn	RX01 floppy disk
DYn	RX02 floppy disk
LPn	Line printer
LSn	LPS laboratory peripheral system
MMn	TU16/45/77/TE16 magnetic tape
MSn	TS11/TS05 magnetic tape
MTn	TU10/TE10/TS03 magnetic tape
MUn	TU81, TK50 magnetic tape
PPn	Paper tape punch
PRn	Paper tape reader
TTn	Terminal
UDn	UDC 11 universal digital control

In addition, you can define pseudo-devices and associate them with the physical devices in the system. See the *IAS Installation and System Generation Guide* for details on establishing pseudo-devices. See Chapter 2 for a description of the standard pseudo-devices.

6.1.1 Device Characteristics

You set the device characteristics for each device at system generation. You can define the following characteristics:

- A two-character ASCII mnemonic that refers to the device.
- A one- or two-digit octal unit number.
- Either a device type used by Phase 1 of system generation to determine the device characteristics, or the device characteristics themselves (see the *IAS Installation and System Generation Guide*).
- The PDP-11 device interrupt vector address.
- The software priority at which device interrupts are to be serviced.
- The external page address of the device controller.
- An optional default file primitives task name or an ancillary control processor (ACP) name for file-oriented devices. IAS uses the ACP to control all file-processing functions such as CREATE, DELETE, and ACCESS. The ACP is associated with a volume when that volume is mounted. You can override the default ACP at mount time. The device handler controls basic operations such as read and write.

See the *IAS Installation and System Generation Guide* for a detailed description of device characteristic settings.

6.1.2 Device Handlers

A device cannot be used unless a device handler that services that device is resident. Device handler tasks perform the functions that enable the physical I/O operations to occur on supported PDP-11 devices. Because handlers are tasks, you can load infrequently used handlers when you need them and unload them after use.

NOTE: Loading handlers only when required might cause fragmentation of the partition. If a handler that connects to interrupts is loaded in the timesharing partition, the handler cannot be shuffled. This applies to all supplied handlers with the exception of the system pseudo-device handlers PI,MO,NL, and BA.

You can also write an installation-specific handler to service non-standard devices.

See the *IAS Device Handlers Reference Manual* for a detailed description of the device handlers supplied with IAS.

See the *IAS Guide to Writing a Device Handler Task* for details on how to write a device handler for a non-standard device.

6.1.3 Device Availability

Specify devices configured in the system at system generation, and specify those available to timesharing users at timesharing system startup. Devices can be used by all timesharing users in the system and can be shared or single-user, according to the device characteristics. Additionally, devices that are normally shared can be temporarily used exclusively by a single user on demand by means of the PDS ALLOCATE or MOUNT/NOSHARE commands. At runtime, a timesharing terminal belongs to the CLI that is running for the terminal, and thus to the user who is logged in on that terminal. SCI performs dynamic re-allocation of terminals to CLIs.

All system devices (specified as /S timesharing devices during startup by the SET DEVICE timesharing start-up command), any output spooled devices, and the user terminal are allocated to the user when the CLI becomes active (when the user types CTRL/C or enters the RUN/CLI command). If a system device is a files device, the volume on that device is accessible to any user only if the volume is already mounted *globally*. To mount the volume globally, mount it before timesharing is started, or specify the PDS command MOUNT/GLOBAL. Mounting any volume globally ensures that it does not get dismounted, even when there are no active timesharing users on the system. This is especially useful for volumes of user files that must be permanently on-line.

The PDS SHOW DEVICES command lists information about all the devices in the system. This list can include the following information:

- Device name
- Unit number
- Whether the device is mounted
- Type of user the device is mounted for
- Number of timesharing users
- Whether the device is spooled
- Any redirection information

On a multiuser or real-time system, all devices are available for all users.

6.2 Volumes

IAS supports the following types of volumes:

1 FILES-11 disk and DECtape

2 FILES-11 magnetic tape

3 Other (foreign)

Files-11 is the IAS file control system that provides facilities for the dynamic creation, extension, and deletion of files on disk, DECtape, or magnetic tape. To use the FILES-11 facilities, initialize your volume as FILES-11.

A FILES-11 disk or DECtape volume consists of a collection of files that reside on a single volume. The system can address each file on the volume directly by means of file pointers that reside in the volume directory files (that is, the master file directory and user file directory). Directory files are described in the *IAS I/O Operations Reference Manual*.

Magnetic tape volumes are supported according to ANSI standards for single- and multi-volume tapes.

All file processing is controlled through the following types of *ancillary control processors (ACPs)*:

- F11ACP—Used for FILES-11 disk and DECtape volumes
- MTAACP—Used for magnetic tapes

Because F11ACP processes serially, if you use it for DECtape, performance is adversely affected. Use DTAACP for installations requiring significant amounts of DECtape processing. See the *IAS Performance and Tuning Guide* for further details.

You can also gain access to other volumes in formats other than FILES-11 as follows:

1 Designate the volume “foreign.”

2 Perform accessing functions not included in the IAS file system.

Prior to an operator request for volume load, the system checks for device availability. While a volume is being mounted, the system checks your access rights to that volume, based on the volume protection code (see Section 6.2.3). If you do not have the appropriate access rights, your requests are rejected.

6.2.1 Initialization

Use the INITIALIZE or INI command to initialize volumes. (See the *IAS PDS User's Guide* or the *IAS MCR User's Guide*.) Any data stored on the volume is destroyed at initialization time.

Each FILES-11 volume contains a *volume label* that identifies the volume. To avoid mounting an incorrect volume, give each volume a unique volume label.

You can use the PDS INITIALIZE command to create any of the following types of volumes:

- FILES-11 structured volumes on disk or DECtape.
- ANSI structured volumes on magnetic tape.
- DOS structured volumes on disk, DECtape, or magnetic tape.
- RT11 structured volumes on disk, DECtape, cassette tape, or floppy disk.

You can use the MCR INI command to create either of the following types of volumes:

- FILES-11 structured volumes on disk or DECtape.
- ANSI structured volumes on magnetic tape.

You can use the FLX (/ZE switch) utility to initialize DOS or RT11 volumes or to create either of the following types of volumes:

- DOS structured volumes on disk, DECtape, or magnetic tape.
- RT11 structured volumes on disk, DECtape, cassette tape, or floppy disk.

6.2.2 Labeling

The *volume label* assigned to a FILES-11 volume can range from one to twelve alphanumeric characters. The volume label assigned to a magnetic tape volume can range from one to six alphanumeric characters.

6.2.3 Protection

FILES-11 provides you with a multilevel scheme for protecting volumes and files against unauthorized access. You can specify protection attributes for any of the following entities:

- An entire volume
- A directory
- Each file within the volume, regardless of its contents

IAS defines four levels of access for the purpose of assigning protection masks:

- Read (R)
- Write (W)
- Extend (E)—You must have extend access to the task image file to install or run a task.
- Delete (D)

The *protection mask* designates the type of access permitted to each *ownership category*. Table 6–2 describes the ownership categories.

Table 6–2 File Ownership Categories

Ownership	Description
System	Comprises all tasks that run under a system UIC. A system UIC has a group number of 10 octal or less (for example, [1,10]).
Owner	Comprises all tasks that run under the same UIC as the owner of the volume or file.
Group	Comprises all tasks that run under a UIC with the same group number as the UIC of the owner's volume or file.
World	Comprises all tasks, including the three categories described above.

Each volume has a protection mask. The default is (RWED, RWED, RWED, RWED). Each volume has a default file protection that is assigned to any file created on that volume if you do not specify protection attributes for that file. Magnetic tape volumes have only a volume level protection mask (that is, the protection assigned to the volume applies equally to every file within that volume).

See the *IAS PDS User's Guide* or the *IAS MCR User's Guide* for further details on volume and file protection.

6.2.4 Volume and File Access

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

1. Compares the task UIC to the file or volume owner UIC to determine the task category (System, Owner, Group, World).
2. After the task category is ascertained, inspects the volume protection word to determine whether a task of this category is allowed access to the file or volume.
3. Further analyzes the file protection word to determine if the desired function (Read, Write, Extend, or Delete) is permitted for this category of task.

If all the above checks yield positive results (that is, the task is authorized access), access is granted.

If the task belongs to more than one category, steps 2 and 3 are repeated until each category has been checked. If all the checks yield positive results (that is, the task is authorized access), access is granted.

When a file is accessed, these checks are performed three times:

1. For the volume
2. For the UFD
3. For the particular file

The task must pass all three checks to be granted access.

When a user mounts a FILES-11 magnetic tape or a foreign volume on a device, exclusive access is implicitly granted to the device on a timesharing system. FILES-11 disk and DECtape volumes can either be shared or, if necessary, be exclusively allocated.

6.3 Spooling

A single-user device can be made *spooled*. This enables you not to have to wait for a serial-processing, single-user device to become free. In this event, output directed to the device is automatically redirected to disk files, and requests to output the files are queued. The queued requests are serviced in priority order, then first-in/first-out order; the files are output to the specified device as soon as possible after receiving the request for output. Alternatively, you can specify that you want the specified output to be processed after a certain time of day, by means of the /AFTER qualifier to the PDS QUEUE or PRINT command (or the /AF switch to the MCR QUE command). See the *IAS PDS User's Guide* or the *IAS MCR User's Guide* for details.

IAS also provides spooling for files that are submitted from a card reader. This operation is called input spooling (see Section 6.3.2).

Note that the size of the spool area affects system performance. The system manager should also ensure that the disk containing the spool area has sufficient free space for all the files being spooled at any given time.

6.3.1 Output Spooling

Output spooling is only available to programs using the IAS file system.

To use output spooling, you must meet the following system requirements:

- Create a pseudo-device called SP during system generation. Normally, SP is assigned to SY. However, you can redirect the SP pseudo device to any disk device. The device to which SP is assigned or redirected must have a UFD of [1,4] with protedtion [RWED,RWED,RWED,RWED]. This UFD contains the queue of files to be output-spooled (the file SPRQUEUE.SYS) and sometimes the spooled files themselves.
- Write-enable the device associated with SP and mount the appropriate volume on this device.

You can set output spooling as either enabled or disabled on an output device. The system rejects any attempt to set output spooling for a device other than a record output device. You can set a terminal spooled only if it is given the NONTRM device characteristic at system generation time. You cannot set a device NOSPOOLED while the device SP is mounted.

The following operations are available with output spooling:

- Queue files to be printed on any spooled terminal or line printer.
- List all files queued for spooling.
- Change the form type.
- Control processing of the queue (for example, abort a file's output).
- Modify the current status or attributes of a file that is queued for printing.

The priority assigned to each queue entry determines the order in which files are printed (unless you have specified the /AFTER qualifier or /AF switch). The priority is a decimal number in the range 1 through 250. Priority values assigned to queue entries have no restrictions. If the entry is made by means of the PDS QUEUE OR PRINT commands, the default value is 100. If the entry is made by means of the PRINT$ macro from a task, the default value is the priority value of that task. See the *IAS I/O Operations Reference Manual* for further details about the PRINT$ macro. Files queued with the same priority are processed in first-in/first-out order.

You can choose the type of form on which a file is to be printed by specifying a number from zero to six when the file is queued for printing. Each number corresponds to a type of stationery. A file in the output spool queue is not printed until the form type of the output device matches its specified form type.

The form type value itself resides in the device system table. The system is generated with a form type 0 (zero) for every terminal or line printer. Form types for values zero through six are determined by the system manager. Form type seven is reserved for the spooling of test files to verify the form alignment. Several form types can correspond to the same type of printer paper.

To specify the number of copies of a file to be printed, select a decimal number from 1 through 31. You can also specify whether the file is to be deleted after printing or retained for subsequent use. The /AFTER qualifier or /AF switch allows you to specify that a file is to be printed only after a given time of day.

6.3.2 Input Spooling

Input spooling is provided in IAS for job files submitted from an input device for batch processing. The input spooler is a multiuser task that can service two or more input devices simultaneously. Spooled files are temporarily stored in a file under UFD [1,4] on the SP device. A batch queue entry is made for each job file encountered on any spooled input device.

6.4 System Library Tasks

IAS enables you to provide a library of installation-dependent *system library tasks* for use by the PDS or MCR user. Any task can be installed as a system library task by means of the PDS INSTALL/SYSTEM command (see Chapter 12). This makes the task available to users with PR.SYS privilege.

System library tasks are similar to system tasks. System task names are in the form ...xxx, where xxx is the task name. System library tasks are installed with names in the form $$$xxx (where xxx is the task name). To run system library tasks, enter the task name xxx.

If you issue a command that is not recognized by PDS or MCR, the list of installed tasks is scanned for a name of a task installed as a system library task. If such a name is found, the task is run.

This facility has the following advantages:

- It can provide a local library of installed system tasks, which provides installation-dependent facilities to PDS or MCR users.
- Because the system library tasks are preinstalled, they can be loaded quickly by the system.
- The system library tasks can be run in the system as multiuser tasks.

6.5 Shareable Global Areas

If two or more tasks must use the same area of code or data, they can link to a *shareable global area* (SGA) that contains that code or data. This can significantly reduce the amount of memory used by each task. SGA can consist of libraries, common areas, or installed regions.

The different types of SGA are fully described in the *IAS Executive Facilities Reference Manual.*

The SYSRES SGA contains many commonly used routines (for example, overlay routines and file control services). The HNDLIB SGA contains commonly used handler routines. Both SYSRES and HNDLIB are installed and loaded during system generation because they are needed by the terminal handler and the system disk handler.

To fix an SGA in memory, use the FIX command to fix a dummy task that uses that SGA. (See the *IAS PDS User's Guide* or *IAS MCR User's Guide* for details.) Build the dummy task with the following options, which are described in the *IAS Task Builder Reference Manual.*

For PDS:

```
/NOCHECKPOINT/NOFLOATINGPOINT/FIX/OPTIONS
STACK=10
UNITS=0
SGA=sganame
```

For MCR:

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

The dummy task uses 96 words; this includes the task header and minimum stack size, plus one fixed task list (FTL) entry. The above approach is the most effective way to fix an SGA. fixing the dummy task results in the loading and fixing of the SGA that it uses.

SGAs are fully described in the *IAS Executive Facilities Reference Manual*. Details and examples of how to build SGAs and the tasks that refer to them are in the *IAS Task Builder Reference Manual*.

6.6 Object Module Libraries

You can store commonly used object modules in an object module library. The *IAS PDS User's Guide* and the *RSX11M/11M-PLUS Utilities Manual* describe how to create object module libraries and how to select them for use.

A system object module library, SYSLIB, is provided in the standard distribution. SYSLIB is assumed to be on the pseudo device LB, which can be the system device SY. SYSLIB is automatically searched at link time to resolve any unresolved references in user programs.

6.7 MACRO Libraries

You can store commonly used MACROs in a MACRO library.

A system MACRO library, RSXMAC.SML, is provided in the standard distribution. MACRO definitions called in by the .MCALL assembly directive are taken from RSXMAC.SML.

See the *IAS PDS User's Guide* and the *RSX11M/11M-PLUS Utilities manual* for details of the creation of MACRO libraries.

See the *IAS PDS User's Guide* and the *PDP-11 MACRO-11 Reference Manual* for details on using MACRO libraries.

6.8 Universal Libraries

You can create universal libraries with non-specific contents as, for example, with object libraries (default file type .OLB) and MACRO libraries (default file type .MLB). For details see the *IAS PDS User's Guide* and the *RSX11M/11M-PLUS Utilities Manual*.

7 User Authorization and Privilege

This chapter describes user authorization and privilege for PDS users only. If MCR is the user interface, refer to the *IAS MCR User's Guide* for the equivalent information. This chapter describes the different types of user and task privileges (see Section 7.1), as well as user authorization and accounting. All these functions are performed by means of the USERS command (see Section 7.4).

7.1 Privileges

This section describes the following types of privileges:

1 PDS command privilege (see Section 7.1.1)

2 Task privilege (see Section 7.1.2.)

3 UIC privilege (see Section 7.1.3)

7.1.1 PDS Command Privilege

Your command privilege regulates your ability to issue a command or set of commands. Apart from a set of basic commands available to all PDS users, each command (or set of commands) can be allowed for or withheld from a user depending on a bit setting in a command mask. Batch and interactive PDS commands each have separate command masks. Thus, single users might have different sets of commands available when they use batch or when they use an interactive terminal; for example, compilations could be restricted to batch mode only.

If you do not explicitly specify batch privilege, it is treated as a function of interactive privilege; that is, batch privilege is the same as interactive privilege without PR.RTC.

Set PDS command privilege by specifying the PRI and BPR parameters to the USERS command (see Table 7–4).

NOTE: If you change a user's interactive privilege and do not specify batch privilege, batch privilege is also changed.

Each mask consists of 16 bits. To render the command(s) available, set the corresponding bit to 1. The bits are referred to by symbolic names.

Table 7–1 PDS Command Privilege Classes

Bit	Symbol	Command or Class of Commands
0	PR.FIL	File manipulation facilities
1	PR.RUN	Task manipulation
2	PR.BAS	BASIC
3	PR.COB	COBOL
4	PR.COR	CORAL

Table 7–1 (Cont.) PDS Command Privilege Classes

Bit	Symbol	Command or Class of Commands
5	PR.FOR	FORTRAN
6	PR.LIN	LINK
7	PR.MAC	MACRO
8	PR.SCI	Timesharing system—Privilege to log in at operator console Multiuser system—Privilege to issue SCI commands
9	PR.SUB	SUBMIT to batch
10	PR.MCR	Privilege for MCR mode
11	PR.DEV	Device management
12	PR.DUM	DUMP
13	PR.LIB	LIBRARIAN
14	PR.SYS	System library tasks ($$$xxx)
15	PR.RTC	Real-time commands

Certain commands are available to all logged-in users. These are marked ANY in Table 7–2 and are independent of the command masks.

Table 7–2 shows the privilege required for each PDS command. A few commands require two privileges. A few commands require you to have an associated UIC of [1,1] (and are so marked). The extra (SCI) commands available at the console terminal do not require any extra privileges on a timesharing system. However, to log in at the console, you must have PR.SCI. On a multiuser system, you must have PR.SCI to issue these commands from any terminal.

If you use MCR mode (PR.MCR), you can circumvent many other command privilege restrictions by calling directly the system tasks that perform these functions. However, privileged MCR commands (see the *IAS MCR User's Guide*) are accepted only if you also have PR.RTC privilege or are a "system user" (that is, with a group code of less than 10 octal). Thus, MCR privilege gives you the same rights as all other privileges except RTC and SCI. If you log in at the console terminal, you are automatically real-time (PR.RTC) privileged.

Table 7–2 shows the PDS command privilege masks.

Table 7–2 PDS Command Privilege Mask

Command	PDS Command Privilege(s) Required
ABORT[1] (timesharing)	ANY
ABORT/REALTIME	PR.RTC
ALLOCATE[1]	PR.DEV
APPEND	PR.FIL
ASSIGN[1] (timesharing)	PR.RUN
ASSIGN/TASK	PR.RTC
BASIC	PR.BAS
CANCEL	PR.RTC

[1]This parameter applies to timesharing systems only.

Table 7–2 (Cont.) PDS Command Privilege Mask

Command	PDS Command Privilege(s) Required
COBOL	PR.COB
CONTINUE[1] (timesharing)	ANY
CONTINUE/MESSAGE	PR.RTC
CONTINUE/REALTIME	PR.RTC
COPY	PR.FIL
CORAL	PR.COR
CREATE	PR.FIL
DCL	ANY
DEALLOCATE[1]	PR.DEV
DEASSIGN[1]	PR.RUN
DEASSIGN/TASK	PR.RTC
DELETE	PR.FIL
DIRECTORY	PR.FIL
DISABLE	PR.RTC
DUMP	PR.DUM
EDIT	PR.FIL
ENABLE	PR.RTC
$EOJ	N/A
FIX	PR.RTC
FORTRAN	PR.FOR
GOTO	ANY
HELP	ANY
IDENTIFY	ANY
INITIALIZE	PR.DEV
INSTALL	PR.RTC
$JOB	N/A
LIBRARIAN	PR.LIB
LINK	PR.LIN
LOGIN	N/A
LOGOUT	N/A
MACRO	PR.MAC
MCR	PR.MCR
MERGE	PR.FIL
MESSAGE	ANY
MOUNT	PR.DEV
ON	ANY
PRINT	PR.FIL

[1]This parameter applies to timesharing systems only.

Table 7–2 (Cont.) PDS Command Privilege Mask

Command	PDS Command Privilege(s) Required
QUEUE	PR.FIL
REMOVE	PR.RTC
RENAME	PR.FIL
RUN[1] (timesharing)	PR.RUN
RUN (real-time)	PR.RUN PR.RTC
SET BOOTSTRAP	PR.DEV
SET DEFAULT[1]	ANY
SET END_OF_FILE	PR.FIL
SET [NO]QUIET	ANY
SET PASSWORD	ANY
SET PRINTING	ANY
SET PRIORITY	PR.RTC
SET PROTECTION	PR.FIL
SET REAL_TIME_CONTROL[1]	PR.RTC
SET UIC	PR.RTC
SET TERMINAL attribute	ANY
SET TERMINAL: (TTm,...,TTn) attribute	[1,1]
SHOW CLI[1]	ANY
SHOW CLOCK_QUEUE	ANY
SHOW DAYTIME	ANY
SHOW DEFAULT[1]	ANY
SHOW DEVICES	ANY
SHOW GLOBAL_AREAS	ANY
SHOW IO_QUEUES	ANY
SHOW LUNS	PR.RTC
SHOW MEMORY	ANY
SHOW PARTITIONS	ANY
SHOW SHAREABLE_GLOBAL_AREAS	ANY
SHOW SWITCH_REGISTERS	ANY
SHOW STATUS	ANY
SHOW TASKS	ANY
SORT	PR.FIL
STOP	ANY
SUBMIT	PR.SUB
TRUNCATE	PR.FIL

[1]This parameter applies to timesharing systems only.

Table 7–2 (Cont.) PDS Command Privilege Mask

Command	PDS Command Privilege(s) Required
TYPE	PR.FIL
UNFIX	PR.RTC
UNLOCK	PR.FIL
USERS	[1,1]
VERIFY	PR.DEV

7.1.2 Timesharing Task Privilege

The system manager can restrict the type of task that each timesharing user can run. By appropriate setting of bits in the task privilege masks, the user can be granted the following rights:

1 To run tasks that make requests to the timesharing control services. (See Section 7.1.2.1.)

2 To run automatically installed executive-privileged tasks (mapped onto SCOM by virtue of being linked with the /PR option) and to install privileged tasks.

3 To run tasks that issue real-time privileged directives (as described in the *IAS System Directives Reference Manual*).

4 To run tasks that issue memory management directives (as described in the *IAS System Directives Reference Manual*).

To set the bits in the task privilege mask, specify the TP1 parameters to the USERS command (see Table 7–3).

This privilege control is available on timesharing systems. On multiuser or real-time systems, these privileges are given to all users (except for the first one, which would be useless except on a timesharing system). Real-time tasks running on a timesharing system are automatically given privileges (3) and (4).

The privilege controls apply only to tasks run from a file, using the RUN *filename* command. They do not affect system tasks or system library tasks that are installed by and under the control of the system manager.

7.1.2.1 Multitasking

Typically, you need never have more than two concurrently operating tasks, which include a CLI (probably PDS) and a user program. However, in the following two cases, you must be able to run more than two tasks concurrently:

- When you run multitasking applications (for example, if you are going to write a task that performs functions by means of separate subtasks).
- If you write your own CLI.

Each user is allocated a value for the maximum number of timesharing tasks that can run concurrently. The default value is three timesharing tasks, which allows for an additional nonsuspendable task while another task is suspended. The value is assigned to the user by means of the MTS parameter in the USERS command (see Section 7.4.1). However, a further limit is imposed when the CLI is allocated to the terminal by means of the ALLOCATE/TERMINAL command (see Chapter 12). In this case, the default is 1 for user CLIs and 255 for the default CLI (PDS).

CLI privileges are specified in the INSTALL/CLI command (see Chapter 12). Each task initiated by a CLI or other timesharing task is given TCP privileges by the initiator. No task is allowed to have more privilege than its initiator.

7.1.2.2 Task Privilege Masks

The task privilege mask consists of 24 bits that are logically split into two groups known as TP1 and TP2.

- TP1 comprises eight bits (of which only four bits are currently defined).
- TP2 comprises 16 bits.

The execution of every timesharing task is subject to the restrictions imposed by the task privilege masks. These masks are set up for tasks in three different ways, depending on the type of task:

- CLI tasks—A CLI is assigned its mask when it is installed. Use the PRIV:n qualifier to the INSTALL/CLI command to pass a 20-bit mask. The first four bits correspond to TP1 and the last 16 bits correspond to TP2. If all privileges are required, specify PRIV:ALL.
- User tasks run from PDS—All users are assigned values for TP1 and TP2 in their user profile entries. (See Section 7.2.6.) When PDS runs a task on behalf of the user, these masks are applied to the task. In certain situations (for example, LINK/CROSS), PDS temporarily applies additional mask bits.
- Subtasks run from user tasks or user-written CLIs—The owner task is responsible for defining TP1 and TP2 when it runs a subtask. This mechanism is described fully in the *IAS Guide to Writing a Command Language Interpreter*.

Table 7–3 gives TP1 privilege mask definitions.

Table 7–3 TP1 Privilege Mask Definitions

Bit	Symbol	Definition
0	JP.PI	Set this bit if the task makes I/O requests on TCP. The only user tasks that should use TCP are the ones using TCS facilities (see the *IAS Guide to Writing a Command Language Interpreter*). Use of TCP is further restricted by the contents of the second mask, TP2.
1	JP.PD	Set this bit if the task executes real-time, privileged directives (see the *IAS System Directives Reference Manual*).
2	JP.PT	Set this bit if the task is run by means of an auto-install (for example, using the PDS> RUN filename command), and if it is also executive-privileged. The mask prevents users from writing and running their own executive-privileged tasks. Note that setting this bit does not affect the running of previously installed executive-privileged tasks.
3	JP.PP	Set this bit if the task executes memory-management directives or tasks using SGAs that contain memory-resident overlays (see the *IAS System Directives Reference Manual*).

Digital does not support direct user access to any of the TP2 bits. Access by user tasks is restricted to indirect support by means of TCS. The following bits are relevant to the corresponding TCS usage:

Relevant Bits	TCS Usage
1, 4, 7	Subtask initiation and control (PR.TSK, PR.JNC, PR.TEV).
7	Event recognition (PR.TEV).
8	Task chaining and timesharing send/receive (PR.CHN).
14	Ctrl/C trapping (PR.CTC).

If you run a task that performs these actions, ensure that the task TP1 and TP2 privilege masks are set appropriately. See the *IAS Guide to Writing a Command Language Interpreter*.

All other bits are solely for the use of system tasks (such as PDS) supplied by Digital. PDS needs all bits set; thus, when PDS is installed, set the privilege mask to ALL (that is, TP1=17, TP2=177777).

7.1.3 System UIC Privilege

Some system facilities are restricted to system UIC privileged users (or those with either PR.RTC privilege or with a UIC group code of less than 10 octal). This privilege enables users to perform the functions described in the privileged commands chapter of the *IAS MCR User's Guide*. These users can also perform the following functions:

- Run tasks whose I/O requests can break through an attach (see the *IAS Device Handlers Reference Manual*). Typically, when a device is attached, no other user can perform I/O on that device at any level (including QIO). A task running for a system UIC privileged user can always violate this restriction.
- Have system access to all files if the group code is less than 10 octal.

7.2 Authorizing Users

You can allocate system resources to users according to their ability and requirements. The range of facilities available to each system user is controlled by authorization.

IAS system usage in PDS interactive and batch modes is controlled by user authorization information held in the system user profile file (UPF) named [1,100]PDSUPF.DAT.

NOTE: Make entries in the UPF by specifying the USERS command.

Sections 7.2.1 through 7.2.6 apply to DCL (PDS, PDX, or SCI) users only.

7.2.1 User Profile File

The user profile file (UPF) contains information pertinent to each authorized PDS user on a system where the user interface is DCL. The IAS distribution kit includes a UPF for system and SCI users. To recreate a UPF, enter the following command:

```
PDS> RUN [11,1]CREUPF
```

This creates a UPF (that is, the file [1,100]PDSUPF.DAT) capable of holding 64 users and containing two system users: SCITERMINAL and SYSTEM. If you need a larger UPF, perform either of the following operations:

- Edit the file [11,100]CRETKB.CMD to increase the MAXUSE value, then rebuild CREUPF.
- Run the extend UPF task EXTUPF (see Section 7.2.1.1).

When you allocate a unique user name for a PDS user, an entry is made in the UPF. When this happens, you can use the USERS command as necessary to modify the authorization information. This updates the entries in the UPF. To modify the authorization information (see Section 7.4), you must specify operation name and user name. The operation name indicates which maintenance function is to be performed (for example, UPDATE or REMOVE).

When you create a new user, the default values are taken if you omit any of the parameters. The system manager can alter these default values with the USERS DEFAULT command. You can examine the current setting of the defaults by specifying the command USERS EXAMINE DEFAULT.

NOTE: The system manager should keep all USERS commands in a command file. Use this file to recreate all UPF records if the UPF becomes corrupted.

The format of the user profile file is given in Appendix A.

7.2.1.1 Extending the UPF

You can extend an existing UPF to accommodate more users by running the task EXTUPF as follows:

```
PDS> RUN [11,1]EXTUPF

PLEASE SPECIFY THE TOTAL NUMBER OF USERS REQUIRED:
```

Enter the total number of users required in response to the prompt. The task extends the UPF to hold the number specified (up to the next multiple of eight). If the UPF is already sufficiently large for the number of users specified, an error message is displayed and the total number is again requested. If you do not require any further extension, press [Return] and the task exits.

7.2.2 User Names

A user name is a unique alphanumeric string of one to twelve characters that identifies a user to the system.

When a user is authorized, the system manager assigns that user a unique user name. The system manager then registers the user name with the system by means of an entry in the UPF.

7.2.3 Passwords

A password is an additional security measure that prevents unauthorized users from gaining access to the system. A password is an alphanumeric string of from one to twelve characters.

The system manager assigns each user an interactive password and can optionally assign each user a batch password. Thereafter, each user can change this password (see the SET PASSWORD command in the *IAS PDS User's guide*). The default batch password for all users is no password.

7.2.4 User Identification Codes

A User Identification Code (UIC) must be associated with each PDS user name. The UIC is used for five purposes:

1 It is the UIC under which all the user tasks run.

2 It is the default owner code of all user-created files.

3 It governs the user's access rights to files.

4 In some cases (that is, if it is a system user), it gives the user extra command privileges, particularly in MCR mode.

5 It is the user default User File Directory (UFD).

It is not the user default UFD if the user changes it by using the PDS SET DEFAULT command (see Section 7.2.6.)

The UIC consists of a group code and member code. The User Identification Code does not have to be unique. The system allows more than one user to be authorized with the same UIC.

7.2.5 Default Devices

On a timesharing system, a default device is associated with each user. Input and output from timesharing tasks to the system device always goes to the default device. The bootstrapped system device is specified in the default record of the UPF and is used if you do not specify a default device when you authorize a user. The user can alter the allocated default device for the duration of a session at any time by using the SET DEFAULT command (see the IAS PDS User's Guide). The default device assignments during a session do not replace that recorded in the UPF and therefore revert to the original default assignment when the user begins a new session.

To find out what your current default devices are, specify the PDS SHOW DEFAULT command.

The SET DEFAULT facility is not available on real-time and multi-user systems. However, if most user files are held on a device other than the bootstrapped system device, you can redirect SY: to this device after executing the task SYD. This task changes the disk indicator for installed tasks and shareable global areas to the actual device from which they are installed, that is, the booted system device.

NOTE: Do not save the system after SYD has executed.

To redirect SY:, specify the following commands each time the system is bootstrapped:

```
MCR>RUN [11,1]SYD
MCR>DMO SY0:
MCR>RED ddnn:=SY
MCR>MOU bootstrapped system device
MCR>MOU ddnn:
```

Where:

- ddnn: = Device specification for your device.

7.2.6 Default User File Directories

Each user is allocated a default User File Directory (UFD). The user's initial UFD is identical in value to the UIC associated with the assigned user name.

On a timesharing system, the user can, however, change the default UFD during a session at any time by using the SET command (see the IAS PDS User's Guide). As for the default device assignments, the user-set default UFD is only in force for the duration of the session, reverting to the original value when the user begins a new session.

PDS users can find out what their current default UFD is by specifying the PDS SHOW DEFAULT command (see the IAS PDS User's Guide).

7.3 PDS ACCOUNTING

PDS account reporting is achieved through the EXAMINE option of the USERS command. The system manager can inspect any one of any group of user's accounting information. The accounting information is updated every time a user logs out of the system.

You can examine any of the three following categories of accounting information:

1. The total connect time used (that is, the time between login and logout) for the current accounting period.
2. The total system utilization for the current accounting period.
3. All information comprising fixed defaults, attributes and limits.

If all PDS accounting information is required, the following is listed:

1. User name.
2. UIC.
3. Default device.
4. Number of users currently logged in under this user name.
5. Connect time limit (in minutes) for the accounting period (timesharing systems only).
6. Connect time limit (in minutes) for the session.
7. System utilization limit (in kilo-core-ticks; that is, time in ticks multiplied by the amount of memory used) for the accounting period (timesharing systems only).
8. System utilization limit for the session (timesharing systems only).
9. Task run time limit (in ticks) (timesharing systems only).
10. Total connect time used (in minutes) for the current accounting period (timesharing systems only).
11. Total system utilization used System utilization (in kilo-core-ticks) for the current accounting period (timesharing systems only).
12. Interactive privilege.
13. Batch privilege.
14. Task level privilege (timesharing systems only).

15 Maximum number of concurrent timesharing tasks.

NOTE: Per-session and task run-time limits are not currently enforced, and are provided only to allow for future extensions to the system.

7.4 USERS Command

The USERS command enables the system manager to examine or modify the User Profile File (UPF) (see Section 7.2.1). This command is specified for several different operations, as follows:

```
USERS operation
```

Where operation is:

- EXAMINE = To examine an existing user profile.
- NEW = To create a new user.
- REMOVE = To remove an existing user.
- UPDATE = To update the information on an existing user.
- DEFAULT = To update default parameters within the UPF.

SCITERMINAL is the user name for the special entry in the UPF for auto-login at the operator console. This is a fixed entry and cannot be modified. The only permitted modification to SCITERMINAL is to set to zero its total connect time, total utilization time, and logged-in user count by specifying USERS UPDATE/ZERO SCITERMINAL or USERS UPDATE/ZERO ALL (see Format 4).

Format 1:

```
PDS> USERS EXAMINE[/qualifiers] parameter
```

Where:

- /qualifiers are as follows:
  - /ALL = For all informaton. This is the default.
  - /CONNECT = For total connect time only.
  - /OUTPUT:filespec = The output from the command is to be written to the specified file.
  - /PRINT = The output from the command is printed (equivalent to /OUTPUT:CL:)
  - /UTILIZATION = For total system utilization only.
- parameter is one of the following:
  - username = Is a 1 to 12 character alphanumeric user name.
  - uic = Is one of the following:

    [x,y] = For all users with group code x and member code y.
    [x,*] = For all users with group code x and any member code.
    [*,y] = For all users with any group code and member code y.
    [*,*] = For all users.
    ALL = For all users.
    DEFAULT = For the default profile.

Format 2:

```
PDS> USERS NEW username [(parameters)]
```

Where:

- username = A 1 to 12 character alphanumeric name.
- parameters = Described in Table 7–4.

Format 3:

```
PDS> USERS REMOVE username
```

Where:

- username = A 1 to 12 character alphanumeric user name.

Format 4:

```
PDS> USERS UPDATE[/ZERO] username!ALL!DEFAULT (parameters)
```

Where:

- /ZERO = Indicates that the total connect time, total utilization time, and logged in user count for the specified profile(s) are to be set to zero.
- username = A 1-12 character alphanumeric name identifying a user. If you do not specify a user name, you can specify either ALL or DEFAULT.
- ALL = Indicates that the command applies to all users.
- DEFAULT = Indicates that the default profile is to be updated.
- parameters = Described in Table 7–4.

Format 5:

```
PDS> USER DEFAULT [(parameters)]
```

Where:

- parameters = Are described in Table 7–4.

Examples:

Example 1:

```
PDS> USERS UPDATE OLSEN (PRI:17777,DEV:DU0:)
```

Updates the UPF entry for the user named OLSEN with a new privilege and default device.

Example 2:

```
PDS> USE NEW, DAVIES (UIC:[200,30],PAS:BLOGGS,PRI:17777)
```

Creates a new user name DAVIES with a UIC of [200,30] and a password BLOGGS.

Example 3:

```
PDS> USER REMOVE WADE
```

Removes user name WADE from the UPF.

Example 4:

```
PDS> USERS UPDATE DEFAULT (DEV:DU0:)
```

Changes the default for a user's default device to DU0:.

7.4.1 Parameters

Parameter names, meanings and values are defined in Table 7–4.

Table 7–4 USER Command Parameter Definitions and Values

Parameter Name.	Meaning	Values	Distributed Default
UIC:	User Identification Code.	[m,n], where m and n are octal values in the range 1 to 377.	[200,200]
PAS:	Interactive password.	1 to 6 alphanumeric characters.	DEFPAS
PRI:	Interactive command privilege. (See Table 7–1.)	Octal number from 0 to 177777	0
RCA:[2]	Connect time limit for the accounting period.	Number of minutes	Maximum[1]
RCS:[2]	Connect time limit for the current session. This parameter is not enforced.	Number of minutes.	Maximum[1]
RUA:[2]	System utilization limit for the accounting period.	Number of ticks.	Maximum[1]
RUS:[2]	System utilization limit for the current session. This parameter is not enforced.	Number of ticks.	Maximum[1]
RTT:[2]	Task time limit. This parameter is not enforced.	Number of ticks.	Maximum[1]
DEV:[2]	Default device.	Device name.	SY0:
BPR:	Batch command privilege. (See Table 7–1.)	Octal number from 0 to 177777.	0
BPW:	Batch password.	1 to 6 alphanumeric characters.	None
MTS:	Maximum number of concurrently active or suspended tasks.	Decimal number from 0 to 255.	3
TP1:[2]	4-bit task privilege mask. See Section 7.1.2.2 for a full description.	Octal number from 0 to 17.	0

[1]When maximum appears in the Distributed Default column, the distributed default is the maximum value to which this parameter can be set.

[2]This parameter applies to timesharing systems only.

Table 7–4 (Cont.) USER Command Parameter Definitions and Values

Parameter Name.	Meaning	Values	Distributed Default
TP2:[2]	16-bit task privilege mask mask specifying use of TCP facilities. See Section 7.1.2.2 for a full description.	Octal number from 0 to 177777.	0

[2]This parameter applies to timesharing systems only.

Part II Error Handling

8 Core Dump Analyzer

The Core Dump Analyzer (CDA) enables you to analyze the state of the software at the time of a system crash. Throughout this chapter, the term crash refers to either an actual system crash or a crash forced by the operator. CDA obtains the data to analyze from the memory dump of a crashed system. The dump is written on the device specified in the crash module that is linked to the Executive. You must include the crash module at system generation (see the *IAS Installation and System Generation Guide*). You must perform the dump analysis on the same version of the system as that where the crash occurred.

In order to benefit from CDA output, you must be familiar with the *IAS Executive Facilities Reference Manual* and with the appropriate PDP-11 Processor Handbook.

The core dump analyzer can produce a listing that contains the following information:

1. An analysis of the system status at the time the core dump was taken, and an indication of the cause of the system crash.
2. The content of all the central processing unit registers and memory management (KT11) registers.
3. The content of the SCOM data area.
4. An analysis of all tasks resident at the time of the crash, including the interpretation of their headers.
5. An analysis of all system lists.
6. A verification of the system node pool, including a comparison of the nodes allocated with those nodes in use in lists.
7. A complete dump of the system node pool in three output formats (octal, ASCII, and Radix 50).
8. A dump of the address space of any task that has memory allocated at the time of the crash.
9. A selective analysis of the timesharing data area IASCOM.

The information produced by CDA falls into two categories:

1. Information always produced by CDA.
2. Information listed only if the appropriate option switch is included in the command string to CDA.

Standard information is described in Section 8.1. Optional information is described in Section 8.2.

For a more detailed description of IASCOM data structures, see the source file [311,107]IASCOM.MAC.

Appendix B contains a sample CDA listing that is referred to for all examples throughout this chapter.

8.1 Standard Information Produced By CDA

The standard CDA information, the system data, is printed at the beginning of the core dump analysis. See the sample analysis in Appendix B.

8.1.1 Crash Stack

The first page of output from CDA displays the data recorded on the crash stack at the time the dump was taken. The crash stack refers to the data area that is internal to the crash module. It contains the following information:

1 Cause of the crash—The reason for the system crash; for example:

```
RED OR YELLOW STACK VIOLATION
```

If the dump did not occur as a result of a condition recognized by CDA, the following message is printed:

```
MANUAL DUMP BY OPERATOR
```

2 The processor status register (PS), the kernel stack pointer (SP), and the user SP after the crash—If the crash was forced by the Executive, these registers reflect the content of the respective registers after the IOT (I/O trap) instruction that signals the crash has been executed.

3 The program counter (PC) and PS before the crash—If the crash was forced by the Executive, the contents of these two registers are pushed onto the kernel stack by the IOT instruction.

4 R0 through R5.

5 Segmentation status registers 0 through 3. Refer to the appropriate PDP-11 Processor Handbook.

6 All the segmentation registers for user, supervisor, and kernel modes for both I and D space. Refer to the appropriate PDP-11 Processor Handbook.

NOTE: IAS does not use D space.

The information contained in the crash stack is the key to the operation of CDA. In particular, the contents of the kernel Page Address Registers (PARs) enable CDA to convert virtual memory addresses into the physical block and word addresses of the dump medium.

8.1.2 Kernel Stack

The kernel stack of 192 (decimal) words is dumped on the second page of the CDA output. The location of the kernel stack is determined from the symbol .SG.TS in the Executive, which is the top of the kernel stack.

8.1.3 System Communication Area

The System Communication Area (SCOM) data area is dumped on the third and subsequent pages of the CDA output. The contents of the following SCOM items are interpreted:

1 Time of the crash.

2 Name, TI, and ATL address of the currently executing task at the time of the crash (.CRTSK).

3 The following parameters:

   a. Floating point hardware indicator (.FP45X), which contains 1 if the floating point option is included.

   b. Processor type (.PDP11), which contains whatever has been specified in the PDP-11 directive during system generation (for example, 40 for a PDP-11/40).

   c. Memory size in 32 word blocks and converted to K (.MSIZE).

4 The default CLI name (.DFCLI), system event recognition flags (.SERFG), and common event flags (.COMEF).

5

   a. The machine indicator word (.UMR22), which contains two bytes:

      low-order byte—indicates where the system was generated
      high-order byte—indicates where the system was bootstrapped

      The bit settings are the same in each byte, and indicate the following:
      bit 4 (ON.22)—operating in 22-bit mode
      bit 5 (ON.UM)—UMRs have been requested
      bit 6 (ON.70)—system generated for a PDP-11/70
      bit 7 (ON.44)—system generated for a PDP-11/44 or PDP-11/84

   b. The timesharing space word (.TSPAE), which contains the space in the timesharing (T) partition used by the IAS scheduler (in 32 word blocks).

   c. The real memory base address (MOD32) of IASCOM.

6 The ATL node of the task being run down (.IORAT), the STD node of the task being run down (.IORTD), and the PUD of the unit to be run down (.IORPD).

7 The entire SCOM data area, .SG.BC to .SG.EC.

8.2 Optional Output from CDA

The optional output switches are divided into three groups:

1 Executive

2 Utility operations

3 Timesharing Libraries

Tables 8–1, 8–2, and 8–3 summarize the available switches within each group.

Table 8–1 Timesharing Libraries

Switch	Function
/CIT	Interprets and dumps the Command Interpreter Table.
/COM	Interprets and dumps the Communication Region.
/DVT	Interprets and dumps the Device Table.
/JNP	Interprets and checks the Job Node Pool.
/STS	Interprets and dumps System Time Statistics.
/TNP	Interprets and checks the Terminal Node Pool.

Table 8–2 Executive

Switch	Function
/ASQ	Interprets and dumps the AST queues.
/ATL	Interprets and dumps ATL entries.
/CKQ	Interprets and dumps the Clock Queue.
/FTL	Interprets and dumps the Fixed Task List.
/GCD	Interprets and dumps GCD entries.
/HDR	Interprets and dumps individual task headers.
/MCR	Interprets and dumps MCR command buffers.
/POL	Dumps the node pool in octal, ASCII, and Radix-50
/PUD	Interprets and dumps PUD information.
/RRQ	Interprets and dumps the Receive-by-Reference queue.
/SFL	Interprets and dumps the Swap File List.
/SRQ	Interprets and dumps SEND/RECEIVE queues.
/STL	Interprets and dumps the SPAWN TASK list.
/STD	Interprets and dumps STD entries.
/TPD	Interprets and dumps TPD entries.
/UTL	Interprets and dumps the User Task List.
/VFY	Verifies the system node pool.

If you use any of the above timesharing library or executive switches on a multiuser system, the following error message is returned:

```
CDA - 'IASCOM' NOT RESIDENT
```

Table 8–3 Utility Operations

Switch	Function
/ALL	Lists all Executive and Timesharing Libraries[1] information.

[1]Applies to timesharing systems only.

Table 8–3 (Cont.) Utility Operations

Switch	Function
/SGA:sganame	Dumps a shareable global area (resident library, common area or installed region).
/DMP:start:end	Dumps a specified part of memory in Octal, ASCII and Radix-50.
/TASK:taskname	[:start:end][/TI:devicename] Dumps all or part of the task's address space for a task that has memory allocated. The /TI:devicename is used to further identify a copy of a named multiuser task.
/EXE	Lists all Executive information.
/TSL[1]	Lists all Timesharing Library information.

[1]Applies to timesharing systems only.

8.2.1 All Lists (/ALL)

The /ALL switch causes all Executive and Timesharing Libraries information to be printed. Using /ALL is equivalent to using the following switch:

```
/EXE/TSL
```

If you specify the /ALL switch, the information is printed in the following order:

1 IASCOM: Data area

Device table

Command Interpreter
Table (CIT) with
User Terminal Nodes (UTNs)
and User Job Nodes (UJNs)

2 Active Task List (ATL)

3 Task Headers

4 User Task List (UTL)

5 Task Partition Directory (TPD)

6 System Task Directory (STD) alpha table

7 Physical Unit Directory (PUD)

8 MCR buffers

9 Fixed Task List (FTL)

10 Asynchronous System Trap (AST) queue

11 SEND/RECEIVE Queue (SRQ)

12 Receive-by-Reference Queue (RRQ)

13 Spawn Task List (STL)

14 Swap File List (SFL)

15 Clock Queue (CKQ)

16 Global Common Directory (GCD)

17 Verification of system node pool

18 Dump of system node pool

8.2.2 AST Queues (/ASQ)

A dump of the AST queues is included if the /ASQ (or /ALL or /EXE) switch is used.

The /ASQ switch causes all tasks with any entries in their AST queue to be listed.

NOTE: The AST queue also includes completed I/O requests for which the task has not yet been notified.

For each task with ASTs pending, the task name, ATL address (real and virtual) and STD address are output as a heading.

I/O completion entries are output as:

1 AST type(=I/O done), AST entry point, if any.

2 Event flag number specified in the QIO, I/O status block address, I/O status block contents (two words).

3 A dump of the node.

For true AST entries the output is:

1 AST type, entry point.

2 Additional information, if any, which will be pushed onto the user's stack.

3 The task exit status if it is a spawn node.

4 A dump of the node (a spawn node AST is half the length of the other AST nodes).

8.2.3 Active Task List (/ATL)

A dump of the Active Task List is included in CDA output if you specify the /ATL switch. This list is linked and printed in priority order.

Either one or five dummy nodes appear in the ATL according to the mode in which the system is running. The dummy nodes do not represent actual tasks, but they are markers used by the Executive. The real-time null task is always present at the end of the ATL. The other four dummy nodes are present when the IAS scheduler is active. Although these dummy nodes do not represent tasks, they have a priority and status, and CDA gives them names for ease of reference. Table 8–1 lists the dummy nodes.

Table 8–4 ATL Dummy Nodes

Name	Priority	Status	Function
.THATL	220	SUS or TS1	Used by scheduler
.TSS1	100	TS1	Used by scheduler

Table 8–4 (Cont.) ATL Dummy Nodes

Name	Priority	Status	Function
.TSS2	100	TS2	Used by scheduler
.TSNUL	1	IDL	Scheduler null task
.NULL	0	IDL	Real-time null task

The /ATL switch causes all entries in the ATL to be listed and provides interpretation of the following items in the order described:

1 Task name.

2 ATL and STD address.

   ATL address—virtual and real node addresses
   STD address—virtual and real node addresses

3 Requester and load address.

   Requester—name of the task that requested this task
   Load address—real memory address of the task's header

4 Name of the partition in which the task was running.

   The task run priority—tasks under the control of the IAS scheduler are held on the ATL at a priority of 1 until they are run.

   I/O in progress count

   I/O pending count

   Status—the same 3-character codes as those described in the *IAS Executive Facilities Reference Manual*.

   TI assignment—the terminal where the task is being run.

5 Actual I/O pending, and I/O released for swapping.

6 Flags and masks.

   Flags—task's 32 event flags (two words)
   Masks—task's 64 event flag masks (four words)

7 Task flags.

8 Entire ATL node dumped with virtual addresses.

8.2.4 Command Interpreter Table (/CIT)

A dump of the Command Interpreter Table is performed if you include the /CIT (or /ALL or /TSL) switch in the command string. The table header is printed first and contains the following information:

1. The base and end address of the CIT table.
2. The number of entries in use.
3. The maximum number of entries and the size of each table entry.

For each used CIT entry the following is printed:

1. Entry number.
2. The CLI task name and STD node address.
3. Flags byte.
4. Default TCP and Job privileges.
5. User Terminal Node (UTN) list head forward and backward links.
6. A dump of the node.

For each UTN linked to each CIT entry, the following information is printed:

1. The address of the terminal node.
2. Forward and backward links.
3. Function code.
4. Flags byte.
5. Entry address for this terminal in the Device Table.
6. Context byte and context control byte.
7. Terminal responder task UJN address and name if there is one. The terminal responder task is the subtask that has claimed notification of Ctrl/C. If there is no such task, the terminal responder task is the CLI.
8. Terminal CIT entry.
9. CLI task UJN address.
10. CLI task UIC, minutes to task exit, and terminal device name.
11. User name.
12. Default device and UFD.
13. Maximum number of tasks (job node allocation).
14. Forward and backward links for UJNs owned by the terminal.

15 Forward and backward links for buffers owned by the terminal.

16 A dump of all buffers owned by the terminal.

17 Device map table entries in use.

18 Device table address for all entries (if any).

19 LUN map entries in use.

20 LUN and device map index for all entries.

21 Dump of the UTN node.

For each UJN the following information is printed:

1 The UJN address and Job name.

2 The forward and backward links.

3 TCP flags.

4 Terminal node address.

5 Privilege flag.

6 UIC of task.

7 The ATL and STD node addresses.

8 The job initiator address and name.

9 The owner task notification event flag.

10 Reason for task event.

11 Task size (32 word blocks).

12 TCP Privilege mask.

13 Maximum number of subtasks (job node allocation).

14 Name and STD address of chain task.

15 Chain task Command Line byte count and spawn node address.

16 Dump of spawn node (including command line if any).

17 Command line byte count and buffer address.

18 Data return block (DRB) buffer address.

19 Dump of DRB buffer.

20 UTN chain forward and backward pointers.

21 Chain task SEND/RECEIVE queue address.

22 Receive queue address.

23 Number of TCP QIOs pending.

24 Dump of UJN node.

For each buffer the following is printed:

1 Buffer address.
2 Forward and backward links.
3 Owner task's UJN address and name.
4 Chain pointer.
5 The buffer contents in octal, Radix-50 and ASCII.

8.2.5 Clock Queue (/CKQ)

A dump of the clock queue is included in CDA output if you specify the /CKQ (or /ALL or /EXE) switch.

The /CKQ switch causes all entries in the clock queue to be dumped. The format of the output depends on the type of the clock queue entry. For all types, the first line indicates the type and the time in ticks which must elapse after the previous entry before this entry will be dequeued (the Delta time).

The types are:

1. Mark time. The following parameters are supplied:
   a. Task name.
   b. STD address - virtual and real node address.
   c. ATL address - virtual and real node address.
   d. Event flag number.
   e. AST entry point address.
   f. A dump of the node.
2. Timesharing. Only a dump of the node is supplied.
3. Schedule; that is, a request for a task to run at a specified time. The additional information is:
   a. Name of requested task.
   b. STD address of requested task - real and virtual addresses.
   c. Task name of requestor.
   d. STD address of requestor - real and virtual addresses.
   e. Partition specified for task to run in, or blank if no partition specified.
   f. Priority specified for task, or zero if no priority specified.
   g. UIC specified for task to run under, or [0,0] if no UIC specified.
   h. TI for which the task will run.
   i. A dump of the node.

4 Reschedule, that is, a schedule which is to be repeated at a regular interval. The additional information is as for schedule in item 3 above, with the addition of the interval between periodic reschedules, in ticks.

8.2.6 Communications Region (/COM)

A dump of the Communications Region is included if you specify the /COM (or /ALL or /TSL) switch. The following information is printed.

Scheduling Information:

1 Current scheduled task, next task to be loaded and number of active tasks.

2 Idle time, maximum run quantum and quantum constant.

3 Ticks between schedular promotions and ticks till next promotion.

4 Ticks between batch schedules, batch quantum and ticks till next batch schedule.

5 Allocation factor memory size and ticks per memory size.

6 Number of active terminals, and the current batch space.

Maximum parameters:

1 Timesharing jobs and terminals.

2 Task size, and removable volume devices in system.

3 LUNs per job, and mountable volumes per job.

4 Batch space.

Sizes—The sizes in bytes (octal) of a User Job node (UJN), a User Terminal node (UTN) and the buffer.

Partitions—The partitions used for User Jobs and the Executive.

The timesharing priority.

The default CLI task name.

A dump of the IASCOM data area.

8.2.7 Dump of Real Memory (/DMP)

The /DMP switch causes a portion of real memory to be dumped in three notations: octal, ASCII, and Radix 50. The /DMP switch has the following syntax.

```
/DMP:start:end
```

Where:

- start = Starting real memory address (an octal number representing up to 18 bits).
- end = Ending real memory address (an octal number representing up to 18 bits).

You must specify both the start and end addresses. You can specify only one /DMP switch in a line.

8.2.8 Device Table (/DVT)

A dump of the Device Table is printed if you include the /DVT (or /ALL or /TSL) switch in the command string. The table header, printed first, contains the following information:

1 Base and end address of the table.
2 Number of entries and entry size.
3 The pointer to the first entry for a removable volume device.

For each entry the following parameters are printed:

1 Entry number, type (non-removable or removable), the PUD address, and device name.
2 Device usage count.
3 Flags word.
4 Exclusive user UTN address and UTN pointer.
5 Dump of entry.

8.2.9 All of Executive (/EXE)

If the /EXE (or /ALL) switch is included in the command string, CDA causes the following information to be printed:

1 Active Task List (ATL)
2 Task Headers
3 User Task List (UTL)
4 Task Partition Directories (TPD)
5 STD entries
6 Physical Unit Directories (PUDs)
7 MCR command buffers
8 Fixed Task List (FTL)
9 AST queue
10 SEND/RECEIVE Queue (SRQ)
11 Receive-by-Reference Queue
12 Spawn Task List (STL)
13 Swap File List (SFL)
14 Clock Queue (CKQ)
15 Global Common Directory (GCD)
16 Verification of system node pool
17 Dump of system node pool

Including /EXE is equivalent to the following:

```
/ATL/HDR/UTL/TPD/STD/PUD/MCR/FTL/ASQ/SRQ/RRQ/STL/SFL/CKQ/GCD/VFY/POL
```

8.2.10 Fixed Task List (/FTL)

A dump of the fixed task list is included in CDA output if you specify the /FTL (or /ALL or /EXE) switch.

NOTE: All active, fixed tasks are linked and printed by specifying the /ATL switch (see Section 8.2.3). Therefore, fixed tasks that are active appear on the ATL scan; fixed tasks that are inactive appear on the FTL scan.

The /FTL switch causes all entries in the FTL to be listed and provides interpretation of the following items in the order described:

1. Task name.
2. FTL address—virtual and real node addresses.
3. STD address—virtual and real node addresses.
4. Load address—real memory address of the task's header.
5. Name of the partition in which the task was installed.
6. The task run priority.
7. I/O in progress count.
8. I/O pending count.
9. Status—the same 3-character codes as those described in the *IAS Executive Facilities Reference Manual*.
10. TI assignment—the terminal where the task was being run.
11. Flag word—A.TF.
12. Flags—task's 32 event flags (two words).
13. Masks—task's 64 event flag masks (four words).
14. Entire FTL node dumped with virtual addresses.

8.2.11 Global Common Directory (/GCD)

A dump of the global common directory is included in CDA output if you include the /GCD (or /ALL or /EXE) switch in the command string.

The GCD contains one entry for each SGA and dynamic region in the system, together with an entry for each task with a pure area. The formats of the entries are identical, except that the SGA nodes give the name of the SGA, and the pure area nodes give the name of the task. Note that dynamic regions have blank names. The following information is displayed for each GCD when you specify the /GCD switch.

1. Name of the SGA, or task if the node is for the pure area of a task.
2. Base address of the SGA/pure area.
3. Size—Size in 32-word blocks of the common block, library, or pure area.

4 TI assignment.

5 Name of the partition in which the SGA/pure area is installed.

6 Status of the global area using the same 3-character codes as those described in the *IAS Executive Facilities Reference Manual* with the addition of NUL to indicate not in use or not loaded.

7 Owning UIC.

8 Active reference count and installed reference count.

9 Starting APR and swap file index.

10 Load device and disk address.

11 Protection and creation data.

12 Flags word.

13 Dump of the node.

8.2.12 Task Header (/HDR)

A dump of the task headers for each resident task is included in CDA output if you include the /HDR (or /ALL or /EXE) switch in the command string.

Interpretation of the following items is provided for each task header:

1 The task name.

2 A dump of the header.

3 Header address—real address of the task header.

4 ATL address for the task.

5 H.CR1—context buffer reference point 1.

6 User Page Descriptor Registers (PDRs).

7 User Page Address Registers (PARs).

8 Contents of PS, PC, R0 through R5, and SP at the last context switch (these values do not necessarily reflect live register values of a running task).

9 Initial PS, PC and SP.

10 Size of the header in 32-word blocks.

11 Directive status word (DSW)—task's virtual zero.

12 Number of LUNs assigned to the task.

13 A list of all regions and attachment descriptors.

14 Logical Unit Table (LUT)—the device and unit address for each entry in the LUT. If there is a window open for the entry, the following additional items are dumped:

- From the window: W.CTL, W.VBN, W.FCB
- From the FDB: F.FNUM, F.FSEQ, F.STAT, F.NACS, F.NLCK

8.2.13 Job Node Pool (/JNP)

A scan of the User Job Node pool is performed if you include the /JNP (or /ALL or /TSL) switch in the command string.

Each UJN is tested to check that it is correctly clear. Warning messages are printed if the job node has not been correctly cleared.

Finally, a count of the number of free UJNs scanned is printed.

8.2.14 MCR Command Buffers (/MCR)

A dump of the MCR command buffer is included in CDA output if the /MCR (or /ALL) switch is used in the command string.

The /MCR switch causes all MCR command buffers to be dumped and provides interpretation of the following items for each buffer:

1. Node address—virtual address of the node.
2. Name of the MCR task requested.
3. TI assignment—terminal initiating the task.
4. Length of the buffer.
5. The first 40 characters in the buffer converted to ASCII.

In a timesharing system, no MCR Command Buffers should be allocated.

8.2.15 Dump of System Node Pool (/POL)

Inclusion of the /POL (or /ALL or /EXE) switch causes the entire system node pool to be dumped with virtual and real addresses. Each node (8-word block) is dumped in three formats: octal, ASCII, and Radix 50. In addition, if the node is in use (allocated), an asterisk (*) is printed at the left hand margin.

8.2.16 PUDs and I/O Request Queues (/PUD)

Inclusion of the /PUD (or /ALL or /EXE) switch causes all Physical Unit Directories (PUDs) to be dumped. If an I/O request is queued for the device, the I/O request node is dumped after the associated PUD.

The following items are interpreted for each PUD entry:

1. Device type and unit number.
2. Flag byte.
3. Name of the task or terminal attached to the device, if any.
4. Device to which this one has been redirected, or itself if it has not been redirected.
5. Handler task name (if resident).
6. Count of express requests in the queue (EXP CNT).

7 U.CH—characteristics word in PUD (CHAR).

8 First three characters of the default Ancilliary Control Processor (ACP) task name in Radix-50 (DACP).

9 Name of the ACP task.

10 Terminal flags byte (TERM).

11 A dump of the entire PUD.

The following items are interpreted for the I/O request node:

1 Name of the task generating the I/O request.

2 ATL node address of the requester.

3 Size of the DPB (DPB SZ).

4 Priority of the request.

5 LUN.

6 Event flag number (EFN).

7 Function code—if the function code is one of the more common function codes, it is printed symbolically, for example, IO.WLB. If it is not a common code, IO.XXX is printed.

8 Request type—either normal or express.

9 A dump of the entire I/O request node.

8.2.17 Receive-by-Reference Queue (/RRQ)

A dump of the Receive-by-Reference queue is included in the CDA output if the /RRQ (or /ALL or /EXE) switch is used. The following information is printed for each node.

1 Receiver task name and STD address.

2 Sender task name and STD address.

3 Region name, GCD address.

4 TI assignment, flags byte.

5 Event flag mask and event flag mask address.

6 Window offset and length.

7 Data buffer.

8 A dump of the node.

8.2.18 Swap File List (/SFL)

A dump of the swap file list is included in the CDA output if you specify the /SFL (or /ALL or /EXE) switch.

The start and end addresses (virtual and real) of the swap bitmap are printed and the following is printed for each swap file:

1 The file identification (file number and sequence number).

2 The swap file starting logical block number (LBN) and the device that contains the file.
3 The length of the file in blocks, and the number of blocks allocated.
4 The flags byte.
5 A dump of the node.
6 The file bitmap start address (virtual and real) and length.
7 A dump of the file bitmap in which a setting of 0 means in use and 1 means free.

8.2.19 Dump of an SGA (/SGA)

The /SGA switch is used to dump a shareable global area (library, common area, or region). The SGA must have been resident in memory.

The /SGA switch has the following syntax, with brackets indicating the optional portion:

```
/SGA:sganame[:start:end]
```

Where:

- sganame = Name of the SGA to be dumped.
- start = Starting virtual address in octal of the portion of the SGA to be dumped.
- end = Ending virtual address in octal of the portion of the SGA to be dumped.

Default: if the start and end addresses are omitted the entire SGA is dumped.

For a position-independent SGA, addresses are assumed to start at 0, both in the dump and in calculating the start and end values. For a non-position-independent SGA, addresses correspond to those used when the task was built, for example, and SGA mapped through APR3 would start at 60000.

The following information is printed for the SGA:

1 SGA name.
2 Dump of the GCD node for the SGA (if it is in memory), in octal, ASCII and Radix-50.

This switch can only appear once in a CDA command string.

8.2.20 SEND/RECEIVE Queues (/SRQ)

Inclusion of the /SRQ (or /ALL or /EXE) switch causes all the SEND/RECEIVE queues to be listed and provides interpretation of the following items for each queue:

1 Task name of the receiver.
2 Task name of the sender.
3 TI—terminal of receiver or sender task.
4 Priority of the send.
5 Data block contents.

8.2.21 System Task Directory Entries (/STD)

Inclusion of the /STD (or /ALL or /EXE) switch causes all the entries in the System Task Directory (STD) to be dumped. The STD is printed in the order given by the alpha table (that is, alphabetical order), and not in the order that the STD is linked. At the beginning, the alpha table and the number of entries are printed. CDA provides interpretation of the following items for each STD entry:

1. Name of the task (TASK).
2. Name of the partition in which the task is installed (PAR).
3. STD flag word (FLAGS).
4. Priority at which the task is installed (PRI).
5. Disk from which the task is to be loaded (DISK).
6. Size of the task (SIZE).
7. Number of active versions of the task (AV).
8. Pool limit specified in 8-word blocks (LIM).
9. Pool utilization specified in 8-word blocks (UTIL).
10. SEND/RECEIVE listhead (SRQ LISTHEAD).
11. A dump of the entire STD node.

8.2.22 Spawn Task List (/STL)

A dump of the spawn task list is included in the CDA output if you specify the /STL (or /ALL or /EXE) switch. The following information is printed for each node.

1. Requester task name, ATL address and TI assignment (none if requested by TCP).
2. Spawn task name, ATL address and TI assignment.
3. AST routine address (if any).
4. Exit Status Block address, event flag.
5. Flag byte.
6. Command line, if any, and command line length.
7. A dump of the node.

If the ATL address of the requesting task is zero, the message NO REQUESTOR TASK NOTIFICATION is printed instead of list item 1, above.

8.2.23 System Time Statistics (/STS)

Inclusion of the /STS (or /ALL or /TSL) switch causes the System Time Statistics to be produced, as follows:

1 Elapsed time since system start-up.

2 Elapsed time since start of recording.

3 Null job and/or batch executing.

4 Total time users jobs executing.

5 Total time without timesharing because of swapping.

6 Total time given to batch.

8.2.24 Dumps of Task Address Space (/TAS)

Inclusion of the /TAS switch causes the RW address space of a task that has memory allocated to it at the time of the crash to be dumped. The task must have been either active or fixed in memory. Each task area is dumped in three formats: octal, ASCII, and Radix 50.

The /TAS switch has the following syntax with brackets indicating the optional portion:

```
/TAS:taskname[:start:end][/TI:devicename]
```

Where:

- taskname = Name of the task to be dumped.
- start = Starting virtual address (in octal) for the portion of the task to be dumped.
- end = Ending virtual address (in octal) of the portion of the task to be dumped.
- devicename = TI for the required task (optionally followed by a colon).

If the starting and ending addresses are omitted, the entire task is dumped. When this switch is used to dump a multi-user task, only its impure area is dumped.

The switch /TI:devicename is used to identify further a copy of a named multiuser task.

NOTE: Only tasks with entries in the ATL can be dumped using the /TAS switch. SGAs and pure areas of tasks cannot be dumped using /TAS. Use the /SGA switch instead (see Section 8.2.19).

8.2.25 Terminal Node Pool (/TNP)

A scan of the User Terminal Node pool is performed if you include the /TNP (or /ALL or /TSL) switch in the command string.

Each UTN is tested to check that it is clear. Warning messages are printed if the UTN has not been cleared.

Finally, a count of the number of free UTNs scanned is printed.

8.2.26 Task Partition Directory (/TPD)

Inclusion of the /TPD (or /ALL or /EXE) switch causes all entries in the task partition directory to be dumped. Also dumped are any entries in the Memory Required List (MRL) and the Checkpointable Task List (CTL) for each partition. An allocation map is printed for each partition showing the unused area, active and fixed tasks, and shared areas in the partition.

The following items are interpreted for each partition:

1 Partition name.

2 TPD entry address—virtual and real address of TPD entry.

3 Base address—real starting address of the partition.

4 Size of the partition in bytes.

5 TPD flag word.

And then for user and system partitions:

1 Unused areas (holes)—if the partition is system-controlled, the address and size of each unused area is printed.

2 Allocation map—all active or fixed tasks and shared areas in the partition are listed with the real address of the space allocated. One of the following letters is used to indicate the type of task or area:

A—active task
F—fixed task
P—pure area of a task
L—library SGA
C—common area SGA

For Timesharing partitions the MUL listhead is printed, containing:

1 Forward and backward links

2 Space not in MUL

3 Base Hole size

4 Base address

5 Total free space

And then each MUL entry is printed with the following information:

1 Forward and backward links.

2 Active I-O count and flags word.

3 ATL/GCD /FTL node address and name of task loaded in this part of memory.

4 Segment size and hole size above segment.

5 Base address, in 32 word blocks

If any tasks in a partition are checkpointed, the following items are interpreted:

1 Virtual address of the ATL node (NODE ADR)

2 Name of the task

3 Size of the task (SIZE)—S.TZ
4 Run priority of the task (PRI)
5 Checkpoint priority (CP PRI)
6 Status (STS)
7 Checkpoint status (CP STS)
8 TI.
9 A dump of the ATL node

If the partition has any entries in the CTL, the following items are interpreted:

1 CTL node address
2 Name of the task
3 Run priority
4 Status (STS)
5 TI
6 A dump of the CTL node.

8.2.27 Timesharing Libraries (/TSL)

If you include the /TSL switch in the command string, CDA causes the following information to be printed:

1 Communications Region
2 Job Node Pool
3 Terminal Node Pool
4 Device Table
5 Command Interpreter Table
6 System Time Statistics

Including /TSL is equivalent to the following:

```
/COM/JNP/TNP/DVT/CIT/STS
```

8.2.28 User Task List (/UTL)

Inclusion of the /UTL (or /ALL or /EXE) switch causes the User Task List to be dumped. This includes the UTL header containing the number of levels (UTL nodes).

For each UTL the following is printed:

1 Scheduling level.
2 Number of entries in level.
3 Time factor for level.

4 Flags.

5 Task name given by robin pointer.

6 Task name of next task to swap.

7 Task name of next task to load.

8 Dump of the node.

Then, for each entry in the level, the following is printed:

1 Task name, ATL address (virtual and real) and TI assignment.

2 Timesharing flag.

3 Job status and saved status.

4 UTL listhead address (virtual and real) and CPU time in ticks.

5 Quantum and allocation factor.

6 Dump of the ATL node.

8.2.29 Verification of System Node Pool (/VFY)

Inclusion of the /VFY (or /ALL or /EXE) switch causes the system node pool to be verified. The following procedures are used to verify the node pool:

1 The number of nodes (8-word blocks) in use is determined.

2 The total number of nodes in the pool is also calculated with the following formula:

   - Total number of nodes = (Mod32(.STDTA+(.STDTC*2))-.PUDBA/8)
   - .STDTA is the starting address of the STD alpha table.
   - .STDTC is the capacity of the STD. .PUDBA is the beginning address of the PUD.

   This formula provides the total number of nodes available in the pool, that is, those that are not allocated permanently to fixed lists such as the PUD, TPD, and STD alpha table.

3 All the dynamic system lists are scanned to find the nodes that have been picked from the pool. The lists that are searched include the following:

   - Fixed task list
   - I/O request nodes
   - Volume control block
   - File control block
   - Vector connection (..CINT) nodes
   - AST nodes
   - Window nodes
   - AST nodes linked to task headers
   - Active Task List
   - Clock Queue
   - GCD nodes

- MCR Command Buffer
- Memory Required List
- System Task Directory
- SEND/RECEIVE Queue
- Receive-by-Reference queue
- Spawn Task List
- Timesharing nodes
- MUL nodes

As a result of this search, a bit map is constructed to show which nodes are in use in system lists. Although every list is searched, it is still possible to miss some nodes because any privileged task can pick nodes. Nodes in use for privileged tasks are not found when the system lists are searched.

4 The bit map created in step 3 above is scanned and the number of nodes marked as being in use is printed. A bit setting of 0 indicates the node is in use, while a bit setting of 1 indicates that the node is free.

5 The pool is tested for discrepancies. First, the nodes actually allocated, but not found in any list, are dumped. There are always some of these.

6 Next, the nodes found in the lists, but not allocated, are dumped. If there are any of these, they should be examined carefully. They indicate that the current pool allocation does not reflect the status of the nodes in the pool accurately.

7 Finally, the bitmaps representing the nodes allocated and the nodes found in the lists are dumped.

8.3 Operational Information

The operational information required for CDA includes the process of linking the crash modules into the Executive and the command used to call CDA.

8.3.1 Linking the Crash Modules into the Executive

You can select a crash module when the system is configured during the question and answer session at system generation. See the *IAS Installation and System Generation Guide*.

8.3.2 Calling CDA

The Core Dump Analyzer is a task that can execute online with other tasks.

To run CDA, you must mount the dump device as foreign and issue one of the following command sequences:

For PDS:

```
PDS> MOU/FOR ddn: volid
PDS> INS [11,1]CDA
PDS> ASSIGN/TASK:...CDA ddn 3 ! ddn is the dump device name
PDS> RUN[11,1]CDA
CDA>[output filespec][/switches]
```

For MCR:

```
MCR>MOU ddn:/CHA=[FOR]
MCR>INS [11,1]CDA
MCR>REA ...CDA 3 ddn ! ddn is the dump device name
MCR>CDA
CDA>[output filespec][/switches]
```

Where:

1 output filespec = File specification to name the output for the analysis listing. The default file specification is LP0:ANALYSIS.LST. If you omit any or all of the filespec fields, the defaults apply. To inhibit automatic spooling on a crash dump, specify SY: as the output device.

2 switches = Switches summarized in Section 8.2 and described in Sections 8.2.1 to 8.2.29. You can use any combination of switches or you can omit all switches, which causes only the system data to be analyzed.

Terminate CDA by typing Ctrl/Z.

You can also call the commands from a command file. For example:

```
CDA>@command file
```

8.3.3 Examples of CDA Command Strings

The following are examples of CDA command strings:

Example 1:

```
CDA>
```

Result: Only the system data is output to the line printer.

Example 2:

```
CDA>/ALL
```

Result: All the standard and optional output is sent to the line printer.

Example 3:

```
CDA>SY0:POOLDUMP/POL
```

Result: The system node pool information is output to the file SY0:POOLDUMP.LST.

Example 4:

```
CDA>/ATL/HDR
```

Result: The ATL nodes and task header information are output to the line printer for those tasks that were active when the system crashed.

Example 5:

```
CDA>@TTHAND
```

Result: The file TTHAND.CMD contains the command /PUD/TASK:TT.... All the PUDs are dumped followed by a dump of the entire task "TT....".

Example 6:

```
CDA>/TASK:XYZ:0:2000
```

Result: If the task XYZ had memory allocated at the time of the crash, its virtual address space from 0 to 2000 is output to the line printer. If no memory was allocated for XYZ a message is printed instead.

Example 7:

```
CDA>/DMP:44000:44300
```

Result: Real memory from 44000 to 44300 is dumped.

8.4 CDA Error Messages

The following messages are reported when a CDA error is detected.

CDA, DEVICE READ ERROR

Explanation: Input device is not ready or is otherwise unavailable.

Suggested User Action: Check status of device. Call field service if problem persists.

CDA, PLEASE MOUNT INPUT DEVICE AS FOREIGN

Explanation: Input device not mounted as a foreign volume.

Suggested User Action: Mount the input device as foreign.

CDA, ERROR ON OUTPUT FILE

Explanation: Error on output file, which might be due to hardware problems or lack of space.

Suggested User Action: Look for lack of space on disk or hardware problem and act accordingly.

CDA, ILLEGAL SWITCH

Explanation: An unrecognized switch was found in the string.

Suggested User Action: Type in the command again using the correct command string.

CDA, DUMP ABORTED—KERNEL PARs CLOBBERED

Explanation: Either the crash corrupted PARs or the format of the crash stack on the dump medium is incorrect.

Suggested User Action: Try to obtain another crash dump representing the same system error.

CDA, FAILED TO READ GML COMMAND BUFFER

Explanation: Error reading command line or indirect command file.

Suggested User Action: Try typing command again or check that command file exists.

CDA, SWITCH LIST SYNTAX ERROR

Explanation: Mistake occurred in the switch list.

Suggested User Action: Correct the error and retype the command.

CDA, FILESPEC SYNTAX ERROR

Explanation: A mistake occurred in the file specification.

Suggested User Action: Correct the error and retype the command.

CDA, OUTPUT FILE OPEN FAILED

Explanation: Unable to open output file.

Suggested User Action: Check that device not write locked, UFD exists, necessary privileges exist, and so on.

CDA, 'IASCOM' NOT RESIDENT

Explanation: Request to analyze timesharing libraries when libraries not resident at time of crash.

Suggested User Action: For a multiuser or real-time system, the error dictates that one of the timesharing library switches was used, so this error can be ignored. Otherwise, examine the GCD dump to determine why IASCOM is not in memory.

8.5 Input to CDA

The usual source for input to CDA is the dump produced by one of the IAS Executive crash modules that can be linked in when the Executive is built (see Section 8.3).

Inclusion of a crash module does not necessitate a rebuild of CDA.

It is also possible that you may want to crash the system in order to study the dump produced, or because the system has failed but has not actually crashed (for example, there is no response from any terminals). To force a crash, perform the following steps:

1 If the system is executing in User Mode, set Kernel location 100 to a value of 1, as described in the appropriate PDP-11 Processor Handbook.

2 If the system is executing in Kernel Mode, set the Kernel PC to a value of 44, as described in the appropriate PDP-11 Processor Handbook. Kernel location 44 contains a jump into the crash module.

If the console has no switches, use the MCR OPE command or the SCI EXAMINE command to force a crash.

For example:

```
SCI>>OPE 100/KNL
00000100 004252/1 ESC
```

or:

```
SCI>EXAMINE/KERNEL 100
00000100 004252/1 ESC
```

WARNING: Causing an active, running system to crash can have serious consequences. It always causes a complete loss of all data in memory. It can cause corruption of disk file structures. Use extreme caution.

9 Error Logging

IAS performs hardware error logging for both memory and device errors. The Executive logs memory errors, which can be either main memory or cache memory parity errors. Depending upon the device type, either the Executive or the MSCP error log server log device errors.

Two separate error logs exist. One contains items logged by the Executive, and the other contains items logged by T/MSCP device handler tasks. The system manager can use the output from the error logging analysis task to determine the reliability of memory and devices used in the IAS system.

Handlers accumulate error statistics for the following devices:

1 Disks

2 DECtapes

3 Magnetic tapes

For further information on the handling of parity errors, see Chapter 10.

The report produced by the executive error logger can contain itemized error statistics with optional summary information, or it might contain only the summary information. Additionally, the system manager can select the time interval that a report is to encompass, and can indicate that a report is to include only memory parity errors or device errors. Further, if you want a report on device errors you can specify the device type, unit, or volume for which the report is required.

Appendix C contains sample reports for disk and tape errors.

9.1 Functional Description of the Executive Error Logger

Error logging consists of two distinct functions:

1 Gathering of information pertinent to the errors that occur.

2 Error analysis and the creating of a list file.

These functions are performed by three tasks: ERRLOG, PSE, and SYE.

ERRLOG gathers volatile information when a device error occurs. It places this information in a temporary file named ERR.TMP under UFD [1,6] on the system device or a user-specified device. To specify a device other than SY, type one of the following commands:

```
SCI> ASSIGN/TASK:ERRLOG ddnn: 4
```

or:

```
MCR>REA ERRLOG 4 ddnn:
```

Where:

ddnn: = Device on which ERR.TMP is to be placed.

The error log device must contain the UFD [1,6], where all error log files are placed. You can create the directory with one of the following commands:

```
SCI> CREATE/DIRECTORY dev:[1,6]/PRO:(SY:RWED,OW:RWED,GR:RWED,WO:RWED)
```

or:

```
MCR> UFD dev:[1,6]/PRO=[RWED:RWED:RWED:RWED]
```

When you want an error report, run the preanalyzer (PSE) and the analyzer (SYE) tasks. When PSE starts, it attempts to open the ERROR.TMP file that has the highest version number. If PSE cannot locate an ERROR.TMP file, it requests that ERRLOG close its current log file (ERR.TMP and renames that file to ERROR.TMP). When the next error occurs, ERRLOG creates a new ERR.TMP file and continues logging errors. The preanalyzer uses the information in ERROR.TMP to produce a formatted file. The default file name and UFD are ERROR.SYS and [1,6], respectively. However, any operator-specified name and UFD can be used. When the analyzer is run, it uses that formatted file to produce a list file as illustrated in Appendix C.

ERROR.SYS remains on disk until you delete it, because it contains information that can be reprocessed by the analyzer or processed by user-written tasks to provide a report with different content. Appendix C shows the layout of records in ERROR.SYS.

9.2 Functional Description of the T/MSCP Error Log Server

ER$LOG is the IAS MSCP error log processor. ER$LOG receives error information from the T/MSCP disk handlers (DU... and MU...) and the host initiated bad block replacement program (HI$BBR). The server formats the error information for processing by the T/MSCP error log reports generator (RPTGEN). ER$LOG places these records into the error file SY:[1,6]ERRDSA.TMP.

NOTE: The ER$LOG server only accepts information from tasks DU..., DC..., MU..., and HI$BBR. Therefore, you should not change the handler device pseudo names (for example, DU... becomes DX...). (Since double digit device unit numbers are now supported, this practice is no longer necessary.)

The ER$LOG server is an integral part of T/MSCP error reporting and should be installed whenever an IAS T/MSCP handler (DU... or MU...) is installed. The ER$LOG server is not compatible with T/MSCP handlers available before IAS Version 3.3.

To install the ER$LOG server, type the following command at the PDS> or MCR> prompt:

```
INS LB:[11,1]ERRDSA
```

To turn the ER$LOG server off, type the following command at either the PDS> or MCR> prompt:

```
RUN [11,1]ERROFF.
```

ERROFF reports when ER$LOG is stopped. Once ER$LOG is stopped, remove the server from the installed task list by typing the following command at either the PDS> or MCR> prompt:

```
REM ER$LOG
```

NOTE: ERROFF also stops the traditional error logging process ERRLOG.

Failure to remove the server from the installed task list results in the server restarting at the next error encountered by DU... or MU....

9.3 MSCP Report Generator—RPTGEN

Read this section to determine when it is necessary to call Digital field service.

The IAS RPTGEN produces reports based on the raw data gathered by ER$LOG in file SY:[1,6]ERRDSA.TMP. The RPTGEN reports provide Digital field service with information necessary to maintain the T/MSCP disk facilities supported by the IAS operating system.

The T/MSCP error log reports contain eight separate data formats. Each format includes information Digital field service personnel use to determine the state of the T/MSCP hardware. Each data format is listed under the "MSCP Packet Format = " as follows:

- For disk:
  - Disk Transfer Error
  - Bad Block Replacement
  - Standard Disk Interface Error
  - Small Disk Error
  - Controller Error
  - Host Memory Access Failure
  - Last Fail
  - Invalid Format Type
- For tape:
  - Tape Errors
  - STI (standard tape interface) failures

The following sections detail the data formats in the MSCP error log reports.

9.3.1 Data Packets

Disk Transfer Error

A Disk transfer error (DTE) packet indicates that an error occurred on a device during an I/O operation. These errors occur most frequently when one or more accessed blocks exceed the recovery threshold (ECC) limit for the device. Detection of a DTE usually results in the invocation of the BBR algorithm either by the device controller or by the host operating system.

Infrequently during the course of normal operation, random DTE errors might occur. However, during bad block (BAD utility) processing, the BAD utility makes several read and write passes over the entire disk looking for suspect (bad) blocks. As suspect blocks are accessed, DTE packets are generated. Therefore, many DTE packets can be generated during BAD utility processing of an MSCP device.

If disk transfer errors occur frequently during normal operations, a device might be malfunctioning. Notify Digital field service and remove the device from service until a determination is made as to the cause of the errors.

For more information on the BAD utility, refer to the *IAS Utilities Manual*.

Bad Block Replacement

A bad block replacement (BBR) packet is generated when a block is reported as suspect and is processed by the BBR algorithm. Expect to see a BBR packet for each disk transfer error (DTE) packet logged.

The BBR packet provides information about the results of BBR processing. This is the product of normal I/O operations and its presence should not be considered an indication of a hardware failure.

Standard Disk Interface

The presence of a standard disk interface (SDI) packet indicates errors detected in the communications path between a controller and a drive, or errors that occur within a drive (for example, the fault light comes on). These failures can be the result of something as simple as a loose SDI cable or as complex as failing electronics in the communications path.

When SDI errors occur frequently during normal operations, a device might be malfunctioning. Notify Digital field service and remove the device from service until a determination is made as to the cause of the errors.

Small Disk Error

This packet is essentially a combination of the SDI and DTE error log packets. It is used for *low end* disk controllers such as the RQDX3 and RD series disk drives.

Controller Error

This packet type contains information about failures within a controller. Continued logging of controller error packets is an indication of a hard disk failure and should be reported to Digital field service.

Host Memory Access Failure

This packet type contains information about the controller's failure to access host memory for data transfers.

Last Fail

Last fail packets can be logged either when a controller error occurs or a controller is initialized. In the case of a controller error, the last fail packet contains information about what last went wrong before a controller entered a *hung* state. A controller reinitialization can occur when a controller *times out* (hung after a controller error) resulting in the MSCP handler resynchronizing the hardware, or when the system is rebooted.

The last fail packet provides information about the reasons for controller reinitialization. This is often the product of normal I/O operations and its presence should not be considered an indication of hardware failure.

Invalid Format Type

This packet type indicates that the report generator cannot decipher a logged error log packet.

Tape Errors

This packet contains information about errors occurring due to media problems. When these errors ocur it is best to remove the suspect tape rom service.

NOTE: These errors can be produced by writing at 6250 BPI... to a tape that is not certified for 6250 bpi operations.

STI Errors

Tape equivilent to disk SDI errors.

9.3.2 Running T/MSCP RPTGEN

To run T/MSCP RPTGEN, enter the following PDS or MCR command:

```
RUN LB:[11:1]RPTGEN ESCAPE
```

Once RPTGEN is invoked, it prompts with the following instructions:

```
IAS MSCP Error Log Report Generator
```

Enter a command line in the following format:

RPTGEN

FORMAT

RPTGEN *[outdev][outufd]outfile name] = [indev][inufd][infile name][/switches]*

PARAMETERS

outdev

Any device name and unit number with ":". Default is SY:

For example:

```
DU0:, TI:
```

outufd

The UFD that comprises a group and owner numbers. Default is the current user UFD.

For example:

```
[100,10], [1,6].
```

outfile

Standard IAS filename in the form name.ext, where the name comprises up to nine alphanumeric characters and the extension comprises up to three alphanumeric characters. Default is ERRDSA.RPT.

indev

Any device name and unit number with ":". Default is LB:

For example:

```
DU0:, DL1:
```

inufd

The UFD that comprises group and owner numbers. Default is [1,6].

For example:

```
[100,10], [1,6].
```

infile

Standard IAS filename in the form name.ext, where the name comprises up to nine alphanumeric characters and the extension comprises up to three alphanumeric characters. Default is ERRDSA.DAT.

switches

The following list gives the valid switches.

Switch	Explanation
/EM	Include end message packets in report.
/BG	Exclude all data with a date earlier than the date in this switch.
/ED	Exclude all data with a date later than the date in this switch.
/TA	Include only T/MSCP (tape) data in report.
/DI	Include only MSCP (disk) data in report.

Format for the BG and ED switches is:

```
/xx:hh:mm:ss:dd:mm:yy
```

where:

xx = (BG) beginning date/time
xx = (EN) ending date/time
hh = hours (as a decimal number)
mm = minutes (as a decimal number)
ss = seconds (as a decimal number)
dd = day of month (as a decimal number)
mm = month (as a decimal number)
yy = year (as a decimal number)

All date/time entries require data in all fields, including the zeros if time is not to be used.

EXAMPLE(S)

```
RPT> =LB:[1,6]ERRDSA.DAT
```

This command generates a report on the default file (SY:ERRDSA.RPT) in the current UFD that contains all error log events except *end message* events.

```
RPT> SY:[1,1]ERRLOG.RPT=
```

This command generates a report to the file ERRLOG.RPT in the [1,1] UFD on the system disk using the default input data file (LB:[1,6]ERRDSA.DAT) that contains all error log events except end message events.

```
RPT> =
```

This command generates a report on the default file in the current UFD that uses the default input data file (LB:[1,6]ERRDSA.DAT). The report contains all error log events except end message events.

```
RPT> =/EM/BG:0:0:0:6:1:90/ED:23:59:59:14:2:90
```

This command generates a report on the default file in the current UFD that uses the default input data file (LB:[1,6]ERRDSA.DAT). The report contains all error log events, including end message events that occur between January 6, 1990 through February 14, 1990.

```
RPT> SY:[1,6]ERRLOG.RPT=LB:[1,6]RAWDAT.DAT/EM/BG:12:05:00:14:12:89
```

This command generates a report to the file ERRLOG.RPT in the [1,6] UFD on the system disk using the input data file (LB:[1,6]RAWDAT.DAT). The report contains all error log events including end message events that occur between 12:05 on December 14, 1989 to the last one in the data file.

If you do not choose the default input and output files, enter only the section to be changed. The following example shows how you could change the UIC of the output file to [10,11] and accept the rest of the default output file.

9.4 Executive Error Log Generation

This section provides operating procedures for the three error logging and analysis tasks (ERRLOG, PSE, and SYE) and for the task that terminates error logging (ERROFF).

9.4.1 Running ERRLOG

ERRLOG must be running for error statistics to be accumulated and for the raw data file to be passed to the preanalyzer (PSE), if an ERROR.TMP file is not ready for PSE to process. Normally, you install ERRLOG during system generation. To run ERRLOG, type the following command to SCI:

```
SCI> RUN/NOPROMPT ERRLOG
```

The MCR command is:

```
MCR>RUN ERRLOG ESC
```

The task responds with the following message:

```
NUMBER OF ERROR BUFFERS?
```

Type a value in the range 1 through 5 and press [Return]. The value indicates the number of error log buffers (of 80 words each) to be picked from the system node pool. These are used to hold the error log information until the ERRLOG task can run and write it to disk. A value of 2 is normally sufficient, unless very high error rates are being experienced.

After you have entered the number of buffers, type [Ctrl/C] to reactivate MCR or SCI.

Errors might occur more rapidly than they can be handled by ERRLOG. In this case, the summary report contains an entry under the heading NUMBER OF ERRORS MISSED. This entry indicates the number of errors that were not logged due to insufficient node space. Because the device-specific reports provide sequential numbers for errors, you can determine at which point errors occurred but were not logged.

If a large amount of node space is allocated, it may adversely affect the ability of other tasks to acquire enough dynamic memory to run.

9.4.1.1 Abnormal Termination of ERRLOG

The ERRLOG task terminates automatically in three cases:

1. When the desired number of nodes cannot be obtained.
2. If the error logging device used by ERRLOG becomes full.
3. If an error occurs when writing to the logging device.

For procedures to terminate error logging normally, see Section 9.4.5.

The ERRLOG task terminates at task startup when the task cannot obtain the number of nodes that you have specified. When this situation occurs, the following message displays on the console:

```
FAILED TO PICK ERROR LOG BUFFER
```

To attempt to run ERRLOG again, type the request to run ERRLOG as described above. When the request for the number of nodes is printed, respond with a smaller number to the question NUMBER OF ERROR BUFFERS?

The second case that causes ERRLOG to terminate is when the error logging device becomes full. The following message prints on the console:

```
ERROR LOGGING DEVICE ddn FULL
```

In this case, direct the error log files to an alternative device, as described above, and restart error logging. Alternatively, free some space on the error logging device (see the *IAS Performance and Tuning Guide*).

```
CREATE/DIRECTORY dev:[1,6]/PRO:(SY:RWED,OW:RWED,GR:RWED,WO:RWED)
```

The third case where ERRLOG is terminated is if an error occurs while you attempt to write to the logging device. The following information displays on the console:

```
ERROR - xx ON ERROR LOGGING DEVICE device name
"ERRLOG" TASK EXITING.
TASK "ERRLOG" TERMINATED
VIA 'EXIT' WITH PENDING I/O
```

Where:

- xx = Standard system code as defined in the *IAS Device Handlers Reference Manual*. If MO is loaded, additional error information is printed on the terminal. If you want to continue error logging, re-assign the logging device.

9.4.2 Running PSE

The preanalyzer (PSE) formats the raw data collected by ERRLOG into a file to be processed by SYE. To run PSE, you must be operating under a privileged UIC. In addition, if no ERROR.TMP file is available for PSE to process, the ERRLOG task must be running. ERRLOG can rename the ERR.TMP file and pass it to PSE.

Use the following command to run PSE:

```
RUN [11,1]PSE
```

The preanalyzer responds with the prompt PSE> and waits for you to type a command line. The format of the PSE command line follows:

```
outdev:[ufd]file.typ=indev:
```

The output file specification is a standard IAS specification.

The input file specification consists only of the input device specification. The file name is always ERROR.TMP and it is under UFD [1,6]. The name is assigned by ERRLOG.

The following defaults are used for omitted portions of the file specifications:

- outdev—defaults to SY:
- ufd—defaults to [1,6]
- file.typ—defaults to ERROR.SYS

- indev—defaults to SY:

If the default values are to be used, type [Return] in response to the PSE prompt. When PSE prompts again, type [Ctrl/Z] to return to SCI or MCR.

9.4.3 Running SYE

The analyzer produces an error report in the form of a printed listing or a listing file.

Before SYE can run, you must be operating at a privileged terminal. Type the following command to run SYE:

```
RUN [11,1]SYE
```

The analyzer responds with the prompt SYE> and waits for you to type a command line. The format of the SYE command line is:

```
outdev:[ufd]=indev:[ufd]file.typ/switch1.../switchn(WIDE)
```

The output file name corresponds to the values specified for xxx and yyy for the /BR: switch described below. The file type is LST. Do not include the output file name and type in the command line.

The input file specification is a standard IAS file specification. The input device, UFD, file name, and file type must be identical to the output file specification used when you run PSE. SYE analyzes the output of PSE.

Use the following switches as part of the input file specification:

- /BR:xxxyyy—The breakout switch that determines what information is to be included in the report. xxx can have one of the following values:
  - ALL indicates that error statistics for all disk, magnetic tape, and DECtape units are to be included.
  - DEV indicates that only device errors are to be itemized in the report; that is, no memory parity errors are to be included.
  - MEM indicates that only memory parity errors are to be included in the report.
  - SYS indicates that only start and stop entries are to be included in the report.
  - DSK indicates that error statistics for all disk units are to be included.
  - MAG indicates that error statistics for all tape devices, both magnetic tape and DECtape, are to be included.
  - ALL is the default value for xxx.
  - yyy can have one of the following values:

    ALL indicates that both the device-specific and the summary information is to be included in the report.

    SUM indicates that only the summary information is to be included.

    DSK indicates that only disk errors are to be included in the summary report.

    MAG indicates that only magnetic tape errors are to be included in the summary report.

SUM is the default for yyy.

- The output file name is formed from the xxxyyy portion of the /BR switch. The output file type is .LST.

- /ID:name—Indicates that the report of errors is to contain only those errors that occurred while a specified volume is mounted. The value name provides the volume identification.

  The volume can be mounted using the /OVR switch if the volume identification is not known. The volume information printed on the console gives the volume identification once the device is mounted.

- /DV:ddnn—Indicates that the report is to contain only errors that occur on a specified device type or on a specified unit. For example, if ddnn is specified as DK, error statistics for all RK03 or RK05 units are provided. If ddnn is specified as DK1, error statistics for RK03 or RK05 unit 1 are provided.

- /BG:time:date—Indicates that only those errors that occur after the specified time and date are to be included in the report. The format of the time and date specification follows:

  ```
  hh:mm:ss:dd:mm:yy
  ```

  All numbers are decimal and you must specify all six fields.

- /ED:time:date—Indicates that only those errors that occurred on or before the specified time and date are to be included in the report. Time and date have the same format as in the /BG switch.

- /DE—Indicates that a more detailed report than that produced using the defaults is desired. If you specify /DE, additional information is included if it is available as follows:

  a. Device error register bit description.
  b. Task and function that caused the entry to be logged.
  c. Statistical information about the device.
  d. Information relating to concurrent activity on the UNIBUS.

     The default is NODE or -DE.

- /-SU—Indicates that the summary report is not to be produced. The default is /SU.

The following are the default values for the SYE command string:

```
SY0:[user uic]ALLSUM.LST=SY0:[1,6]ERROR.SYS/BR:ALLSUM/NODE/SU
```

9.4.4 Terminating Error Logging

Use the following procedures to terminate the error logging task (ERRLOG) in an orderly fashion:

```
SCI> INSTALL [11,1]ERROFF
SCI> RUN/REALTIME ERROFF
```

or:

```
MCR> INS [11,1]ERROFF
MCR>RUN ERROFF
```

One of two messages displays on the terminal when ERROFF has terminated error logging. The following message is printed to indicate that ERROFF executed successfully:

```
"ERRLOG" TASK TERMINATED
```

The following message is printed to indicate that ERRLOG was not active and, therefore, was not terminated:

```
"ERRLOG" TASK NOT ACTIVE
```

9.4.5 Error Messages

Both the preanalyzer (PSE) and the analyzer (SYE) issue error messages to inform you of operational difficulties.

9.4.5.1 PSE Error Messages

After each error message, PSE prompts again for a new command line.

PSE, COMMAND STRING PARSE ERROR

Explanation: A syntax or semantic error was encountered while examining the input command string to PSE. PSE prompts again for a new command line.

User Action: Type a corrected version of the command line.

PSE, DELETE ERROR

Explanation: After the preanalyzer processed the input file ERROR.TMP, it was unable to delete it.

User Action: Attempt to delete the file using PDS or MCR PIP.

PSE, INPUT FILE ERROR

Explanation: An error was encountered while trying to open or obtain data from the input file ERROR.TMP. ERROR.TMP is closed, processing is terminated, and the input file is not deleted. unable to delete it.

User Action: PSE prompts again for a new command string. If this fails, delete the file.

PSE, NO ERROR FILES FROM SYSTEM

Explanation: The preanalyzer is unable to locate a file named ERROR.TMP. This message can be caused by one of the following situations:

1 No errors have occurred. Therefore ERRLOG has no data file to pass to PSE.

2 ERRLOG is not running and, therefore, cannot rename the ERR.TMP file to ERROR.TMP and pass it to PSE.

3 ERRLOG is not writing to the specified device. Check the LUN assignment of ERRLOG. The logging device is assigned to LUN 4.

User Action: If the cause of the message is that ERRLOG is not running, follow the procedures in Section 9.4.1 to run the task.

If the cause of the message is that the ERROR.TMP file resides on a device other than the one specified, use the device assigned to LUN 4 of ERRLOG as the indev for the PSE command line.

PSE, OUTPUT FILE ERROR

Explanation: An error was encountered while working with the PSE output file. Both the input and output files are closed. ERROR.TMP is not deleted.

User Action: Try to rerun PSE.

PSE, PREANALYZER OUTPUT DEVICE FULL

Explanation: The output device became full while PSE was writing data to the output file. Both the input and output files are closed. ERROR.TMP is not deleted.

User Action: Rerun PSE using a different output volume.

PSE, UNABLE TO CLOSE INPUT FILE

Explanation: PSE is unable to close the file ERROR.TMP. The file is not deleted. File processing is terminated.

User Action: Use PDS or MCR PIP to delete the file.

PSE, UNABLE TO CLOSE OUTPUT FILE

Explanation: PSE is unable to close the output file.

User Action: Rerun PSE or use a PDS or MCR PIP command to delete the file.

9.4.5.2 SYE Error Messages

SYE_COMMAND, STRING ERROR—PORTION OF THE STRING IN ERROR.

Explanation: The format convention within a particular portion of the command string is violated. No files remain open. SYE issues a prompt.

User Action: Correct the error and type the command.

SYE_COMMAND, STRING ERROR—ERROR NUMBER n

Explanation: The command string interpreter detects an error while attempting to get a command line. n is a CSI error code. See the *IAS I/O Operations Reference Manual* to determine the meaning of n. No files remain open. SYE issues a prompt.

User Action: Correct the error and type the command.

SYE_COMMAND, STRING SYNTAX ERROR

Explanation: The proper format was not used in the command string. No files are open. SYE issues a prompt.

User Action: Type the corrected command.

SYE_DEVICE, ERROR INPUT FILE—FATAL ERROR n

Explanation: SYE was attempting to obtain further information from the input file but could not get the next record. n is an FCS error code. See the *IAS I/O Operations Reference Manual* to determine the meaning of n.

User Action: Both the input and output files are closed. SYE issues a prompt for the next command.

SYE_DEVICE, ERROR OUTPUT FILE—FATAL ERROR n

Explanation: SYE was unable to write information in the output file. n is an FCS error code. See the *IAS I/O Operations Reference Manual* to determine the meaning of n.

User Action: Both the input and output files are closed. SYE issues a prompt for the next command.

SYE, ILLEGAL BREAKOUT SWITCH—/BR:xxxyyy

Explanation: SYE issues this message when the operator attempts to request a breakout of the input file that is not legal. No files remain open. SYE issues a prompt for another command.

User Action: Retype the command with a correct use of /BR:.

SYE_OPEN, FAILURE ON INPUT FILE—FATAL ERROR n

Explanation: SYE was unable to open the input file. No files remain open. n is an FCS error code. See the *IAS I/O Operations Reference Manual* to determine the meaning of n.

SYE_OPEN, FAILURE ON OUTPUT FILE—FATAL ERROR n

Explanation: SYE was unable to open the output file. No files remain open. n is an FCS error code. See the *IAS I/O Operations Reference Manual* to determine the meaning of n.

SYE_n, PAGES filename

Explanation: This message does not indicate an error; rather, it indicates the number of pages (n) in the finished report. The value filename indicates the file name for future reference.

SYE_SUMMARY, TABLE OVERFLOW REPORT CONTINUES WITHOUT SUMMARIES

Explanation: Summary table overflow occurred because more devices received errors than the analyzer is built to handle.

User Action: Use the breakout switches to reduce the number of devices included in the summary.

10 Memory Parity Support

Under IAS, the handling of processor faults is divided into three categories:

1 Main memory parity errors.

2 Cache memory parity errors (where cache is used).

3 Additional PDP-11/70 hardware errors.

For a description of cache memory, refer to the approriate PDP-11 Processor Handbook.

All memory parity errors are handled by one of the following Executive modules selected during system generation.

MP45	for PDP-11/34 (without cache), PDP-11/35, PDP-11/40, PDP-11/45, PDP-11/50, PDP-11/55.
MP60	for PDP-11/34 (with cache), PDP-11/44, PDP-11/60.
MP70	for PDP-11/70.
MPNCNE	for systems without parity memory.

Use the correct module. Otherwise system operation will be unpredictable if a parity error occurs.

It is possible to select MPNONE on a system which does have parity memory. This will save about 100 words of memory. If a memory parity error does occur, the system will crash.

10.1 Main Memory Parity Support

Main memory parity errors are an indication of a memory malfunction. When a parity error occurs, it causes a trap. A check is made to determine whether the error was fatal, (irrecoverable) or non-fatal, and then appropriate action is taken. On PDP-11 computers other than the PDP-11/70, all main memory parity errors are fatal and are treated as main memory errors.

The PDP-11/70 hardware recognizes two types of main memory errors: address and data. IAS handles both types of errors identically, although errors are logged as being one of the two types. See Chapter 9 for information on error logging.

When a fatal main memory parity error occurs, the task executing where the error occurred is terminated unless that task has a hardware priority greater than zero in which case it will crash. The aborted task remains in memory, which prevents further access to or execution of the task. Locking the task in memory also makes the entire area of memory allocated to the task inaccessible to the system or other tasks. When a task is terminated in this way, an error message is printed on the terminal that initiated the task.

If any of the following conditions are encountered, a system crash occurs and system operation halts:

1 The hardware priority of the task executing where the error occurred is greater than zero.

2 No task is currently executing.

3 The error occurred in Kernel mode.

The only difference between parity handling on the PDP-11/70 and other PDP-11s, is that the 11/70 indicates whether the error was fatal. Fatal errors are handled as described above. Non-fatal errors are transparent to the task executing where the error occurred, but are logged by the system as described in Chapter 9.

10.2 Cache Parity Support

Because cache contains a copy of information also maintained in main memory, fatal parity errors are never returned for cache. Rather, the system performs the following sequence:

1. Determines the cache group in which the error occurred.
2. Determines whether the group has exceeded the number of errors allowed in a one-minute period.
3. Turns the group off if the limit has been exceeded.
4. Logs the error. This process is transparent to the task using the cache group in which the error occurs.

   The default number of errors allowed in a one-minute period is 50 (decimal).

The PDP-11/70 hardware recognizes two types of cache errors: address and data. Both types of errors are handled identically, although errors are logged as being of the two types; refer to Chapter 10 for information on error logging. See Chapter 12 for details on the SET MEMORY CACHE or SET MEMORY PARITY commands and the *IAS MCR User's Guide* for details on the MCR SET /CAC command.

10.3 Additional PDP-11/70 Hardware Error Support

IAS handles three other types of hardware errors that the PDP-11/70 reports:

1. Nonexistent memory
2. UNIBUS error
3. CPU/UNIBUS abort

10.3.1 Nonexistent Memory

A nonexistent memory error implies that faulty mapping or a bad address has caused the central processing unit to refer to a nonexistent memory location. This type of error is treated in the same manner as a main memory error.

10.3.2 UNIBUS Error

A UNIBUS error occurs when bad memory is referred to during a Direct Memory Access (DMA) transfer. Since the task currently executing is probably not the task for which the I/O was being done, no attempt is made to determine which task issued the I/O request, or to locate the bad memory. The error is logged. Some UNIBUS device controllers detect the memory error and report it at the end of the transfer.

10.3.3 CPU/UNIBUS Abort

The CPU/UNIBUS abort occurs when bad data is transferred during a non DMA I/O transfer or a read of UNIBUS External Page registers. The task currently executing is terminated because it is the task that issued the I/O request. The error is logged.

11 Shadow Recording

Shadow recording is a technique used to increase the reliability of data retrieval by recording data on two disks instead of one. The effect of shadow recording is transparent to user-level tasks.

Shadow recording uses a primary disk and a secondary disk, referred to together as the shadow set. The secondary disk duplicates, or *backs up* the primary disk.

When shadow recording is in effect, the IAS Executive duplicates each write I/O packet written to the primary disk and directs the copy to the secondary disk.

During a read operation from a shadow set, the IAS Executive reads the primary disk first. If a read error occurs on the primary disk, the Executive reads the secondary disk.

Shadow recording has the following advantages:

- Near simultaneous data recording and backup.
- Quicker restoration of data.
- Decreased liklehood of a hard disk read error.

Shadow recording has the following disadvantages:

- Doubles the number of I/O write packets.
- Decreases throughput.
- Uses an entire disk drive.

Shadow recording must be added to the system during system generation. Refer to the *IAS Installation and System Generation Guide* for information about selecting the shadow recording feature.

NOTE: Two restrictions apply to the use of shadow recording:

- **Both disks must be of the same device type**
- **A disk cannot be shadow recorded repeatedly. For example, if you make DU0: and DU1: a shadowed pair, you can make DU2: and DU3: a shadowed pair but not DU1: and DU2:.**

11.1 Shadow Recording Control

Shadow recording is controlled by five commands, which are listed and described in the following section. These commands can be used in two formats.

1 You can invoke shadow recording and specify a shadow recording command on the same line, as follows:

```
>SHAdow command
```

2 You can first invoke shadow recording, then enter a shadow recording command to the SHA> prompt:

```
>SHA
SHA> command
```

11.2 Shadow Recording Commands

Five commands control shadow recording. You can enter these commands only from a privileged terminal. The commands enable you to start, stop, abort, or continue shadow recording, and to display all device pairs that are being shadowed.

ABORT Command

```
SHA> ABORT ddnn:
```

Where:

ddnn: = The device-unit of the primary Files-11 device.

The ABORT command stops shadow recording even if catch-up is in progress.

The ABORT command performs the following functions:

- Verifies that the primary disk is part of a shadowed pair.
- Aborts catch-up if it is in progress and then executes a STOP command.

CONTINUE Command

```
SHA>CONTINUE ddnn: TO ddxx:
```

Where:

- ddnn = The primary Files-11 device to be duplicated on the secondary Files-11 device.
- ddxx = The secondary device to be the shadowed copy of the primary device.

The CONTINUE command enables you to restart shadow recording on the same two disks that you were using when the STOP command was issued. The CONTINUE command assumes that the two disks are physically alike and does not check the disks for equality. For this reason, you should take care that nothing occurs to make the two disk packs different (for example, writing to the primary disk with shadow recording turned off) before the CONTINUE command is issued.

The CONTINUE command performs the following functions:

- Assumes that the contents of the primary and secondary disks are identical.
- Verifies that the primary disk is a Files-11 pack and is mounted properly.
- Verifies that the secondary disk is mounted with the /FOREIGN qualifier.
- Verifies that both disk drives are of the same type.
- Sets up the shadow recording data structure, which starts the Shadow Recording operation within the Executive.

DISPLAY Command

```
SHA>DISPLAY
```

The DISPLAY command shows all shadowed pairs. Entering the DISPLAY command causes a display of the shadowed device pairs in the following format:

```
SCB     PRIMARY     SECONDARY
xxxxxx   ddnn:       ddxx:
```

Where:

- xxxxxx = Address of the SCB control block for shadow recording. This control block is in the system Executive pool space.
- ddnn = Primary Files-11 device to be duplicated on the secondary Files-11 device.
- ddxx = Secondary device to be the shadowed copy of the primary device.

START Command

```
SHA>START ddnn: TO ddxx:
```

Where:

- ddnn = Primary Files-11 device to be duplicated on the secondary Files-11 device.
- ddxx = Secondary device to be the shadowed copy of the primary device.

The START command performs the following functions:

- Verifies that the primary disk is a Files-11 device and is mounted.
- Verifies that the secondary disk is mounted using the /FOREIGN qualifier.
- Verifies that the primary and secondary disks are identical disk types.
- Sets up the shadow recording data structure (SCB), which starts the shadow recording operations within the Executive.
- Starts copying the primary disk to the secondary disk (catch-up).

After the START command initiates shadow recording, all data written to the primary disk is also written to the secondary disk. Since shadow recording is implemented by duplicating I/O write packets, the secondary disk might not logically duplicate the primary disk at any given time.

The information about the last block copied is in the SCB control block.

A fully redundant secondary disk does not exist until the catch-up operation has completed.

NOTE: START must be executed before timesharing is active.

STOP Command

```
SHA>STOP ddnn:
```

Where:

- ddxx: = Device-unit of the primary device.

The STOP command prevents shadow recording from continuing, unless catch-up is in progress. If you issue the STOP command during catch-up, you will receive an error message.

To stop shadow recording during catch-up, use the ABORT command.

The absence of outstanding I/O does not necessarily mean that all file activity has stopped. You must make certain that all tasks that write onto the shadowed pair have stopped before you issue STOP or ABORT to ensure the integrity of the data on the secondary disk.

The STOP command performs the following functions:

- Verifies that the primary device is part of a shadowed pair.
- Unlinks the SCB data structure, if no outstanding I/O exists, thereby stopping the Executive from shadow recording.
- Marks the data structure for deletion if outstanding I/O does exist.

NOTE: You cannot start shadow recording if timesharing is active.

11.3 Preparing to Use Shadow Recording

Before you use shadow recording, you must make the two disk packs to be used for shadow recording logically equivalent. This involves identifying and recording the sections of both disk packs that cannot be written to (called bad blocks).

NOTE: The term *shadow set* is used in the following explanation. At a minimum, the shadow set contains two disk packs. However, it is a worthwile precaution to prepare three disk packs, in case there is trouble with a disk pack. In this case, the shadow set contains three disk packs.

To make the disk packs of the shadow set logically equivalent, the bad blocks on each disk in the shadow set must be located and recorded on each disk in the shadow set. Thus, the bad block information on each disk pack is the logical OR of its own bad blocks and the bad blocks of all the disk packs in the shadow set. Because the same bad block information is recorded on each disk pack, a write operation to the secondary disk pack occurs at the same physical location as on the primary disk pack.

To make the disk packs logically equivalent, use the following two utilities:

- Bad Block utility (BAD)—Identifies and locates bad blocks on all disks. For a complete description of the BAD utility, see the *RSX-11M/M-PLUS Utilities Manual.*
- INIVOLUME utility—Uses information provided by the BAD utility to initialize each disk pack and record the locations where bad blocks occur on other disk packs in the shadow set. For a complete description of the INITVOLUME utility, see the *IAS MCR User's Guide.*

Follow these steps to prepare the shadow set disk packs.

1 Determine which two disk drives are to be the shadow pair. (One of them is the primary disk and the other is be the secondary disk.)

2 Obtain the disk packs to make up the shadow set. These disk packs are initialized before they are written to, so do not use disk packs that contain important information.

3 Use the BAD utility to locate bad blocks on each disk pack. Invoke the BAD Utility as follows:

```
BAD/LI
```

The /LI qualifier causes the bad block locations found on the disk pack to be displayed on the terminal. Write down the locations of the bad blocks. Repeat this operation for each disk pack in the shadow set.

NOTE: If you are using DU disk devices and HIBBR, you can skip Step 4 as long as the BAD utility reported zero (0) bad blocks. HIBBR ensures that the disk has zero bad blocks.

If BAD reports any bad blocks on a DU device, follow these steps:

❶ Verify that HI$BBR and ER$LOG are installed on your system. If not, install LB:[11,1]HIBBR and LB:[11,1]ERRDSA and repeat the bad block scan. If BAD now reports zero (0) bad blocks, proceed to Step 5. Otherwise, continue.

❷ Run RPTGEN to determine the nature of the errors. If they appear to be minor media errors, change disks and repeat the bad block scan.

❸ Call your Digital field service representative and show him or her the RPTGEN listing.

4 To make the disk packs of the shadow set logically equivalent, you must now initialize each disk pack in the shadow set with the bad blocks on the other disk packs. (Bad block information is recorded on the bad block descriptor block of a disk pack.)

   To initialize a disk pack and record bad block information, run the INITVOLUME utility with the /BAD qualifier. Invoke INITVOLUME as follows:

```
INIT/BAD=MAN
```

   The utility displays the following prompt:

```
INI>BAD=
```

   At the prompt, enter the locations of all bad blocks found on each of the disk packs. To exit, press Ctrl/Z.

   You can also use BRU to initialize a disk. BRU is useful if you are planning to populate your primary disk with data from another pack. You must use the /BAD:MANUAL qualifier for BRU to enter the bad block information you obtained earlier. Refer to the *RSX-11M/M-PLUS Utilities Manual* for more information about BRU.

5 Once each pack has been properly initialized for shadow recording, you can mount the disks. The primary disk is mounted as a Files-11 device and the secondary pack is mounted as a foreign device.

You are now ready to use shadow recording. Use the START command to begin shadow recording. Once shadow recording is started, all write operations to the primary disk are duplicated to the secondary disk.

Any errors that occur during shadow recording are reported to the console device.

11.4 Bad Block Handling

During a write operation, the Executive writes the data from a task buffer to the primary disk, then writes it to the secondary disk. If write errors occur on one of the disks, you might not know it until the Executive attempts to read back the block in error.

To be notified of write errors as they occur, you can run shadow recording with write check enabled for both disks of the shadowed pair. To enable write checking on the device you specify in ddnn:, use the following commands:

For MCR:

```
>SET /WCHK=ddnn:
```

For DCL:

```
SET DEVICE:DDn: WRITECHECK
```

Be aware that a write check of every write operation takes more time to execute than a write only.

If errors occur, the shadow recording program prints the error information on the console terminal. This information consists of the device and logical block numbers when available. In the case of a read, the Executive reads from the primary disk first.

If it encounters an error on the primary disk, it reads the same record from the secondary disk. It is extremely unlikely that both disks would develop an error in exactly the same place. However, if they do, an error message display about read errors on the secondary disk means that the primary disk is also bad in the same location.

If errors occur, follow the procedures detailed in Section 11.4.1.

11.4.1 Errors On The Primary Disk

If errors occur when the Executive tries to read from the primary disk, the Executive then tries to read the same record from the secondary disk. You are notified of the errors by a message on the console and, at this point, can let your task continue. If you do decide to let your task continue, be aware that the shadowed pair is no longer alike.

If you decide not to let your task continue, stop the processing task (or tasks) and stop shadow recording. If catch-up has finished, the secondary disk contains all the data that was on the primary disk, including the bad block record. Use the secondary disk as the new primary disk. Mount one of the spare prepared disk packs on the secondary disk using the /FOREIGN qualifier.

After you mount the old secondary disk pack as the new primary disk, issue the START command. The START command copies the entire primary disk to the secondary disk. The START command reinitializes the secondary disk.

11.4.2 Errors On The Secondary Disk

If errors occur on the secondary disk, the shadow of the primary data is no longer valid. Shadow recording should be stopped, a new secondary disk should be mounted and shadow recording restarted. The catch-up task then copies the primary disk to the new secondary disk.

11.4.3 Errors On The Primary And Secondary Disks

The occurrence of bad data in the same blocks of two different disks is very unlikely; this kind of error is not recoverable. However, whether or not an error occurs at the same or different places on both disks the only recourse is to stop shadow recording and your tasks. Then, select a new primary disk and new secondary disks, obtain and record the bad block information on the new primary disk as before, and copy the old primary disk to the new primary disk.

If the error occurred on different blocks on both disks, the previously processed information can be salvaged by using the shadowed pair as input to the Backup and Restore Utility (BRU), to produce a new disk. The output disk from BRU must be part of a shadow set, that is, must have the logical OR of bad block information of the replacement secondary disk. The output disk from BRU can be used as the new primary disk. Refer to Section 11.3 for information on creating a shadowed pair.

Information about how to use BRU is in the *RSX-11M/M-PLUS Utilities Manual*.

11.5 Shadow Recording Messages

HH:MM:SS,

HANDLER ERROR CODE-nn. on DU2:
SHADOW PAIR: DU0: DU2:
TASK: EDIT23
LOGICAL BLOCK: 10025
BYTES IN XFER: 00512
FILE ID: 35311,2
FILE NAME: AA.MAC;7

Explanation: An error occured on DU2:. If an error occurs at the same location on both disks, the first line of the message is repeated for each disk.

- HH:MM:SS = Time of the error in hours, minutes, and seconds.
- nn. = I/O error code of the error in decimal. I/O error codes are listed in the *IAS I/O Operations Manual.*
- FILE ID and FILE NAME are printed only when they can be determined at the time of the error.

User Action: See Section 11.4.

SHA, ABORT ERROR

Explanation: This message occurs if the shadow recording ABORT command cannot abort shadow recording during catch-up.

User Action: Wait until catch-up terminates, then issue the STOP command.

SHA, BAD INDIRECT FILE SPECIFICATION

Explanation: The indirect command file name is incorrect.

User Action: Reissue the indirect command file specification.

SHA, BAD VOLUME TYPE

Explanation: The volume is not a Files-11 device or the primary and secondary devices are different device types.

User Action: Check the command line of devices that you are trying to use.

SHA, CATCHUP COMPLETED

Explanation: This is an informational message. The primary disk has been completely copied to the secondary disk.

User Action: None.

SHA, CATCHUP IN PROGRESS

Explanation: You have tried to issue a STOP command during the catch-up process.

User Action: Issue an ABORT command if catch-up is still in progress.

SHA, I/O ERROR

Explanation: An I/O error occurred during catch-up.

User Action: None. If the I/O errors are frequent, enter the ABORT command, change disks, and restart shadow recording.

SHA, ILLEGAL COMMAND

Explanation: The command you entered is not a legal command.

User Action: Reenter the command.

SHA, INPUT ERROR IN INPUT COMMAND FILE
FILE=ddnn:filename.filetype, HANDLER ERROR CODE –26.

Explanation: An error occurred while the indirect command file was being read.

User Action: Try using a different indirect command file or a copy of that command file.

SHA, INTERNAL ERROR. ILLEGAL PARSE DATA

Explanation: The command was parsed, but the parsing caused an error.

User Action: This is an internal system error. The command was correctly entered. Retry the command.

SHA, INVALID DEVICE

Explanation: The device is not a disk device.

User Action: Check the device specification and the device.

SHA, MAXIMUM INDIRECT COMMAND FILE DEPTH EXCEEDED

Explanation: The indirect command file was nested to too many levels. Only 3 levels are permitted.

User Action: Alter command file nesting.

SHA, NO ROOM AVAILABLE FOR SCB

Explanation: Not enough pool space exists to allocate space for the SCB control block (for shadow recording).

User Action: Check your system. The system could be on the verge of running out of pool space.

SHA, NOT A SHADOWED PAIR

Explanation: When STOP or ABORT was issued, the wrong device was specified.

User Action: Reissue the command with the correct device-unit for the primary device.

SHA, OPEN ERROR ON INPUT COMMAND FILE
FILE=ddnn:filename.filetype, HANDLER ERROR CODE –26

Explanation: The input command file could not be found.

User Action: User Action: Check the command file and its location.

SHA, PRIVILEGED COMMAND

Explanation: You are not using a privileged account or terminal.

User Action: Log in on a privileged account.

SHA, SEND/RECEIVE ERROR

Explanation: A RECEIVE Executive directive error occurred in a shadow recording task during the catch-up process.

User Action: None. This is a system error.

SHA, SHADOW RECORDING NOT IN SYSTEM

Explanation: Shadow recording is not part of your system.

User Action: Shadow recording must be included in the system at system generation.

SHA, STARTUP ERROR

Explanation: Shadow recording cannot spawn the catch-up task. Shadow recording exits.

User Action: Restart shadow recording.

SHA, SYNTAX ERROR

Explanation: The entered command has incorrect syntax.

User Action: Reenter the command.

SHA, SYSTEM ERROR

Explanation: A system error occurred.

User Action: Determine the effects of the error if possible.

SHA, UNKNOWN COMMAND ERROR

Explanation: Shadow recording received a command line that it could not recognize.

User Action: Check the command that you entered or the indirect command file.

SHA, VOLUME CURRENTLY PART OF SHADOWED PAIR

Explanation: The specified device is already being shadow recorded.

User Action: Check the devices being shadowed with the DISPLAY command.

SHA, VOLUME NOT MOUNTED PROPERLY

Explanation: The primary device is not mounted with the Files-11 qualifier, or the secondary device is not mounted with the /FOREIGN qualifier.

User Action: Check the devices.

Part III Command Specifications

12 SCI Commands

12.1 Introduction

This chapter describes all the SCI commands, in alphabetical order. For detailed descriptions of all PDS and MCR commands see the IAS PDS User's Guide and the IAS MCR User's Guide. SCI commands can be issued only from the operator's console on a timesharing system, or by users with PR.SCI privilege on multiuser systems. All PDS commands are also available to the SCI user on the console terminal.

Table 12–1 lists all the PDS and SCI commands. Table 12–1 also lists the MCR commands that correspond to the system manager SCI commands, (those needing PR.SCI privilege). If your user/system interface is MCR, look up the required command in Table 12–1, then see the IAS MCR User's Guide for a detailed description.

PR.SCI privilege is used on the different types of IAS systems as follows:

1. On a timesharing system (that is, a system with the Timesharing Control Primitives (TCP)), PR.SCI is used only to check that a user can log in at the console terminal. After this, any SCI command described in this chapter is accepted on the console whatever privilege the user is assigned in the user's command privilege mask. SCI commands are not accepted on PDS (non-console) terminals, even if the user has PR.SCI privilege.

2. On a multiuser or real-time system (that is, a system without the Timesharing Control Primitives (TCP)), any user with PR.SCI privilege can issue SCI commands from any terminal.

Table 12–1 Command Summaries

PDS Command	Description	Required Privilege	MCR Command
ABORT/REALTIME	Abort a real-time task.	PR.RTC	
ABORT/TIMESHARING	Abort a timesharing task.	ANY	
ABORT/JOB[1]	Abort a timesharing job by job I.D.	PRI.SCI	N/A
ABORT/CLI[1]	Abort a CLI	PRI.SCI	ABO
ABORT/REGISTER_DUMP[1]	Display the contents of the registers when a task is aborted.	ANY	
ALLOCATE/DEVICE[1]	Allocate a device.	PR.DEV	
ALLOCATE/TERMINAL	Allocate terminals to a CLI.	PR.SCI	SET /CLI
APPEND	Add records from one or more files to the end of an existing file.	PR.FIL	
ASSIGN[1]	Assign a LUN to a device.	PR.RUN	
ASSIGN/TASK	Change the LUN assignment for an installed task.	PR.RTC	

[1]This command is applicable only to timesharing systems.

Table 12-1 (Cont.) Command Summaries

PDS Command	Description	Required Privilege	MCR Command
ASSIGN/REDIRECT	Redirect I/O requests from one device to another.	PR.SCI	RED
BASIC	Invoke the BASIC language translator.	PR.BAS	
BOOTSTRAP	Stop the operating system and bootstrap that system or another system from any device.	PR.SCI	BOO
CANCEL	Cancel the periodic scheduling of requests for a real-time task.	PR.RTC	
COBOL	Compile a COBOL source program.	PR.COB	
COMPARE	Compare two files with one another.	PR.FIL	
CONTINUE[/TIMESHARING]	Continue the execution of a previously suspended timesharing task.	ANY	
CONTINUE/MESSAGE	Continue a task suspended by the "suspend" form of message output (that is, the MO message handler).	PR.RTC	
CONTINUE/REALTIME	Continue the execution of a previously suspended real-time task.	PR.RTC	
COPY	Copy the contents of file(s) to another file.	PR.FIL	
CORAL	Compile a CORAL 66 source program.	PR.COR	
CREATE	Create a file.	PR.FIL	
CREATE/DIRECTORY	Create a directory.	PR.FIL	
CREATE/NAME	Create a synonym for a file.	PR.FIL	
CREATE/SWAP	Create a swap file.	PR.SCI	SWA
DCL	Enable the user to return from MCR mode to DCL mode.	ANY	
DEALLOCATE/DEVICE[1]	Deallocate a specified device.	PR.DEV	
DEALLOCATE/TERMINAL	Deallocate terminals from a CLI.	PR.SCI	SET /-CLI
DEASSIGN[1]	Dissociate a device from a logical unit.	PR.RUN	
DEASSIGN/TASK	Dissociate a logical unit for an installed task.	PR.RTC	
DELETE	Delete one or more specified files.	PR.FIL	
DELETE/SWAP	Delete a swap file.	PR.SCI	SWA

[1]This command is applicable only to timesharing systems.

Table 12–1 (Cont.) Command Summaries

PDS Command	Description	Required Privilege	MCR Command
DIRECTORY	List information regarding a file or group of files held within a specified UFD or disk.	PR.FIL	
DISABLE	Inhibit task execution of an installed task.	PR.RTC	
DISMOUNT	Dismount a volume from a specified device.	PR.DEV	
DISMOUNT/KEEP[1]	Dismount a volume but do not deallocate the device.	PR.DEV	
DISMOUNT/GLOBAL[1]	Dismount a globally mounted device.	PR.DEV	
DISMOUNT/REALTIME[1]	Dismount a device that was mounted for exclusive use by real-time tasks.	PR.DEV	
DISMOUNT/LOCK	Lock access on a volume and dismount the volume when the current file access is complete.	PR.DEV (multiuser systems only)	
DUMP	Produce a listing of the contents of a file.	PR.FIL	
EDIT	Invoke either the interactive editor (EDI), the batch editor (SLIPER), the DEC standard editor (EDT), or the keypad editor (KED or K52).	PR.FIL	
ENABLE	Reverse the effect of the DISABLE command.	PR.RTC	
$EOJ	Terminate a batch job.	NOPE	
EXAMINE	Open a memory address for examination and optional modification.	PR.SCI	OPE
FIX	Fix a task in its installed partition.	PR.RTC	
FORTRAN	Invoke the FORTRAN compiler to compile a FORTRAN IV, FORTRAN IV PLUS, or FORTRAN-77 source file.	PR.FOR	
GOTO	Transfer control to the next occurrence of a command line prefixed by a specified label.	ANY	
HELP	Display information.	ANY	
IDENTIFY	Display the version of the utility in use.	ANY	
INITIALIZE	Initialize a DOS, RT-11, or Files-11 volume.	PR.DEV	

[1]This command is applicable only to timesharing systems.

Table 12–1 (Cont.) Command Summaries

PDS Command	Description	Required Privilege	MCR Command
INSTALL	Install a task, library, common area or dynamic region.	PR.RTC	
INSTALL/CLI[1]	Install a CLI in the system.	PR.SCI	INS
$JOB	Initiate a batch job.	N/A	
LIBRARIAN	Create, delete and maintain object module libraries, MACRO-11 macro libraries, and Universal libraries.	PR.LIB	
LINK	Link object modules to form an executable task.	PR.LIN	
LOAD[1]	Respond to a load request when the requested volume has been loaded on the device.	PR.SCI	N/A
LOGOUT	Terminate an interactive session. Also release any allocated devices and mounted volumes.[2]	N/A	
MACRO	Assemble one or more source files containing MACRO-11 statements into a single relocatable binary object file.	PR.MAC	
MCR	Enable the user to enter MCR mode, or issue a single MCR command.	PR.MCR	
MERGE	Record from a sequential, indexed, or relative (transaction) file with an indexed relative (target) file.	PR.FIL	
MESSAGE	Send a message to another terminal(s). Certain qualifiers are not available on multiuser systems.	ANY	
MESSAGE/FORCE	Force a message to a terminal even if the terminal has requested not to receive messages.	PR.SCI	N/A
MOUNT	Request a volume to be mounted on the specified device and be made available to the user.	PR.DEV	
NOLOAD[1]	Respond to a load request when the requested volume cannot be loaded on the device.	PR.SCI	N/A
ON	Specify the action to be taken if the completion of a command returns an error.	ANY	
PRINT	Cause one or more specified files to be queued for output on the line printer.	PR.FIL	

[1]This command is applicable only to timesharing systems.

[2]This description is applicable only to timesharing systems.

Table 12–1 (Cont.) Command Summaries

PDS Command	Description	Required Privilege	MCR Command
QUEUE	Enables access to the queue to interrogate, insert, remove, or modify entries.	PR.FIL	
QUEUE/TEST	Test a line printer forms alignment.	PR.SCI	QUE /TE
REMOVE	Remove an installed task, common area, library, or region from the system.	PR.RTC	
REMOVE/CLI[1]	Remove an installed CLI from the system.	PR.SCI	REM
RENAME	Rename an existing file.	PR.FIL	
RUN[/TIMESHARING]	Run an executable timesharing task.	PR.RUN	
RUN/REALTIME	Run an executable real-time task either immediately or, optionally, at a specified future time.	PR.RTC	
RUN/CLI[1]	Run a CLI at a specified terminal.	PR.SCI	RUN
RUN/HANDLER	Load a device handler task for a specified device.	PR.SCI	LOA
SAVE	Save a generated system.	PR.SCI	SAV
SET ALLOCATION	Set the allocation factor scheduling parameters.	PR.SCI	UTL
SET BOOTSTRAP	Write the bootstrap block on the device.	PR.DEV	BOO /WB
SET CONTROLC[1]	Set the action of Ctrl/C on specified terminals.	PR.SCI	N/A
SET DAYTIME	Set the time and date to be used by the system.	PR.SCI	TIM
SET DEFAULT[1]	Establish a new default device and/or UFD.	ANY	
SET DEVICE	Set the buffer size and maximum record length for specified device.	PR.SCI	SET /WIDTH
SET ECHO	Enable or disable command echoing from an indirect command file.	NONE	
SET END_OF_FILE	Reset end-of-file pointers for a file.	PR.FIL	
SET EXTENDED_TASK_SIZE	Set the maximum extension size of a task.	PR.SCI	SET /MAXEXT
SET FORMS	Set the forms type on a specified device.	PR.SCI	OPR /FO
SET MEMORY	Set memory requirements for parity errors and setting memory cache groups on and off.	PR.SCI	SET /CAC

[1]This command is applicable only to timesharing systems.

Table 12–1 (Cont.) Command Summaries

PDS Command	Description	Required Privilege	MCR Command
SET PASSWORD[/BATCH]	Change the user's interactive or batch password.	ANY	
SET PRINTING DEFERRED	Defer the printing of spooled files generated by timesharing tasks run from the user's terminal.	ANY	
SET PRIORITY	Alter the priority of an active real-time task.	PR.RTC	
SET PRIVILEGED	Set specified terminals as privileged.	PR.SCI (available only on multiuser systems)	SET /PRV
SET PROTECTION	Change a file's protection code.	PR.FIL	
SET QUANTUM	Set the quantum scheduling parameters.	PR.SCI	UTL
SET QUIET	Suppress or allow the output information messages.	ANY	
SET REAL_TIME_CONTROL[1]	Enable CLIs and real-time tasks to run at a high priority.	PR.RTC	
SET SCI	Enables execution of SCI commands from a non-console terminal.	PR.SCI	
SET SERVICE[1]	Set the time between batch schedules or scheduler promotions parameters.	PR.SCI	UTL
SET SPOOL	Set or inhibit a device as an output spooled device.	PR.SCI	SET /SP
SET SWITCH_REGISTERS	Set software switch register.	PR.SCI	SET /SWR
SET TERMINAL	Change the characteristics of the user's terminal.	ANY	
	Alternatively, users logged in under a user name whose UIC is [1,1] or a system operator can change the characteristics of any terminal.	PR.SCI or [1,1]	TER
SET UIC	Set the terminal UIC.	PR.RTC or group code of less than 10 octal (available only on multiuser systems)	SET /UIC
SET VERIFY	Identical to the SET ECHO command.	NONE	
SHOW CLI	Display information on CLIs currently running in the system.	ANY	

[1]This command is applicable only to timesharing systems.

Table 12–1 (Cont.) Command Summaries

PDS Command	Description	Required Privilege	MCR Command
SHOW CLOCK_QUEUE	Display system clock queue.	ANY	
SHOW DAYTIME	Display the current time and date.	ANY	
SHOW DEFAULT[1]	Display the user's current default device and UFD.	ANY	
SHOW DEVICES	Display information about all or selected devices known to the system.	ANY	
SHOW EXTENDED_TASK_SIZE	Show the current task extension size.	ANY	
SHOW GLOBAL_AREAS	Display information about resident global areas.	ANY	
SHOW IO_QUEUES	Display system I/O queues.	ANY	
SHOW LUNS	Display current assignment of LUNs for an installed task.	PR.RTC	
SHOW MEMORY	Display the use of the system memory.	ANY	
SHOW PARTITIONS	Display information on memory partitions.	ANY	
SHOW SCHEDULER	Display the current settings of the scheduling parameters.	PR.SCI	UTL
SHOW SHAREABLE_GLOBAL_AREAS	Display shareable global areas that are bound to each user task.	ANY	
SHOW STATUS	Display information about the current status of the user's job.	ANY	
SHOW SWAPFILES	Display information about specified swap files.	PR.SCI	SWA
SHOW SWITCH_REGISTERS	List the current software switch register settings on a PDP-11/34, /44, or /60.	ANY	
SHOW SYSTEM_TIMES[1]	Display information about current system times.	PR.SCI	N/A.
SHOW TASKS	Display information on tasks in the system.	ANY	
SHOW VOLUMES[1]	List all outstanding load requests for volumes.	PR.SCI	N/A
SHUTDOWN[1]	Terminate timesharing in a specified time.	PR.SCI	N/A
SORT	Sort files into a specified sequence.	PR.FIL	
SPOOL	Control the processing of the output queue.	PR.SCI	OPR

[1]This command is applicable only to timesharing systems.

Table 12–1 (Cont.) Command Summaries

PDS Command	Description	Required Privilege	MCR Command
STOP[/JOB]	Prevent all further processing within a file (used only in an indirect or batch command file).	ANY	
STOP/CLI[1]	Stop the CLI running on the specified terminal(s).	PR.SCI	Not available
STOP/HANDLER	Unload a device handler from the system.	PR.SCI	UNL
SUBMIT	Send one or more specified files (containint batch commands) to the batch processor.	PR.SUB	
TRUNCATE	Truncate a file to the logical end of file.	PR.FIL	
TYPE	Cause the contents of one or more specified files to be printed at the user's terminal.	PR.FIL	
UNFIX	Enable a task to be freed from memory.	PR.RTC	
UNLOCK[/FILE]	Unlock a file that was locked as a result of being improperly closed.	PR.FIL	
UNLOCK/MEMORY	Unlock tasks that were locked in memory when a main memory parity error occurred.	PR.SCI	
USERS	Examine or modify the user profile file (UPF). Available only for users logged in as [1,1].	[1,1]	
VERIFY	Verify the structure of a device.	PR.DEV or PR.SCI	VFY

[1]This command is applicable only to timesharing systems.

12.1.1 Conventions Used in Command Descriptions

Throughout the remainder of this chapter, square brackets ([ ]) are used to surround optional parameters.

For example:

```
SCI> CREATE/SWAPFILE[:n]
```

indicates that the value n is optional.

Additionally, ellipses (...) indicate that a list of values all of the same type could be given in place of a single value. For example, if the description is given as:

```
(terminal 1... terminal n)
```

Then:

```
(TT1:,TT5:,TT6:)
```

is valid (the commas are optional, but either a comma or a space must occur).

12.2 Command Format

Commands must be written as follows:

1. The command name describes the action that the command is to perform. Command names can be abbreviated to the number of characters necessary to make them unique. For example, you can abbreviate ABORT to AB and ALLOCATE to AL. Additional letters are accepted so that ALLO is also a valid abbreviation for ALLOCATE. Note that the system recognizes only the first 12 characters of any command or qualifier.
2. A parameter either describes a value that a command is to use when it executes, or it further defines the action a command is to perform. At least one space must separate the command-name from the first parameter. Parameters are separated from each other by one or more spaces and/or a single comma.
3. A qualifier either modifies the default action of a command or describes a parameter more fully. A qualifier always begins with a slash (/). Both command names and parameters can be qualified. ABORT/REALTIME is an example of a qualified command name. In this case a real-time task is to be aborted.
4. Whenever you enter a command without supplying all the required parameters, the system prompts for missing mandatory parameters. You can, however, enter a command on a single line.

   For example:

   ```
   SCI> INSTALL/CLI:INP
   FILE? [200,200]INPUT
   ```

   has the same effect as:

   ```
   SCI> INSTALL/CLI:INP [200,200]INPUT
   ```

5. An optional parameter will not be prompted for unless the previous line was terminated with ESC. With some commands, you can replace a single parameter with a list of values. Whenever a list is to be given and the parameter is not the last parameter to be entered, the list must be surrounded by parentheses.

   For example:

   ```
   SCI> ALLOCATE/TERMINAL
   TERMINAL? (TT1: TT2: TT5:)
   CLI? PDS
   ```

6. Whenever a command cannot be written on a single line, use a hyphen at the end of the line to indicate that the command will be continued. The system then reprompts on the following line.

   For example:

   ```
   SCI> INITIALIZE/PROTECTION:-
   >(SYSTEM:RWED,GROUP:RW) DU2: -
   >RSXCOR66
   ```

7. If you omit an optional parameter and want to specify subsequent parameters, you must delimit the null parameter by commas from both its neighbors.

   For example:

   ```
   SCI> SET PRIORITY XYZ,,130
   ```

ABORT

FUNCTION

For timesharing systems, the ABORT/CLI and ABORT/JOB commands either terminate the specified CLI or terminate a timesharing job by job number.

PDS VARIATIONS

Use the ABORT/REALTIME and ABORT/TIMESHARING commands to abort either a real-time task or a timesharing task running on the specified terminals. Additionally, you can use ABORT/REGISTER_DUMP as a qualifier to the ABORT/TIMESHARING command. See the *IAS PDS User's Guide* for details.

FORMAT

For timesharing systems:

1 ABORT/CLI

```
SCI> ABORT/CLI
CLI? cli
TERMINAL? terminal-list
```

Where:

- cli = name of the CLI to be aborted.
- terminal-list = a list of terminals in the form (TTn:,...TTm:). Alternately, enter ALL to abort the specified CLI on all terminals.

Do not use ABORT/CLI when a user has devices mounted or allocated, because the devices then become inaccessible. Instead, use the STOP/CLI command.

If you must use ABORT/CLI in such a situation, you must take down timesharing to recover the device(s).

2 ABORT/JOB

```
SCI> ABORT/JOB
JOB NUMBER? job-number
```

Where:

- job-number = 3-digit number that represents the job number of the timesharing task to be aborted. You determine the job number by using the SHOW TASKS/TIMESHARING command (see the *IAS PDS User's Guide*.)

EXAMPLE(S)

For timesharing systems:

1 This example aborts the CLI named PDS on terminals TT1 and TT4.

```
SCI> ABORT/CLI
CLI? PDS
TERMINAL? (TT1,TT4)
```

2 This example aborts the CLI named XYZ on all terminals.

```
SCI> ABO/CLI XYZ ALL
```

3 This example aborts the timesharing task with a job number of 136.

```
SCI> ABORT/JOB 136
```

ALLOCATE

FUNCTION

The ALLOCATE/TERMINAL command enables you to allocate terminals to a CLI.

On a timesharing system, CLIs allocated to a batch stream run automatically in the batch scheduling level unless you specify NOBATCH.

On a multiuser system, the optional parameters are illegal and the system does not prompt for this information.

NOTE: All timesharing terminals are automatically allocated to PDS at timesharing system startup, with the default parameters listed below.

To change any parameters or to allocate the terminal to a different CLI, you must first deallocate the terminal (see the DEALLOCATE command).

PDS VARIATIONS

On timesharing systems:

Use the ALLOCATE/DEVICE command to allocate a device to a terminal. See the *IAS PDS User's Guide* for details.

FORMAT

For non-timesharing systems, the format is as follows:

```
SCI> ALLOCATE/TERMINAL
TERMINAL? terminal-list
CLI? cli
```

For timesharing systems, add the following line:

```
[PARAMETERS? [ (] [RUN] [NOBATCH] [MAX:n]p) ] ]
```

Where:

- terminal-list = Terminal(s) to be allocated to the CLI. ALLOCATE does not allocate any active terminal.

Alternatively, you can specify ALL (on timesharing systems only) to indicate all timesharing terminals.

- cli = CLI to be allocated. To allocate PDS as the CLI on a multiuser system, specify the CLI name as DCL. DCL is the name of a task that provides the interface between the terminal handler and PDX (a special version of PDS for multiuser systems). Do not specify PDS nor PDX as the CLI name on a multiuser system, because the action taken when you type [Ctrl/C] is unpredictable.

For timesharing systems:

- NOBATCH = Indication that the CLI, if allocated to a batch stream, is not to run in the batch scheduling level.
- RUN = Indication that the terminals allocated to the specified CLI are to run immediately; that is, [Ctrl/C] is not needed to activate these terminals.
- MAX:n = Maximum number of job nodes that can be allocated to the terminal. The CLI itself needs a job node, as does each concurrent timesharing task that runs on the terminal. If you do not specify this qualifier, the terminal is given a default value of 255 if the CLI is PDS. In all other cases, the terminal is given a value of 1.

NOTE: The maximum number of tasks a PDS user can run is also limited by the value of the MTS parameter in the USERS command. See Table 7–4 for more information.

EXAMPLE(S)

For timesharing systems:

1 The following example allocates and starts up terminal TT3 to run PDS.

```
SCE> ALLOCATE/TERM
TERMINAL? TT3
CLI? PDS
PARAMETERS? RUN
```

2 The following example allocates CLI CLC to terminals TT1 and TT2 and runs CLC on these terminals. The CLI can run only three subtasks concurrently because one of the nodes is required for CLC itself.

```
SCI> ALLOC/TERM (TT1:,TT2:) CLC (RUN MAX:4)
```

3 The following example allocates MCR to terminal TT1.

```
SCI> ALLOC/TERM
TERMINAL? TT1
CLI? MCR
```

4 The following example allocates DCL to terminals TT1 and TT3 on a multiuser system. This specifies PDS as the CLI for those terminals.

```
SCI> ALLOCATE/TERMINAL (TT1,TT3) DCL
```

ASSIGN

FUNCTION

The ASSIGN/REDIRECT command enables you to redirect all I/O requests from one device unit to another. You can use this command if one of the I/O units needed for a task is inoperable for some reason. If you use this command during timesharing, be very careful.

PDS VARIATIONS

You can also use the ASSIGN command for the following purposes:

For non-timesharing systems:

- To change a LUN assignment for an installed task (ASSIGN/TASK).

For timesharing systems:

- To assign a device to a logical unit (ASSIGN).

See the *IAS PDS User's Guide for details.*

FORMAT

```
SCI> ASSIGN/REDIRECT
FROM? source-device
TO? target-device
```

Where:

- source-device = Device to be redirected.
- target-device = New device unit.

TECHNICAL NOTES

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

If, through a sequence of ASSIGN/REDIRECT commands, you establish a redirect chain which returns to the source-device, an error message is issued and the command is rejected.

EXAMPLE(S)

```
SCI> ASS/RED DU1 DU0
```

This example redirects all I/O requests from DU1 to DU0.

BOOTSTRAP

FUNCTION

The BOOTSTRAP command enables you to stop the present operating system and bootstrap that system or another system from any device. The system must be quiescent before you issue this command (that is, no users are logged in, and no devices are mounted).

FORMAT

```
SCI> BOOTSTRAP
FILE? filespec
```

Where:

- filespec = File specification for the file that contains an IAS saved system image. Table 12–2 contains a list of default values for BOOTSTRAP file specifications.

Table 12–2 Default Values for BOOTSTRAP File Specifications

Qualifier	Default
dev:	SY0:
ufd	The UFD that corresponds to the UIC under which the user logged in, or the UIC specified by a SET DEFAULT[1] or SET UIC command.
filename	IAS
.type	.SAV
;ver	Latest version

[1]This command applies to timesharing systems only.

EXAMPLE(S)

1 The following example rebootstraps the system, using the latest copy of the system image file SYS.SAV that resides in directory file [10,10] on device DU1.

```
SCI> BOOTSTRAP
FILE? DU1:[10,10]SYS.SAV
```

2 The following example rebootstraps the system, using the latest copy of the system image file IAS.SAV in UFD [11,17] on the device SY0:.

```
SCI> BOO
FILE? [11,17]
```

CREATE

FUNCTION

The CREATE/SWAPFILE command enables you to create a swap file. You must specify the size of the swap file.

PDS VARIATIONS

Alternatively, you can use the CREATE command to create a file, a directory, or an alias name for a file. See the *IAS PDS User's Guide* for details.

FORMAT

```
SCI> CREATE/SWAPFILE[:n]
DEVICE? device
PARAMETERS? [(]SIZE:m [DEDICATED_VOLUME] [[NO]BAD_BLOCKS][REALTIME][)]
```

Where:

- n = Number of the swap file to be created. If you omit this number, the file is created as a swap file n + 1, where n is the highest existing swap is the number of the swap file to be created. If you omit this number the file is created as a swap file n + 1, where n is the highest existing swap file. If you specify a number, it must be no more than one greater than the highest existing swap file number. If you specify a number that is already in use, the new file is assigned that number and the existing file and all higher numbered files are incremented by one. You should use the SHOW SWAP_FILES command to check on existing swap file numbers.
- device = Device on which you want the system to create the swap file.
- SIZE:m = Size of the swap file to be created (specified in logical swap blocks, each of 1K words). This parameter is mandatory.
- DEDICATED_VOLUME = Indication that the volume on which the swap file is to be created is a dedicated swap volume. For example, a fixed head disk). In this case, the volume need not contain a Files-11 structure. The device must not be a removable media device and must not contain last-track bad block information. Any bad block information found will be marked allocated and not used for swapping.
- [NO]BAD_BLOCKS = Indicator whether or not the system should use the bad block information (if any) that exists on the volume.
- REALTIME = Indicator that specifies the swap file being created is for the use of real-time tasks only.

EXAMPLE(S)

```
SCI? CREATE/SWAP:5
DEVICE? DB0:
PARAMETERS? SIZE:511
```

This example creates swap file number 5 on device DB0 with a swap file size of 511 logical swap blocks.

DEALLOCATE

FUNCTION

The DEALLOCATE/TERMINAL command enables you to deallocate a CLI from specified terminals.

PDS VARIATIONS

For timesharing systems:

You can use the DEALLOCATE/DEVICE command to deallocate devices. See the *IAS PDS User's Guide* for details.

FORMAT

For non-timesharing systems, the format is as follows:

```
SCI> DEALLOCATE/TERMINAL
TERMINAL? terminal-list
```

For timesharing systems, add the following line:

```
[CLI? cli]
```

Where:

- terminal-list = List of terminals to be deallocated. Note that DEALLOCATE does not deallocate an active terminal.

On timesharing systems only:

You can specify ALL here and give an optional CLI name (by terminating ALL by pressing the <ESCAPE> key). If you do not specify a CLI name, all CLIs on all timesharing terminals are deallocated.)

- cli = CLI name

For timesharing systems:

This is applicable only if you specified ALL above. The CLI is deallocated on all terminals.

EXAMPLE(S)

1 The following example shows how to deallocate TT3: from its CLI.

```
SCI> DEALLOCATE/TERMINAL
TERMINAL? TT3:
```

2 For timesharing systems:

The following example shows how to deallocate all terminals allocated to XYZ.

```
SCI> DEALLOCATE/TERMINAL ALL XYZ
```

DELETE

FUNCTION

The DELETE/SWAPFILE command enables you to delete a specified swap file. If the specified file or any higher numbered files have space allocated for them, the file is marked for deletion and no further space is allocated from those files. The actual deletion occurs when no space is allocated in any of these files.

PDS VARIATIONS

You can also use the DELETE command to delete one or more specified files or a file alias name. See the *IAS PDS User's Guide* for details.

FORMAT

```
SCI> DELETE/SWAPFILE:n
```

Where:

- n = Number of the swap file to be deleted. You can determine this number by using the SHOW SWAP_FILE command.

EXAMPLE(S)

```
SCI> DEL/SWAP:3
```

This example deletes swap file 3 from the system.

EXAMINE

FUNCTION

The EXAMINE command enables you to open (gain access to) a memory address for examination and optional modification.

When you open a location, the specified memory address and the contents of the address are listed on the terminal. Table 12–3 shows the format of this message.

FORMAT

```
SCI> EXAMINE[/qualifier]
ADDRESS? address
```

Where:

- qualifier = One of the following options to specify the address space to which the memory address applies:
  - /KERNEL = Specification of the Kernel virtual address space.
  - /PHYSICAL = Specification of the absolute address space (this is the default).
  - /TASK:taskname = Name of the task whose memory address is to be opened. The task must be fixed in memory.
  - PARTITION:partitionname = Name of the partition whose memory address is to be opened.
  - /TERMINAL:ti = TI identification for a task. This qualifier is valid only for multiuser tasks and must be used in conjunction with the /TASK qualifier.
- address = Octal memory address location to be examined.

The format of the EXAMINE command output follows.

```
memory-address contents/[new-value] line-terminator
```

Where:

- memory-address = 6-digit (octal) virtual address (printed by the system).
- contents/ = 6-digit (octal) value, followed by a slash (/), (printed by the system).
- [new-value] = Value that is to replace contents, followed by one of the optional line terminators (or if only a line terminator is typed, the contents are unaltered), entered by user.

Table 12–3 shows available line terminators.

Table 12–3 Line Terminators

Terminator	Explanation
ESC	ESCape or ALTMODE: end of command. This key is the only means of exit from the EXAMINE function.
RET	Carriage return: the next sequential location is opened.
^ RET	Up-arrow, carriage return: the previous location is opened.
* RET	Asterisk, carriage return: the location pointed to by the final contents of the open location is opened.

EXAMPLE(S)

```
SCI> EXAMINE/TASK:ABC 4
000004 111111/
```

In this example absolute memory location 4 of task ABC is opened for examination. The memory address (000004) and its contents (111111) are printed.

After examining the contents of this address, you can either enter a new value or examine other locations by typing one of the line terminator options shown in Table 12–3.

HELP

FUNCTION

The HELP command displays information at the terminal to give you the information you need to issue further commands.

PDS VARIATIONS

You can also use the HELP command to display information about PDS commands. See the *IAS PDS User's Guide* for details.

FORMAT

```
SCI> HELP [parameters]
```

TECHNICAL NOTES

Information is provided at three different levels:

1. A HELP command without parameters causes the names of all the available commands to be listed.
2. A HELP command followed by a particular command name causes the names of qualifiers and parameters to the command to be listed.
3. A HELP command followed by a command name and the name of a qualifier or parameter provides more detailed information. The qualifiers and parameters that can be given in this case (level 3) are preceded by two asterisks (**) in the level 2 listing.

The HELP command prints a plus sign (+) against all SCI-only commands (see Example 2).

You can make the HELP command display information about user-written tasks by editing the appropriate HELP file. Depending on the type of system and the terminal you use, the HELP files are contained in one of three different text files:

LB0:[1,2]PDS.HLP
LB0:[1,2]PDX.HLP
LB0:[1,2]SCI.HLP

You can also write your own HELP file to display information. The HELP file must be a text file called LB0:[1,2]nnn.HLP, where nnn is your file name. The command to call this file is of the form:

```
> HELP nnn
```

Where:

- nnn = Filename of your HELP file.

Use the existing HELP files as models for your own file. However, the general format of a HELP file is as follows:

```
1 Keyword-1
  Some text-X

1 Keyword-2
  Some text-Y

2 Keyword-2A
  Some text-Z
```

For example, > *HLP nnn Keyword-2* would display "Some text-Y".

NOTE: Uppercase and lowercase characters are used to distinguish whether or not a keyword is unique. However, this does not apply to numbers. Thus, if a keyword contains numbers, you must type them all in. The nesting of keywords can continue up to nine levels.

EXAMPLE(S)

1 The following example shows the HELP command.

```
SCI> HELP

The following commands are available:

 ABort     ALlocate   APpend     ASsign   BAsic     +BOotstrap
 CAncel    COBol      COMpare    CONtinue COpy       CORal
 CReate    DC1        DEALlocate DEASsign DELete     DIRectory
 DISAble   DISMount   DUmp       EDit     ENable     $EOD
 $EOJ     +EXamine    FIx        FOrtran  Goto       Help
 IDentify INItialize  INStall    $Job     LIBrarian Link
 LOAd      LOGOut     MAcro      MCr      MERge      MESsage
 MOunt    +Noload     On         Print    Queue      REMove
 REName    RUn        SAve       SEt      SHOw      +SHUtdown
 SOrt      SPool      STop       SUbmit   TRuncate   Type
 UNFix     UNLock    +Verify

For more information, type HELP followed by keywords.
```

2 The following example displays information about the STOP command. The plus signs (+) indicate that STOP/HANDLER and STOP/CLI are SCI commands.

```
SCI> HELP STOP
     STOP[/JOB]
  +  STOP/CLI [cli-name] terminal-list keyword-list
                                           TIME:n
                                           END_OF_JOB

  +  STOP/HANDLER  device-name
```

INITIALIZE

FUNCTION

The INITIALIZE command enables you to initialize volumes (DOS, RT-11, or Files-11).

The SCI command syntax is identical to the PDS command syntax.

On a timesharing systems:

When you install a volume, the device where the FILES-11 is loaded need not be allocated to the console (SCI) user.

On a multiuser system, no other privilege besides PR.SCI is required to initialize a Files-11 volume.

See the *IAS PDS User's Guide* for details about the command syntax.

INSTALL

FUNCTION

On timesharing systems:

The INSTALL/CLI command enables you to install a CLI in the system, optionally specifying the following parameters:

- Privilege
- Attribute
- Increment
- Partition
- Priority
- Pool limit

PDS VARIATIONS

You can also use the INSTALL command for the following functions:

- To install a task (INSTALL/TASK).
- To install a common area SGA (INSTALL/COMMON).
- To install a resident library SGA (INSTALL/LIBRARY)
- To install an SGA as a region (INSTALL/REGION).
- To install a system library task (INSTALL/SYSTEM).

See the *IAS PDS User's Guide* for further details.

FORMAT

```
SCI> INSTALL/CLI:cli[/qualifiers]
FILE? filespec
```

Where:

- cli = Alphanumeric CLI name from 1 to 3 characters
- qualifiers = Any of the following options:
  - /PRIVILEGE:n

    Where n is the 20-bit CLI privilege mask specified in octal. If you require all privileges, specify ALL. The bit definitions follow. Bits 18, 19, and 20 give extra privileges and are detailed in Table 7–3.

    bits 1-16: enables CLI use of the TCP system facilities.

bit 17: enables a task to issue TCP requests.
bit 18: enables a task to issue real-time privileged directives.
bit 19: enables a task to install or run privileged tasks.
bit 20: enables a task to issue memory management directives.

NOTE: When you install PDS, specify /PRIV:ALL.

- /ATTRIBUTE:n

  Where n is a 3-bit mask for CLI attributes. If you require all attributes, specify ALL. If the CLI being installed is PDS, do not specify this qualifier because PDS itself assigns the appropriate attribute. The bitmask definitions are as follows:

  bit 1: Sets the terminal privileged and logged-in when the CLI is activated on a terminal.
  bits 2-3: Not used—reserved for future use.

- /PARTITION:partition

  Where partition is the specified partition in which to install CLI.

- /INCREASE:increment

  To override the EXTTSK option specified in the LINK command. See the *IAS PDS User's Guide for details.* This qualifier specifies the decimal number of words by which the uppder read/write area of the CLI being installed is to be extended. The value specified is rounded up to the next 32-word boundary.

- /PRIORITY:n

  To set the execution priority to be assigned to the CLI.

- /POOL:n

  To set the pool limit of the CLI to be installed. The pool limit value can range from 0 to 255 decimal and represents the maximum number of 8-word nodes that the task can use at one time.

- filespec

  The file specification for the CLI that you are installing.

EXAMPLE(S)

For timesharing systems:

1 The following example installs the CLI named XYZ from [1,1]XYZ.TSK in a partition named PNAME with all privileges allocated.

```
SCI> INS/CLI:XYZ/PARTITION:PNAME/PRIV:ALL/ATT:1 [1,1]XYZ.TSK
```

2 The following example installs the CLI named PDS from [11,1]PDS.TSK with all privileges allocated.

```
SCI> INS/CLI:PDS/PRIV:ALL [11,1]PDS.TSK
```

LOAD

FUNCTION

For timesharing systems:

You use the LOAD command in response to a user mount request when the operator has loaded the requested volume. If the user requested the mount by giving a device type but no unit number, the system is informed of the unit number selected by the operator when the LOAD command was issued.

The volume label parameter is optional, but the user must specify it if the load request message (see Section 5.4.2) contained a volume label.

FORMAT

```
SCI> LOAD
DEVICE? ddnn[:]
[VOLUME-ID? volume-label]
```

Where:

- ddnn[:] = Device type and unit number where the volume is loaded. Note that the volume should be loaded and device should be in ready status before this command is issued.
- [volume-label] = Label specified in the load request message (see Section 5.4.2). This parameter is optional, but the user can only omit the volume label if no volume label was specified in the load request message. The only time the user need not specify the volume is when the mount command that caused the load request was in the form:

  ```
  PDS> MCR MOU ddnn[:]/OVR
  ```

  See the *IAS MCR User's Guide* for a description of the MCR mode MOU command.

EXAMPLE(S)

For timesharing systems:

1 The following example loads volume V12359 onto any tape drive.

  ```
  LOAD V12359 ON ANY MU
  ```

2 The following example loads volume V12359 on to tape drive MU1:

  ```
  SCI> LOAD MU1: V12359
  ```

  The user has requested a tape drive without specifying a particular unit number (that is, MU:). The operator selects MU1: and loads the volume on that device. When the loading is complete (that is, the device is ready), the user issues the LOAD command as above.

3 The following example loads volume JEN123 on device DU0:. The operator must physically load the volume before issuing the LOAD command.

```
SCI> LOAD
DEVICE? DU0: [ESC]
VOLUME-ID? JEN123
```

4 The following example loads an unspecified volume on device DB1:. The mount command that requested this was MCR MOU DB1:/OVR. In this case, the user must be certain that the correct volume has been physically loaded on DB1.

```
SCI> LOAD MM2: VOL67
```

MESSAGE

FUNCTION

The MESSAGE/FORCE command enables you to force messages to a terminal that has requested not to receive messages. For each message, one of the following destinations must be specified:

- /ALL
- /ACTIVE
- /TERM
- /CLI
- /USER

PDS VARIATIONS

Under PDS, you can use the following options with the MESSAGE command:

- Send a message to all terminals (MESSAGE/ALL).
- Send a message to the terminals specified in the list (MESSAGE/TERMINAL:terminal-list).
- Send a message to the system console (MESSAGE/OPERATOR).

For timesharing systems:

Under PDS, you can use the following options, applicable only to timesharing systems, with the MESSAGE command:

- Send a message to all active terminals (MESSAGE/ACTIVE).
- Send a message to the terminals allocated to the specified CLI (MESSAGE/CLI:cli).
- Send a message to the terminals logged in with a specified user name (MESSAGE/USER:username).

NOTE: /ACTIVE, /CLI:cli, and /USER:username are not available on multiuser systems.

See the *IAS PDS User's Guide* for further details.

FORMAT

```
SCI> MESSAGE/FORCE/qualifier
MESSAGE? message
```

Where:

- /qualifier = One of the following:

  /ALL

/TERMINAL:terminal-list

Where, for timesharing systems:

- /qualifier = One of the following for timesharing systems only:

  /ACTIVE
  /CLI:cli
  /USER:username

(See PDS Variations, above, for details of these options.)

- message = Character string of up to 60 characters containing the desired message.

EXAMPLE(S)

```
SCI> MESSAGE/FORCE/ALL
MESSAGE? BACKUP WILL TAKE PLACE AT 15.30
```

This example sends the above message to all terminals.

MOUNT

FUNCTION

Use the MOUNT command to make a volume available to a user and optionally to associate a logical name with the volume. This command is identical to the PDS MOUNT command except that the /NOOPERATOR qualifier is implicit when issued from the SCI terminal.

FORMAT

```
SCI> MOUNT [/qualifiers]
DEVICE? device-name
VOLUME-ID? volume-id
[LOGICAL NAME? logical-name]
```

See the *IAS PDS User's Guide* for full details on the command syntax.

EXAMPLE(S)

```
SCI> MOUNT DUO: MYDISK
```

This example mounts MYDISK on device DU0:. Note that MYDISK should be physically loaded and DU0: should be ready before you issue this command.

NOLOAD

This routine applies to timesharing systems only.

FUNCTION

Use the NOLOAD command in response to a user load request when the requested load cannot be completed. This might occur because the particular device requested is out of order, because of hardware failure. or because the requested volume cannot be located.

This command is not available on multiuser systems.

FORMAT

```
SCI> NOLOAD
DEVICE? dd[nn] [:]
[VOLUME-ID? volume-label]
```

Where:

- dd[nn] [:] = Requested device. If the mount request did not include a unit number, the unit number can be omitted.
- volume-label = Label of the volume for which a volume load was requested (see Section 5.4.2). This parameter is optional, but you can only omit the volume label if no volume label was specified in the load request.

EXAMPLE(S)

```
SCI> NOLOAD DU: SYS119
```

This example indicates that a volume labeled SYS119 cannot be loaded on a DU drive.

QUEUE

FUNCTION

The QUEUE/TEST command enables you to test for forms alignment. The priority entry is automatically set to 250 and the forms type entry is set to 7.

PDS VARIATIONS

Under PDS, you can use the QUEUE command to perform the following functions:

- Interrogate the queue (QUEUE/LIST).
- Display all entries in all queues (QUEUE/ALL).
- Remove entries from the queue (QUEUE/REMOVE).
- Add entries to the queue (QUEUE/ADD).
- Modify the current status or attributes of a file that is queued for printing (QUEUE/MODIFY).

FORMAT

```
SCI> QUEUE/TEST
[QUEUE? ddnn]
FILE? filespec
```

Where:

- ddnn = Device where the forms alignment is to be tested. If you do not specify the device, the default is LP0:.
- filespec = Name of the test file.

EXAMPLE(S)

```
SCI> QUE/TEST LP0:
FILE? LIST.TST;5
```

This example queues a test file named LIST.TST;5 to be used for testing forms alignment on the line printer LP0:.

REMOVE

FUNCTION

For timesharing systems only:

Use the REMOVE/CLI command to remove an installed CLI from the system. Terminals with the specified CLI allocated must be inactive before you can issue the REMOVE/CLI command.

PDS VARIATIONS

Under PDS, you can use the REMOVE command for the following purposes:

- To remove a task (REMOVE or REMOVE/TASK).
- To remove a common area SGA (REMOVE/COMMON).
- To remove a resident library SGA (REMOVE/LIBRARY).
- To remove an installed region SGA (REMOVE/REGION).

See the *IAS PDS User's Guide* for further details.

FORMAT

For timesharing systems only:

```
SCI> REMOVE/CLI[/NOHEADER]
CLI? cli
```

Where:

- [/NOHEADER] = Optional qualifier to remove a CLI whose task header has been corrupted.
- cli = Name of the CLI to be removed. If any terminals are actively running the CLI, the CLI will remain installed. All terminals allocated to this CLI will be deallocated if the remove operation is successful.

EXAMPLE(S)

```
SCI> REMOVE/CLI BAS
```

Issue this command only when all terminals running BAS as a CLI are inactive. Use the SCI command STOP/CLI BAS ALL TIME:0 to ensure that BAS is inactive. After you specify the REMOVE command, the terminals previously allocated to BAS will no longer be allocated and you cannot use BAS as a CLI.

RUN

FUNCTION

For timesharing systems only:

The RUN command enables you to run a CLI task at a specified terminal (RUN/CLI).

The RUN command also enables you to load a device handler task for a specified device type (RUN/HANDLER). The RUN/CLI command is not available on multiuser systems.

PDS VARIATIONS

Under PDS, you can use the RUN command for three basic purposes:

1 To run a timesharing task (RUN/TIMESHARING).

2 To request a real-time task to be run as soon as memory is available (RUN/REALTIME).

3 To run a real-time task at a specified interval (RUN/SYNCHRONIZE/SCHEDULE/DELAY/INTERVAL).

See the *IAS PDS User's Guide* for further details.

FORMAT

For timesharing systems only:

1 RUN/CLI

```
SCI> RUN/CLI
TERMINAL? terminal-list
[CLI? cli]
```

Where:

- terminal-list = Terminal(s) where the allocated CLI is to run.

  You can specify ALL as the terminal list and give an optional CLI name if terminal-list is ALL (by terminating ALL with an ESC). If you do not specify a CLI name, all CLIs on all timesharing terminals will be run.

- cli = CLI name (only applicable if you specified ALL in the terminal-list parameter). The CLI specified will be run on all terminals where it is allocated.

2 RUN/HANDLER

```
SCI> RUN/HANDLER
DEVICE? device-type[nn]
```

Where:

- device-type = Two-character mnemonic for the device for which a device handler task is to be loaded.

- nn = Unit number (for multiuser handlers only).

  If you specify a unit number and the device handler is multiuser, the handler is run only for the specified unit.

EXAMPLE(S)

1 For timesharing systems only, the following example runs the CLI that has previously been allocated to batch stream BA0.

```
SCI> RUN/CLI BA0
```

2 For timesharing systems only, the following example runs all terminals allocated to XYZ.

```
SCI> RUN/CLI ALL XYZ
```

3 The following example loads the LP handler for device LP2 only.

```
SCI> RUN/HAND LP2
```

SAVE

FUNCTION

You use the SAVE command to record the core image of an IAS system on the disk from which it was originally bootstrapped, so that a bootstrap can reload it and start up the system. Use the command only when the system is quiescent.

For timesharing systems:

Do not save the system while timesharing is running (that is, while the console is prompting SCI>). Instead, use the SHUTDOWN command to shut down timesharing and use the MCR SAV command to save the system. See the *IAS MCR User's Guide* for details about SAV.

FORMAT

```
SCI> SAVE [/qualifier]
```

Where /qualifier = One of the following optional qualifiers:

- /MOUNTED:devlist

  This qualifier enables the system to be saved with the specified devices mounted. See Technical Note 4.

- /LOGGED_ON:terminal-list

  This qualifier enables the system to be saved with the specified terminals logged on.

- /NOEXTEND

  This qualifier inhibits automatic memory expansion or trunctiona when the saved image is rebootstrapped. If you do not specify this qualifer, SAVE determines exactly how much memory exists and expands or truncates the highest partition in the system.

- /NOINSTALL

  This qualifier inhibits the automatic reinstall of tasks and SGAs when the system is bootstrapped. You can use the qualifier to speed up bootstrapping the system as long as no installed task files are to be changed. In particular, if you use the Backup Restore utility (BRU) or the Disk Save and Compress utility (DSC) to compress the disk, the disk can then no longer be bootstrapped if you specified this qualifier.

TECHNICAL NOTES

1 If you attempt to save a system that still has volumes mounted or terminals logged in, and you have not specified the /MOUNTED or /LOGGED_ON switches, the save does not occur.

2 Because the SAVE command provides a copy of a completed system configuration, you must never use it when timesharing is running, and you must only use it when the system is quiescent. Therefore, never use SAVE when timesharing is active. Use the MCR SAV command instead (see the *IAS MCR User's Guide* for details). SAVE ensures that the system is quiescent by searching the system data base for any of the following conditions:

- Mounted devices (except for SY:).
- Users logged onto terminals other than the one from which the SAVE 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 read-write common areas, being recorded on disk.
- Tasks or SGAs installed from a device other than the system disk.
- Tasks loaded or fixed beyond the amount of memory sepcified at system generation (and hence beyond the end of the save file).
- Shareable global areas including the read-only root of multiuser tasks loaded beyond the end of the save file.

If any of these conditions is detected, SAVE issues an error message and the system save does not occur.

3 The SAVE command attempts to record (on the system device) all memory specified at system generation. If more memory exists than was declared at system generation, only the declared memory is saved. If, however, less memory exists than was declared at system generation, only the amount that exists is saved. For a PDP-11/44, 11/70, 11/24, 11/23+, 11/84, 11/53, 11/73, or 11/83, no more than 124K is saved.

4 SAVE does not permit the system to be saved with volumes other than SY: mounted or terminals logged in, unless you specify the appropriate /MOUNTED or /LOGGED_ON qualifiers.

When you specify the /MOUNTED qualifier, for FILES-11 volumes you must take the following information into account:

a. When a volume is mounted, volume control data is established in memory to reflect the volume's current file status. This data is updated with every file operation.

b. When a volume is dismounted, the volume control data is reset.

c. If you elect to save the system with volumes other than SY: mounted, the volume control data for each mounted volume is saved to reflect the current status of the volume. It is imperative that no file activity occur on the volume between the time the system was saved and the next bootstrapping of the system. Otherwise, the integrity of the volume is destroyed.

d. When the system is bootstrapped again, the status of the volume must be exactly the same as it was when the system was saved, or the integrity of the volume is destroyed on the very first file operation.

e. After the system is bootstrapped again and before any file activity has occurred, you can execute the following commands to ensure the integrity of any volume:

```
PDS> DISMOUNT device or MCR>DMO device
PDS> MOUNT device or MCR>MOU device
```

This resets the volume control data to reflect the current volume status.

f. In view of the above, it is strongly recommended that you do not use /MOU for FILES-11 volumes.

When you rebootstrap the system with the /LOGGED_ON qualifier, the specified terminals are automatically logged in with the same UIC and privileges they had when the system was saved.

5 When the system is saved (unless you specified the /NOINSTALL qualifier), disk information for all installed tasks, SGAs, and the swap or checkpoint 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.

Because the file-id identifies a file independent of its physical position on the disk, the files can be rearranged in any way and the system still bootstraps correctly. In particular, after you use the Backup Restore utility (BRU) or the Disk Save and Compress utility (DSC) to compress a system disk, the system disk can still function correctly. This process can take up to several minutes, depending on the type of disk and the number of installed tasks.

6 When a saved system is rebootstrapped or restarted, the system is expanded or truncated automatically to the amount of physical memory available, unless you specified the /NOEXTEND qualifier when the system was saved.

Because SAVE is active when a save is performed, it is active when the reboot/restart occurs. It is, in fact, the task SAVE that restarts an IAS system and performs the memory size calculations.

To expand the system, SAVE determines how much physical memory responds by testing memory in 1K word increments.

If the address at the top of memory is greater than that at the last save, or at system generation if no save was performed, SAVE performs the following steps:

- Increases the size of the last (highest addressed) partition.
- Prints an explanatory message on the terminal where the SAV was initiated.

If the actual memory is less than the amount contained in the save file, SAVE attempts to truncate one or more partitions starting from the highest end of memory. If truncation fails, 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 or T type 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 or T type partitions also can be truncated. Successful truncation results in a descriptive message printed at the terminal.

7 Regardless 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 You can save the system with the system disk mounted, without specifying the /MOUNTED qualifier. In this case, the system disk is automatically mounted after the new system is bootstrapped. You can disable this feature by dismounting the system disk prior to saving the system.

EXAMPLE(S)

1 In the following example, the current status of the system is saved on the disk from which it was originally bootstrapped. System changes made by the ASSIGN/REDIRECT or other PDS commands are also saved with the system memory image.

```
SCI> SAVE
```

2 In the following example, the system is saved with DF0: mounted.

```
SCI> SAVE/MOUNTED:DF0:
```

SET

FUNCTION

The SET command performs the following functions:

1 Sets the forms type on a specified device (SET FORMS).
2 Sets the time and date to be used by the system (SET DAYTIME).
3 Sets memory requirements for testing parity errors and setting memory cache groups on or off (SET MEMORY).
4 To change the characteristics of any terminal (SET TERMINAL).
5 Sets the scheduling parameters (SET ALLOCATION, SET QUANTUM, and SET SERVICE).
6 Sets a terminal privileged (SET PRIVILEGED) or nonprivileged (SET NONPRIVILEGED). These commands can be used only on multiuser systems.
7 Sets the maximum record length for a specified device (SET DEVICES).
8 Sets a device output spooled (SET SPOOL) or to inhibit output spooling on a device (SET NOSPOOL).
9 Sets the software switch registers on a PDP-11/34, PDP-11/44, or PDP-11/60 (SET SWITCH_REGISTER).
10 Sets the maximum extension size of a task (SET EXTENDED_TASK_SIZE/MAXIMUM).
11 For timesharing systems, sets the action of Ctrl/C on specified terminals (SET CONTROLC) or to inhibit the action of Ctrl/C (SET NOCONTROLC).

PDS VARIATIONS

The following PDS variations of the SET command are available:

1 SET [NO]QUIET—Enable or suppress the output of accounting messages.
2 SET PRINTING—Defer printing of spooled files.
3 SET PASSWORD—Change the user's interactive or batch password.
4 SET TERMINAL—Change the characteristics of the user's terminal.
5 SET PROTECTION—Change the protection code of a file.
6 SET PRIORITY—Alter the priority of an active real-time task.
7 SET BOOTSTRAP—Write the bootstrap block on a device.
8 SET UIC—Set the terminal UIC (available only on multiuser systems).

9 SET END_OF_FILE—Specify the end_of_file pointer for a file.

For timesharing systems only:

1 SET DEFAULT—Set a new default device or UFD.

2

3 SET REAL_TIME_CONTROL—Set real-time control.

Additionally, the SET SCI command enables you to execute SCI commands from a nonconsole terminal.

See the *IAS PDS User's Guide* for full details about these commands.

FORMAT

For timesharing systems:

1 SET [NO]CONTROLC

```
SCI> SET
FUNCTION? [NO]CONTROLC
TERMINAL? terminal-list
```

Where:

- [NO]CONTROLC = If you use CONTROLC, the indicated terminals start the allocated CLI in response to Ctrl/C. If you use NO/CONTROLC, the indicated terminals do not start the allocated CLI in response to Ctrl/C. That is, you cannot use the terminal to initiate a new session, but current users are not affected.
- terminal-list = Terminal device name or a list of terminal device names that are to be affected, or all terminal device names (specified as ALL).)

For example:

```
SCI> SET NOCON (TT2:,TT7:)
```

After you issue this command, TT2: and TT7: are not available for future use. If any user is currently using those terminals, the command becomes effective when that user logs out.

2 SET FORMS

```
SCI> SET
FUNCTION? FORMS
QUEUE? device-name
FORMS-TYPE? forms-type
```

Where:

- device-name = Name of the device where the forms type is to be changed.
- forms-type = Forms type number, in the range one to seven. See Section 5.4.5 for further details.

For example:

```
SCI> SET FORMS LP0: 2
```

3 SET DAYTIME

```
SCI> SET
FUNCTION? [DAY]TIME
TIME? dd-mmm-yy hh:mm[:ss]
```

Where:

- dd-mmm-yy = Date to be used by the system. dd digits represent the day of the month; mmm is the 3-character abbreviation for the month; yy digits are the last two digits of the year.
- hh:mm[:ss] = Time to be used by the system. hh digits are the hour; mm digits are the minute; ss digits are the second.

Always set date and time when the system is bootstrapped. System problems can occur if either the date or time are changed arbitrarily. This command is primarily intended to correct an error in either date or time that was not observed during the system startup procedures.

For example:

If the system was started with a date of 31-JUL-78 when the actual date was August 1, 1989, the following command would correct the error:

```
SCI> SET DAYTIME 1-AUG-78
```

4 SET MEMORY

```
SCI> SET
FUNCTION? MEMORY
ATTRIBUTE? attribute value
```

Where attribute and value are any one of the following:

- CACHE n = Enable a cache group to be turned on. n can be either 0 cache group or 1 cache group, depending on which half of the cache area is to be turned on.
- NOCACHE n = Enable a cache group to be turned off. n can be either 0 cache group or 1 cache group, depending on which half of the cache area is to be turned off.
- PARITY n = Set the maximum number of errors enabled in a one-minute period before the Executive turns off the cache group. The system default is 50 errors. You can determine the error log, as described in Chapter 9.

For example:

```
SCI> SET MEMORY CACHE 1
```

This example turns on cache group one.

```
SCI> SET MEMORY NOCACHE 0
```

This example turns off cache group 0.

```
SCI> SET MEMORY PARITY 40
```

This example enables a maximum of 40 errors to occur in a one-minute period, prior to the Executive's turning off the cache group.

5 SET TERMINAL

```
SCI> SET
FUNCTION? TERMINAL [:terminal-list]
ATTRIBUTE? attribute
```

Where:

- terminal-list = Optional list of terminals whose characteristics are to be changed. If you do not specify terminal-list, the default is the user terminal.
- attribute = One of the values listed in the PDS SET TERMINAL command (see the *IAS PDS User's Guide*.

See the *IAS PDS User's Guide* for full details of the SET TERMINAL command syntax. The SCI version of this command is identical, except that the terminal characteristics can be set for any terminal rather than the user terminal only.

6 SET ALLOCATION

```
SCI> SET
FUNCTION? ALLOCATION
TICKS? [time]
BLOCKS? [size]
```

Where:

- time = Number of clock ticks associated with the number of blocks specified. This value is optional.
- size = Allocation size, specified in 32-word memory blocks (see Section 4.2.2). This value is optional.

See the *IAS Performance and Tuning Guide* for guidance on tuning the scheduling parameters.

For example:

```
SCI> SET
FUNCTION? ALLOCATION 40 1
```

7 SET QUANTUM

```
SCI> SET
FUNCTION? QUANTUM/qualifiers
VALUE? nn
```

Where:

- nn = Time factor, specified in clock ticks.
- qualifiers = any of the following:
  - /CONSTANT = Quantum constant. See Section 4.2.2.
  - /LEVEL:m = Set the time factor used for a level, Where:

    m = Scheduling level. See Section 4.2.2.
    nn = Time factor, specified in clock ticks.
  - /BATCH =

For timesharing systems:

Set the quantum for batch. See Section 4.2.3.

NOTE: Do not set the batch quantum to zero except at timesharing system startup.

The *IAS Performance and Tuning Guide* gives information on tuning the scheduling parameters.

For example:

```
SCI> SET
FUNCTION? QUANTUM/BATCH 10
```

This example sets the quantum for batch where the time factor is 10 clock ticks.[1]

```
SCI> SET QUANTUM/CON 1
```

This example sets the quantum constant to 1 clock tick.

```
SCI> SET QUANT/LEVEL:2 5
```

This example sets the time factor for a level. In this example, the scheduling level is 2 and the time factor is 5 clock ticks.

8 SET SERVICE

```
SCI> SET
FUNCTION? SERVICE/qualifier
VALUE? nn
```

Where:

- nn = Time factor in clock ticks.
- /qualifier = Either of the following:
  - /PROMOTION = Set the time interval between scheduler promotions, specified in clock ticks. See Section 4.2.3.
  - /BATCH =

For timesharing systems:

Set the time between batch schedules, specified in clock ticks. See Section 4.2.3.)

See the *IAS Performance and Tuning Guide* for information on tuning the scheduling parameters.

For example:

```
SCI> SET
FUNCTION? SERVICE/BATCH 40[1]

SCI> SET SERVICE/PROMOTION 25
```

9 SET [NO]PRIVILEGED

This command is available only on multiuser systems.

```
SCI> SET
FUNCTION? [NO]PRIVILEGED
TERMINAL? terminal-list
```

[1] This example applies only to timesharing systems.

Where:

- terminal-list = List of terminals to be set privileged or nonprivileged. This command gives the terminal system privileged (needed to run system tasks such as OPE).

For example:

```
SCI> SET PRIV (TT3:,TT4:)
```

This example sets the terminals TT3 and TT4 as privileged terminals.

10 SET DEVICES

```
SCI> SET
FUNCTION? DEVICE:dd[nn] [:]
ATTRIBUTE? WIDTH:n
```

Where:

- dd[nn] [:] = Device name and unit number to be changed to the indicated value specified in n.
- n = Buffer size or maximum record length for the specified device. The device must be record-oriented (for example, a line printer) and must not be a terminal.

For example:

```
SCI> SET DEV:LP0 WIDTH:132
```

This example sets the width of lines on line printer LP0 to 132 characters.

11 SET [NO]SPOOL

Do not issue this command when timesharing is active. Devices cannot be set NOSPOOL while the SP device is mounted.

```
SCI> SET
FUNCTION? [NO]SPOOL
DEVICE? device
```

Where:

- [NO]SPOOL = Sets the indicated device as an input or output spooled device. When you specify NOSPOOL, spooling is inhibited on the indicated device.
- device = Name of the device to be set as a spooled device, or for which spooling is to be inhibited.

For example:

```
SCI> SET SPOOL LP1:
```

This example sets line printer LP1 as an output spooled device.

12 SET SWITCH_REGISTER

```
SCI> SET
FUNCTION? SWITCH_REGISTEr
TO? parameter
```

Where parameter is one of the following:

- ON:(switch-list) is of the form m[,n,...], which are decimal numbers representing the bits to be set to 1, leaving the other bits unchanged.

- OFF:(switch-list) is of the form m[,n,...], which are decimal numbers representing the bits to be cleared, leaving the other bits unchanged.
- VALUE:num is the absolute octal value in the software switch register.

NOTE: This function is applicable only on a processor supplied without hardware switches; it is ignored on any other processor.

For example:

```
SCI> SET SWITCH_REG ON:(13,14)
```

This example sets bits 13 and 14 in the software switch register.

```
SCI> SET SWIT VALUE: 177777
```

This example sets all bits in the software switch register.

13 SET EXTENDED_TASK_SIZE[/MAXIMUM]

```
SCI> SET
FUNCTION? EXTENDED_TASK_SIZE[/MAXIMUM]
TO? size
```

Where:

- size = Overall system maximum size to which any task can extend itself using the EXTK$ directive (see the *IAS System Directives Reference Manual*. It can take the following forms:
  - nnnn is an octal number of 32-word blocks. The value must be less than or equal to 2000 octal.
  - nnK is a decimal number of K words. The value must be less than or equal to 32K.

To set the maximum size for an individual task, specify the option MAXEXT to the task builder (MCR TKB or PDS LINK). See the *IAS Task Builder Reference Manual* for details.

For example:

```
SCI> SET EXTENDED_TASK_SIZE/MAXIMUM 20
```

This example sets the task extension size to 20 octal blocks.

```
SCI> SET EXTENDED_TASK_SIZE 20K
```

This example sets the task extension size to 20K words.

SHOW

FUNCTION

The SHOW command enables you to display the following:

1 Information on the current settings of the scheduling parameters (SHOW SCHEDULER).
2 Information on all swap files or a specified swap file (SHOW SWAP_FILES).

For timesharing systems only:

1 Information about current system times (SHOW SYSTEM_TIMES).
2 Information on all outstanding load requests for a volume (SHOW VOLUMES).

PDS VARIATIONS

Under PDS, the following variations of the SHOW command are also available:

1 SHOW TASKS—Display information about tasks (for example, installed, active, or fixed tasks).
2 SHOW STATUS—Display information about the current status of the user job.
3 SHOW PARTITIONS—Display information about partitions on the system.
4 SHOW MEMORY—Display the use of the system memory (VDU terminal only).
5 SHOW LUNS—Display current LUN assignments for an installed task.
6 SHOW GLOBAL_AREAS—Display information about resident global areas.
7 SHOW DEVICES[/PUD]—Display information about all or selected devices known to the system.
8 SHOW DAYTIME—Display the date and time.
9 SHOW IO_QUEUES—Display I/O request queues.
10 SHOW CLOCK_QUEUE—Display the system clock queue.
11 SHOW SHAREABLE_GLOBAL_AREAS—Display shareable global areas.
12 SHOW SWITCH_REGISTERS—Display current setting of the PDP-11/34, PDP-11/44, or PDP-11/60 switch registers.
13 SHOW EXTENDED_TASK_SIZE[/MAXIMUM]—Display the current task extension size.

For timesharing systems only:

1 SHOW DEFAULT—Display the current user default device and UPD.
2 SHOW CLI—Display the CLIs installed in the system.

FORMAT

1 SHOW SCHEDULE

```
SCI> SHOW
ATTRIBUTE? SCHEDULER
```

This format displays the current settings of the scheduling control parameters. To find out the distributed default values for the scheduling parameters, see the *IAS Performance and Tuning Guide*.

2 SHOW SWAP

```
SCI> SHOW
ATTRIBUTE? SWAP_FILES [:n]
```

Where:

n = Number of the swap file to be displayed. If you do not specify a number, all swap files are displayed.

For timesharing systems only:

1 SHOW SYSTEM TIMES

```
SCI> SHOW
ATTRIBUTE? SYSTEM_TIMES
```

This format displays information about current system times. The system accumulates time in ticks and converts them to hours, minutes, seconds, and tenths of a second for this display. The tenth of a second column is rounded down.

2 SHOW VOLUMES

```
SCI> SHOW
ATTRIBUTE? VOLUMES
```

This format lists all outstanding volume load requests.

EXAMPLE(S)

1 SCI> SHOW SCHEDULER

```
SCHEDULER PARAMETERS
 SCHEDULER ENABLED IN PARTITION GEN   , SPACE 20015
 TIMESHARING PRIORITY: 100
 TIME BETWEEN SCHEDULER PROMOTIONS: 1344 CLOCK TICKS
 PROMOTION TABLE SIZE: 8
 MAXIMUM TIME SLICE: 25 CLOCK TICKS
 SYSTEM IDLE TIME: 1 CLOCK TICKS
 MAXIMUM TASK SIZE: 1024 32-WORD BLOCKS
BATCH PARAMETERS:
 BATCH QUANTUM: 25 CLOCK TICKS
 TIME BETWEEN BATCH SCHEDULES: 6000 CLOCK TICKS
QUANTUM PARAMETERS:
 QUANTUM CONSTANT: 2 CLOCK TICKS
 ALLOCATION FACTOR: 1 TICKS PER 128 MEMORY BLOCKS
 NUMBER OF SCHEDULING LEVELS: 4
   LEVEL: 1 TIME FACTOR: 1 CLOCK TICKS
   LEVEL: 2 TIME FACTOR: 2 CLOCK TICKS
   LEVEL: 3 TIME FACTOR: 4 CLOCK TICKS
   LEVEL: 4 TIME FACTOR: 8 CLOCK TICKS
```

2 SCI> SHOW SWAP_FILES

```
# DEV  FILE       SIZE  USED  HOLE  FLAGS

1 SY0:  001043,000011  30.    8.   12.   RT
2 DS1:  000000,000000  50.    0.   50.   RT.DV
3 SY0:  000342,000001 200.  176.   14.
4 DS0:  000000,000000 255.  103.   82.   DV.BAD
5 DS1:  000000,000310 200.    8.    0.   DV.DEL
```

In this example, the headings denote the following:

- # = Swap file number, which always starts at 1 and increments by one at a time. This is the number you must use when you delete a file or obtain a selective listing.
- DEV = Device where the swap file is located.
- FILE = For a file on a Files-11 volume, the File-id of the swap file. For a dedicated volume, the starting logical block number (LBN) of the swap file.
- SIZE = Size in 1K swap blocks used in this file.
- USED = Number of swap blocks used in this file.
- HOLE = Size of the largest contiguous area in the swap file.
- FLAGS = Other information, as follows:

  RT = File reserved for real-time tasks.
  DV = File on a dedicated swap volume.
  DEL = File marked for deletion. (In this case, HOLE is always zero.)
  BAD = File on a dedicated volume contains one or more bad blocks that cannot be allocated as swap space.

The following examples apply to timesharing systems only.

3 SCI> SHOW SYSTEM_TIMES

```
TIME SINCE STARTUP: 49:47.36
TIMESHARING: 49:47.36
NO JOBS TO RUN: 47:02.96
INTERACTIVE USERS: 1:46.96
SWAPPING: 2.55
BATCH: 0.00
```

This example shows information displayed about the current system times, as follows:

TIME SINCE STARTUP: TIMESHARING:	Always have the same value. These times display the time since the SET START timesharing system startup command was issued.
NO JOBS TO RUN:	Displays the time during which the system had no runnable timesharing jobs.
INTERACTIVE USERS:	Displays the time during which timesharing (including batch) jobs were running.
SWAPPING:	Displays the time during which the system was unable to run jobs because of loading or swapping of tasks.
BATCH:	Displays the time taken by batch jobs running in the batch level.

All times shown are calculated from the scheduler accounting values. These values are held in ticks and are converted to hours, minutes, seconds, and tenths of a second for this display. The tenth of a second column is rounded down.

Adding together NO JOBS TO RUN, INTERACTIVE USERS, and SWAPPING, gives the TIME SINCE STARTUP. Any resulting discrepancy occurs because tenths of a second are rounded down.

4 SCI> SHOW VOLUMES

This example shows information displayed about the current volumes.

```
OUTSTANDING LOAD REQUESTS
DU0: WITH PAYROLL
DU1: WITH MARTSDISK
MT0: WITH FABTAPE
```

SHUTDOWN

FUNCTION

This applies to timesharing systems only.

The SHUTDOWN command terminates timesharing after a specified time. It causes CLI tasks to be requested to exit. If a user fails to LOGOUT, all tasks are aborted and the terminal is automatically logged out when shutdown becomes due.

FORMAT

```
SCI> SHUTDOWN
TIME? time
```

Where:

- time = Time (in minutes) within which the CLIs are to exit.

TECHNICAL NOTES

You cannot log out the console terminal once a shutdown command has been issued.

Be careful when you issue a shutdown command after a shutdown command was issued previously. Make sure that subsequent shutdown commands specify less time than is left until shutdown.

EXAMPLE(S)

- The following example initiates system shutdown procedures immediately.

```
SCI> SHUTDOWN
TIME? 0
```

- The following example initiates system shutdown procedures in five minutes.

```
SCI> SHUTDOWN
TIME? 5
```

SPOOL

FUNCTION

The SPOOL command enables you to control the processing of the output queue.

FORMAT

```
SCI> SPOOL/qualifier
[QUEUE? device-name]
```

Where:

- qualifier = One of the following:
  - /START—Initiates output on the specified device of all files whose forms type match the forms type currently specified for this device.
  - /ABORT—Aborts the output of the current file on the specified device. The file is removed from the queue. The next eligible file in the queue is then dispatched. Even if the file has been queued with QUEUE/DELETE qualifier (see the *IAS PDS User's Guide*), the file is preserved.
  - /STOP—Suspends the output of the current file on the specified device. You can resume output by specifying /RESUME or you can abort output by specifying /ABORT in a subsequent SPOOL command.
  - /TEST—Sets the forms type of the specified device to 7. When output of the current file is completed, output is suspended. New forms can be mounted and a test file queued (with a subsequent QUEUE command) to test forms alignment.
  - /RESUME—Restarts output of the file from the current position.
  - /RESUME:FORM—Resumes output of the current file from the last encountered form feed. If none was encountered, output begins with the first record in the file.
  - /RESUME:FILE—Restarts the output of the current file from the beginning of the file.
  - /RESUME:RECYCLE—Renders all queue requests for the specified device inactive. It then dispatches queued files whose forms type match the forms type currently sepcified for this device. Use this qualifier 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 output.
- [device-name] = Name of a device where output is processed. It is in the form ddnn: where dd is the device-type and nn is the unit number. The default is LP0:.)

EXAMPLE(S)

- With the following example, the printer starts printing files queued for LP0: whose forms types match the forms type currently specified for the line printer.

```
SCI> SPOOL/START LP0:
```

- With the following example, the print job for the file currently being printed on LP0: is aborted.

```
SCI> SPOOL/ABORT
```

- With the following example, the printer resumes printing the current file on TT3: from the last encountered form feed.

```
SCI> SPOOL/RESUME:FORM TT3:
```

STOP

FUNCTION

The STOP command has two functions:

1 It stops a device handler.

2 For timesharing systems only, it stops a CLI (STOP/CLI).

STOP/CLI stops the CLI running on the specified terminals within a specified time. The CLI is initially requested to exit and, if it is still running when the time has expired, timesharing tasks are aborted and the CLI is automatically logged out.

If a CLI on an interactive terminal becomes inactive before the specified time (for example, through logging out), only the system manager can activate that CLI.

On a batch terminal, you can specify that batch processing for that stream stops after it has finished the current batch job by means of the END_OF_JOB keyword.

You cannot stop the CLI on the console.

PDS VARIATIONS

The STOP command can also be used in PDS to stop all further file processing within a command file (STOP/JOB).

See the *IAS PDS User's Guide* for further details.

FORMAT

1 STOP/HANDLER

```
SCI> STOP HANDLER
DEVICE: dd[nn]
```

Where:

- dd = Name of the device whose handler is to be stopped.
- nn = Unit number of the device whose handler is to be stopped. You must only specify the unit number for multiuser handler tasks. If you do not specify nn, the handler is stopped for all units.

2 For timesharing systems only: STOP/CLI

```
SCI> STOP/CLI
[CLI? cli]
TERMINAL? terminal-list[/quals]
WHEN? keyword-list
```

Where:

- cli = Name of the CLI running on the terminals to be stopped. If the CLI name is omitted, the CLI running for that terminal is stopped.
- terminal-list = List of terminals where the CLI is running. To stop the CLI on all terminals, type ALL.
- /quals = One of the following qualifiers:
  - ALL_TERMINALS stops the CLI on all terminals.
  - BATCH_TERMINALS stops the CLI on all batch terminals.
  - INTERACTIVE_TERMINALS stops the CLI on all interactive terminals.
- keyword-list = One or both of the following keywords:
  - END_OF_JOB stops processing for the batch stream at the end of the current batch job.
  - TIME:n is the time limit in minutes, where n is the number of minutes (decimal).

    If you specify either keyword for a batch stream that has no active batch job running, that stream becomes inactive immediately.

    If you specify END_OF_JOB for a batch stream that has an active job running, that batch stream becomes inactive when that batch job finishes.

    If you specify the TIME:n keyword, (or use the SHUTDOWN command), the batch stream remains active until the time limit has expired, whether the current job finishes before the time limit or not.

    If you specify both TIME and END_OF_JOB for a batch stream, the stream becomes inactive after that time or at the end of the job, whichever comes first.

STOP CLI is similar to the SHUTDOWN command, except that here shutdown is activated only for the specified terminal list. The terminal(s) for which the CLI is to stop receive shutdown messages.)

EXAMPLE(S)

The following examples apply to timesharing systems.

1 The following example stops PDS from running on TT3: in one minute.

```
SCI> STOP/CLI PDS TT3: TIME:1
```

2 The following example stops PDS from running on terminals TT10:, TT11;, and TT12: in five minutes.

```
SCI> STOP/CLI [ESC]
CLI? PDS
TERMINAL? (TT10: TT11: TT12:)
WHEN? TIME:5
```

3 The following example stops the CLIs that are running for all terminals in three minutes.

```
SCI> STOP/CLI
TERMINAL? ALL
WHEN? TIME:3
```

4 The following example stops batch processing on all batch terminals allocated to PDS at the end of their current batch jobs.

```
SCI> STOP/CLI [ESC]
CLI? PDS
TERMINAL? ALL/BATCH_TERMINALS
WHEN? END_OF_JOB
```

5 The following example stops the RK05 handler from running in the system.

```
SCI> STOP/HANDLER
DEVICE? DU
```

6 The following example stops the line printer handler on unit 2.

```
SCI> STOP/HANDLER LP2
```

UNLOCK

FUNCTION

The UNLOCK/MEMORY command enables you to unlock tasks that the system locked in memory when a main memory parity error occurred.

PDS VARIATIONS

You can also use the UNLOCK/FILE command to unlock a file that has been improperly closed. See the *IAS PDS User's Guide* for details.

FORMAT

```
SCI> UNLOCK/MEMORY
TASK? taskname
[TERMINAL? terminal]
```

Where:

- taskname = Name of the task to be freed.
- terminal = Name of the terminal for which the task was active, or can be ALL to indicate all terminals.

EXAMPLE(S)

```
SCI> UNLOCK/MEMORY TNAME TT7:
```

This example unlocks the memory that was locked because of a parity error for the task TNAME running from TT7.

A Format of the User Profile File (UPF)

Offset	Contents
0	No. of 1st Profile Record
2	No. of last Profile Record
4 6	1401 0
10	200 200
12 14	"DEFPAS"
16	0
20	0
22	0
24 26	077777 177777
30 32	077777 177777
34 36	077777 177777
40 42	077777 177777
44	077777
46 50	0 0
52 54	0 0
56 60	"SY0:"
62	077777
64 66	0 0
70	3 0
72	0
74	reserved
76	reserved

Default User Profile
Index Record 1
Index Record 2
Index Record 3
. . .
Index Record n

Offset	Contents
0 2 4 6	UP.USN
10	UP.UIC
12 14	UP.PAS
16	0
20	UP.PRI
22	UP.UC
24 26	UP.RCA
30 32	UP.RCS
34 36	UP.RUA
40 42	UP.RUS
44	UP.RTT
46 50	UP.UCA
52 54	UP.UUA
56 60	UP.DEV
62	UP.BPR
64 66	UP.BPW
70	UP.MTS UP.TP1
72	UP.TP2
74	reserved
76	reserved

User Profile Record 1
User Profile Record 2
User Profile Record 3
. . .
User Profile Record m

KEY

UP.USN	Username
UP.UIC	UIC
UP.PAS	Password
UP.UC	Number of Users logged in
UP.RCA	Connect time limit for accounting period
UP.RCS	Connect time limit for session
UP.RUA	System utilization limit for accounting period
UP.RUS	System utilization limit for session
UP.RTT	Task run time limit (not enforced)
UP.UCA	Total connect time used in accounting period
UP.UUA	Total system utilization used in accounting period
UP.DEV	User's default device
UP.TP1	Task privilege
UP.TP2	Task privilege
UP.MTS	Maximum number of tasks
UP.BPR	Batch command privilege
UP.BPW	Batch password
UP.PRI	Interactive Command Privilege

B CDA Sample Listing

This appendix contains three sample CDA listings, produced by issuing the following commands:

```
CDA>/ALL
CDA>/SGA:HNDLIB
CDA>/RAK:APE...
```

NOTE: The listings are taken from several crash dumps to illustrate examples of various situations. Therefore, ignore all dates and page numbers at the top of each page.

C Error Logging Sample Output and Data Record Formats

This appendix contains sample output from error logging (see Chapter 9) and the format of records in the ERROR.SYS file produced by the preanalyzer (PRE).

C.1 Sample Error Logging Output

Example C–1 System Error Report Summary

```
************************************************************
ERROR LOGGING SYSTEM STARTED
AT 14-NOV-89 13:54:42
************************************************************

************************************************************
TAPE HARDWARE ERROR
LOGGED AT 15-NOV-89 09:46:28            ERROR NUMBER 1.
************************************************************

TAPE PARAMETERS
        UNIT NAME               MM1
        VOLUME LABEL
        VOLUME OWNER UIC        1,1
        DEVICE TYPE             TU16 UNIT-1 CONTROLLER-0

TAPE REGISTERS AT ERROR TIME
        MTCS1           144260
        MTWC            000000
        MTBA            136260
        MTFC            000000
        MTCS2           000100
        MTDS            150761
        MTER            100000
        MTAS            000001
        MTCK            000100
        -               000000
        MTMR            000200
        MTDT            142054
        MTSN            014225
        MTTC            102301
        MTEAB           000004
        MTCS3           002000

ERROR DIAGNOSIS
RECOVERED

RETRIES PERFORMED       0.

SYE V03.00   SYSTEM ERROR REPORT COMPILED AT 13-APR-90 13:28:09 PAGE 10.
```

Example C–1 Cont'd on next page

Example C–1 (Cont.) System Error Report Summary

```
***********************************************************
DISK HARDWARE ERROR
LOGGED AT 16-NOV-89 13:12:13          ERROR NUMBER 1.
***********************************************************

DISK PARAMETERS
        UNIT NAME               DB1
        VOLUME LABEL
        VOLUME OWNER UIC        1,1
        DEVICE TYPE             RP06 UNIT-1 CONTROLLER-0

DISK REGISTERS AT ERROR TIME
        RPCS1           005310
        RPWC            000000
        RPBA            032300
        RPDA            000403
        RPCS2           000101
        RPDS            000700
        RPER1           000000
        RPAS            000000
        RPLA            002560
        -               000000
        RPMR            000400
        RPDT            024022
        RPSN            120044
        RPOF            110000
        RPDC            000643
        RPCC            000000
        RPER2           000000
        RPER3           000000
        RPEC1           000000
        RPEC2           000000
        RPBAE           000006
        RPCS3           002100

ERROR DIAGNOSIS
NON RECOVERED

RETRIES PERFORMED       0.
```

Example C–2 T/MSCP Error Report

```
 IAS MSCP ERROR LOG REPORT  3-APR-90 15:00:38  LB:[1,6]ERRDSA.TMP  PAGE 1

 Date of entry:    13-MAR-90 16:35:10
 Message envelope: 000070 000020
 Message Packet:

        Word 0 Word 1 Word 2 Word 3 Word 4 Word 5 Word 6 Word 7
        000000 147244 000002 000000 040403 000353 154104 047314
        000201 000406 000004 000002 064410 000000 000000 001005
        002007 000000 000000 000000 000000 000000 000023 000200
        105400 161400 001404 000000
```

Example C–2 Cont'd on next page

Example C–2 (Cont.) T/MSCP Error Report

```
 Connection ID = 000000  Message type = 000020
 Command Reference number = 147244000000
 MSCP Packet Format = Standard Disk Interface Error
 Error Log message flag byte = 000101
 Status/Event code = 000353

    Drive Detected Error

 Controller Model = 000006 = UDA50A
 Controller Serial Number = 154104 047314 000201
 Controller Hardware Version = 0.
 Controller Software Version = 4.
 Drive number = 2.

 Drive Serial number = 064410 000000 000000
 Drive Model = 5.= RA81
 Drive Class = 2. = Disk
 Drive Hardware Version = 4.
 Drive Software Version = 7.

 Volume Serial number = 0.
 Logical Block Number (Header Code) = 0.

**** Drive Supplied Information ****

 Drive Status/Error Request Byte = 000023
      * Drive ONLINE or AVAILABLE
      * Drive requires no RECALIBRATE command
      * No diagnostic being requested
      * Spindle Ready
      * No logable information in extended status
      * Port A selected
      * Port Switch IN
      * Run/Stop Switch IN

 Drive Status/Error Mode Byte = 000000
      * Drive ENABLED by controller diagnostic
      * Formatting Operations DISABLED
      * Diagnostic Block Access DISABLED
      * 512 Byte Sector Format

IAS MSCP ERROR LOG REPORT  3-APR-90 15:00:38  LB:[1,6]ERRDSA.TMP  PAGE 2
 Drive Status/Error Error Byte = 000200
      * "DE"  Error Detected in Drive

 Drive Status\Error Controller Byte = 000000
      * 0 = Normal Operation

 Drive Internal Retry Count = 0.
 Last SDI command to Drive Op-Code = 000213 = BRING DRIVE ON-LINE
 SDI Error Byte = 000000
 Front Panel Fault Code = 0.
 Drive Error LED Code = 0.
 Current  Cylinder = 1251.
 Current Select Group = 3.

 Date of entry:    13-MAR-90 16:38:14
 Message envelope: 000054 000020
 Message Packet:
```

Example C–2 Cont'd on next page

Example C–2 (Cont.) T/MSCP Error Report

```
       Word 0 Word 1 Word 2 Word 3 Word 4 Word 5 Word 6 Word 7
       113160 004215 000002 000000 040402 000213 154104 047314
       000201 000406 000004 000002 064410 000000 000000 001005
       002007 000000 030071 000000 000000 000000

 Connection ID = 000000  Message type = 000020
 Command Reference number = 004215113160
 MSCP Packet Format = Disk Transfer Error
 Error Log message flag byte = 000101
 Status/Event code = 000213

    Lost Read/Write Ready During Or Between Transfers

 Controller Model = 000006 = UDA50A
 Controller Serial Number = 154104 047314 000201
 Controller Hardware Version = 0.
 Controller Software Version = 4.
 Drive number = 2.

 Drive Serial number = 064410 000000 000000
 Drive Model = 5.= RA81
 Drive Class = 2. = Disk

IAS MSCP ERROR LOG REPORT  3-APR-90 15:00:38  LB:[1,6]ERRDSA.TMP  PAGE 4
 Drive Hardware Version = 4.
 Drive Software Version = 7.

 Volume Serial number = 12345.
 Logical Block Number (Header Code) = 0.
 Error Retry Count = 0.
 Error Recovery Level = 0.
********************************************************************************

 Date of entry:    2-APR-90 14:20:10
 Message envelope: 000054 000020
 Message Packet:

       Word 0 Word 1 Word 2 Word 3 Word 4 Word 5 Word 6 Word 7
       122100 013473 000001 000000 040402 000213 154104 047314
       000201 000406 000004 000003 065437 000000 000000 001023
       110415 000000 056134 034202 000000 000000

 Connection ID = 000000  Message type = 000020
 Command Reference number = 013473122100
 MSCP Packet Format = Disk Transfer Error
 Error Log message flag byte = 000101
 Status/Event code = 000213

    Lost Read/Write Ready During Or Between Transfers

 Controller Model = 000006 = UDA50A
 Controller Serial Number = 154104 047314 000201
 Controller Hardware Version = 0.
 Controller Software Version = 4.
 Drive number = 1.

 Drive Serial number = 065437 000000 000000
 Drive Model = 19.= RA90
 Drive Class = 2. = Disk
 Drive Hardware Version = 145.
 Drive Software Version = 13.
```

Example C–2 Cont'd on next page

Example C–2 (Cont.) T/MSCP Error Report

```
Volume Serial number = 948067420.
Logical Block Number (Header Code) = 0.
Error Retry Count = 0.
Error Recovery Level = 0.
********************************************************************************

Date of entry:    29-MAR-90 09:23:52
Message envelope: 000070 000020
Message Packet:

       Word 0 Word 1 Word 2 Word 3 Word 4 Word 5 Word 6 Word 7
       123440 053557 000006 000000 040404 000153 015652 000000
       123424 000403 002410 000736 015652 000000 122424 001002
       003400 003055 000000 000000 001413 017000 000000 000013
       001467 000011 000007 020275

Connection ID = 000000  Message type = 000020
Command Reference number = 053557123440
MSCP Packet Format = Small Disk Error
Error Log message flag byte = 000101
Status/Event code = 000153

   Positioner Error (misseek)

Controller Model = 000003 = KLESI+RC25
Controller Serial Number = 015652 000000 123424
Controller Hardware Version = 5.
Controller Software Version = 8.
Drive number = 6.

Drive Serial number = 015652 000000 122424
Drive Model = 2.= RC25
Drive Class = 2. = Disk
Drive Hardware Version = 7.
Drive Software Version = 0.

Volume Serial number = 0.
Cylinder = 1581.
IAS MSCP ERROR LOG REPORT  3-APR-90 13:49:20  LB:[1,6]ERRDSA.TMP  PAGE 1

--------------------------------------------------------------------------------
Error Logging System STARTED at 30-MAR-90 14:57:23
--------------------------------------------------------------------------------


Date of entry:    30-MAR-90 14:57:24
Message envelope: 000030 000420
Message Packet:

       Word 0 Word 1 Word 2 Word 3 Word 4 Word 5 Word 6 Word 7
       000000 000000 000000 000000 000400 000012 000000 000000
       000000 001411 003404 000013

Connection ID = 000001  Message type = 000020
Command Reference number = 000000000000
MSCP Packet Format = Controller Error
Error Log message flag byte = 000001
Status/Event code = 000012
```

Example C–2 Cont'd on next page

Example C–2 (Cont.) T/MSCP Error Report

```
Controller Model = 000011 =  TQK50
Controller Serial Number = 000000 000000 000000
Controller Hardware Version = 7.
Controller Software Version = 4.
```

C.2 Format of Records in ERROR.SYS File

NOTE 3

DEVICE TYPES ARE FURTHER DEFINITION OF THE DEVICE CLASSES. THEREFORE A WHOLE SERIES OF DEVICE TYPE NUMBERS EXIST WITH EACH DEVICE CLASS.

THE FOLLOWING DEVICE TYPES ARE PRESENTLY DEFINED BY DEVICE CLASS:

```
DEVICE TYPE CODES FOR DEVICE CLASS 001  (DISK)
000         UNDEFINED
001         RK05
002         RP03
003         RF11
004         RS04
005         RS03
006         RP04
007         RP02
010         RK06
011         RP05
012         RP06
013         RK05F
014         RK03
015         RX01
016         RM02/03
017         RL01/02
020         RX02
021         RP07
022         RM05
023         RM80

DEVICE TYPE CODES FOR DEVICE CLASS 002  (TAPES)
000         UNDEFINED
001         TU56
002         TU10/TE10
003         TU16/TE16
004         TU11
005         Reserved
006         TS11
007         TU58

DEVICE TYPE CODES FOR DEVICE CLASS 200  (PROCESSOR)
000         UNDEFINED
001         11/45
002         11/70
```

D 11/70 Parity Error Response

Source Of Cycle	Cycle Type	Type of Error	ACTION * All Traps Enabled	Disable Warning Traps (CR00 1)	Disable all Traps (CR 01:00 3)	Bits Set in Error Register
	DATI/P	MAIN BUS TIMEOUT	ABORT	ABORT	ABORT	15,11,8
		MAIN BUS PARITY	ABORT	ABORT	ABORT	15,11,1
		MAIN MEM WANTED WORD	ABORT	ABORT	ABORT	15,11,2 or 3
CPU to CACHE		MAIN MEM OTHER WORD	TRAP	NO ACTION	NO ACTION	11,2 or 3
		FAST MEM ADDRESS	TRAP	NO ACTION	NO ACTION	11,4 or 5
		FAST MEM DATA	TRAP	NO ACTION	NO ACTION	11,6 or 7
	DATO/B	MAIN BUS TIMEOUT	ABORT	ABORT	ABORT	15,11,0
		MAIN BUS PARITY	ABORT	ABORT	ABORT	15,11,1
	DATI/P	MAIN BUS TIMEOUT	UNIBUS TIMEOUT	UNIBUS TIMEOUT	UNIBUS TIMEOUT	NONE
		MAIN BUS PARITY	UNIBUS TIMEOUT	UNIBUS TIMEOUT	UNIBUS TIMEOUT	10,1
		MAIN MEM WANTED WORD	UNIBUS PB TRAP	UNIBUS PB, TRAP	UNIBUS PB	13,10,2 or 3
UNIBUS through MAP to CACHE		MAIN MEM OTHER WORD	TRAP	NO ACTION	NO ACTION	10,2 or 3
		FAST MEM ADDRESS	TRAP	NO ACTION	NO ACTION	10,4 or 5
		FAST MEM DATA	TRAP	NO ACTION	NO ACTION	10,6 or 7
	DATO/B	MAIN BUS TIMEOUT	UNIBUS TIMEOUT TRAP	UNIBUS TIMEOUT	UNIBUS TIMEOUT	10,8
		MAIN BUS PARITY	UNIBUS TIMEOUT TRAP	UNIBUS TIMEOUT	UNIBUS TIMEOUT	10,1
		FAST MEM ADDRESS	TRAP	NO ACTION	NO ACTION	10,4 or 5
	DATI/P	MAIN BUS TIMEOUT	UNIBUS TIMEOUT ABORT TO 4	UNIBUS TIMEOUT ABORT TO 4	UNIBUS TIMEOUT	NONE
CPU to UNIBUS through	MAIN BUS PARITY		UNIBUS TIMEOUT ABORT TO 4	UNIBUS TIMEOUT ABORT TO 4	UNIBUS TIME- OUT, ABORT TO 4	10,1
MAP to CACHE		MAIN MEM WANTED WORD	UNIBUS PARITY ABORT TO 114	UNIBUS PARITY ABORT TO 114	UNIBUS PARITY ABORT TO 114	13,10,9,2 or 3
		MAIN MEM OTHER WORD	TRAP	NO ACTION	NO ACTION	10,2 or 3
		FAST MEM ADDRESS	TRAP	NO ACTION	NO ACTION	10,4 or 5
		FAST MEM DATA	TRAP	NO ACTION	NO ACTION	10,6 or 7
	DATAO/B	MAIN BUS TIMEOUT	UNIBUS TIMEOUT ABORT TO 4	UNIBUS TIMEOUT ABORT TO 4	UNIBUS TIMEOUT ABORT TO 4	
		MAIN BUS PARITY	UNIBUS TIMEOUT ABORT TO 4 TRAP	UNIBUS TIMEOUT ABORT TO 4	UNIBUS TIMEOUT ABORT TO 4	10,1
		FAST MEM ADDRESS	TRAP	NO ACTION	NO ACTION	10,4 or 5

* ALL TRAPS VECTOR TO 114.

NOTES:

1. These tables only cover single errors.
2. The processor will also abort to 114 when doing DATI/P cycles on the UNIBUS and the device asserts PB. In these cases, bit 9 sets in the error register.
3. Errors on MASSBUS cycles are handled by the MASSBUS controls, and have an effect on the error register.

E The I/O Exerciser

E.1 Introduction

The I/O Exerciser (IOX) is a software tool used to diagnose hardware and software I/O problems. IOX is intended for use by system managers, system operators, and system programmers who want to test units in a hardware configuration to determine whether these units are correctly executing I/O operations. In addition, IOX enables you to measure system activity.

IOX provides a command language for executing test functions. IOX output consists of detailed error reporting and general information describing system activity.

To use IOX effectively, you should be familiar with information in the following publications:

1 *IAS System Management Guide*

2 *IAS MCR User's Guide*

3 *IAS Utilities Manual*

E.1.1 What IOX Does

The I/O Exerciser (IOX) tests the I/O units in your configuration to ensure that they correctly execute I/O functions.

IOX exercises I/O units on your system by writing a data pattern to a volume on a unit and reading the data that was written. IOX then checks to ensure that the pattern read matches the pattern written. A mismatch between the written pattern and the pattern being read indicates an error. IOX reports errors when they are encountered. IOX also displays activity reports at specifed intervals and at the completion of an I/O exercise.

IOX has a command language that you can use to direct I/O exercise functions. IOX provides commands to perform the following operations:

- Set up units in a test configuration
- Select specific units to be tested
- Set I/O exercise execution conditions
- Control I/O exercise execution
- Direct command lines to IAS for execution during an exercise
- Measure system activity using the IOX idle loop

Section E.2 summarizes the IOX commands. For complete command descriptions, including formats and usage rules, see Appendix F.

E.1.2 Setting Execution Conditions

IOX execution-time parameters enable you to set conditions under which an I/O exercise executes. You set these parameters using commands to perform the following functions:

- Specify the number of I/O errors IOX tolerates before deselecting a unit
- Set the length of time between interval summary reports
- Direct summary reports to the log file IOX.LOG
- Specify the size of the temporary file IOX uses when it exercises a unit containing a FILES-11-formatted volume
- Specify the size of the buffer IOX uses when it exercises units containing non-file-structured volumes

For more information on how to use the IOX execution-time parameters, see Sections E.3.4 and E.5.

E.1.3 Controlling The I/O Exercise

IOX commands enable you to control execution of an I/O exercise. Using execution control commands you can perform the following operations:

- Interrupt execution of an I/O exercise to enter IOX commands interactively
- Abort execution of an I/O exercise or a task
- Deselect units from the test configuration
- Display current execution status, parameter settings, and summary information
- Modify execution parameters
- Restart IOX

E.1.4 Devices Supported By IOX

IOX supports the following devices:

Device	Abbreviation
RP04/ RP05/ RP06	DB:
TU58	DD:
RA60/70/80/81/82/90	DU:
RD50/51/52/53/54	DU:
RF11	DF:
RK05/RK05F	DK:
RL01/RL02	DL:
RK06/RK07	DM:
RP02/RPR02/RP03	DP:
RM02/03/05	DR:

Device	Abbreviation
RS03/RS04	DS:
RX01	DX:
RX02	DY:
RX33/50	DU:
TU60	CT:
TU56	DT:
TU16/TE16/TU45/TU77	MM:
TU10/TE10/TS03	MT:
TU81/TK50	MU:

E.2 Summary of IOX Commands

Table E–1 contains an alphabetical list of IOX commands with a summary of their functions.

Table E–1 Summary of IOX Commands

Command	Description
ABORT	Terminates execution of an I/O exercise before normal completion
ABORT taskname	Terminates execution of a task initiated by the EXECUTE command
ABORT /ALL	Terminates execution of all tasks initiated by the EXECUTE command
BADBLOCKS	Enters the numbers of bad blocks in the IOX bad block list; displays the blocks in the bad block list
BUFFERSIZE	Sets the default size of the buffer IOX uses to write and read when exercising a unit with a non-file-structured volume
CONFIGURE	Adds a unit to your test configuration; displays the units in your test configuration
CONTROL_C	Enables or disables the interruption of an IOX exercise to enter a command interactively
DENSITY	Sets the tape density for magnetic tape testing; displays the current or default density and characteristics word
DESELECT	Terminates testing on a unit
ERRORLIMIT	Sets the number of errors tolerated before IOX deselects a unit
EXECUTE	Directs a command line to the operating system for execution
EXIT	Exits from IOX to the operating system monitor
FILES11	Selects a unit with a FILES-11 volume mounted; directs IOX to protect the volume data structure during an I/O exercise
HELP	Displays a summary description of IOX commands at your terminal
INTERLEAVE	Sets the block interleave factor for disk testing
ITERATIONS	Sets the number of times through the IOX idle loop for the idle loop iteration count
LOGFILE	Directs activity reports and error reports to either your terminal or the log file IOX.LOG
PARAMETERLIST	Displays the current settings for IOX parameters
PATTERN	Sets the test pattern IOX writes and reads during testing; display all test patterns
PRINTSUMMARY	Displays a summary report of IOX activity at your terminal

Table E–1 (Cont.) Summary of IOX Commands

Command	Description
PROCEED	Resumes IOX test execution after it has been interrupted in interactive mode
RANDOM	Enables or disables testing of random blocks rather than sequential blocks during testing
REPORTERRORS	Enables or disables error reports directed to your terminal
RESTART	Reinitializes IOX: stops testing, deconfigures all units except default units, deselects all units, and resets all parameters to their default values
RETRIES	Enables or disables retries of I/O operations by I/O drivers detecting I/O errors
RUNTIME	Sets the length of time IOX is to test units
SELECT	Selects a unit with a non-file-structured volume for an I/O exercise
SPY	Displays execution-time information for a specific unit or for all selected units
START	Begins an I/O exercise on selected units in your configuration
SUMMARYTIME	Sets the time interval between interval summary reports
TEMPORARYFILE	Sets the size of the temporary file IOX uses to exercise units with FILES-11 volumes mounted
VERIFY	Ensures that data can be read from a unit
VOLUMECHECK	Checks to determine whether a unit to be tested has a FILES-11 volume mounted
WAIT	Specifies the algorithm IOX executes in its idle loop
WRITECHECK	Enables or disables write checks to disks

E.3 Installing, Invoking, and Running IOX in an Exercise Example

This section uses an example of a terminal session to show you how to install and invoke IOX, set up your test configuration, set up execution-time parameters, select units you want to exercise, and start the I/O exercise on those units.

NOTE: All commands are given in MCR format. Use MCR or MCR mode.

E.3.1 Installing IOX

To install IOX on your system, use the MCR INSTALL command:

```
MCR>INS [11,1]IOX
```

The Task Builder command file for IOX built IOX as a nonprivileged task that runs under scheduler control in the GEN partition.

E.3.2 Invoking IOX

If IOX is installed on your system, invoke it with the following command:

```
MCR>IOX
```

IOX responds by prompting you with the task name assigned by the operating system, as follows:

```
IOX>
```

When you receive this prompt, you can enter IOX commands.

E.3.3 Setting Up Your Test Configuration

By default, all units have unit number 0 included in the IOX test configuration. You must explicitly configure any other units you want to exercise. For example:

```
IOX>CONFIGURE DK1:
IOX>CONFIGURE DK2:
IOX>CONFIGURE MM5:
```

E.3.4 Setting Up Execution-Time Parameters

You can set conditions that affect the execution of the I/O exercise. For example, you can specify the length of exercise execution time, how often you want a report of exerciser activity, and whether you want the activity reports directed to the log file SY0:IOX.LOG. You can also request IOX to determine whether the units being tested have FILES-11 volumes mounted.

Use the following commands to set up execution-time parameters for an I/O exercise:

```
IOX>RUNTIME 10
IOX>SUMMARYTIME 4
IOX>LOGFILE YES
```

These commands direct IOX to execute the I/O exercise for 10 minutes, generate an interval report every four minutes, and direct all reports to the log file.

E.3.5 Selecting Units For An I/O Exercise

Once you have set up your test configuration, you can select units to be tested. Only units you explicitly select are exercised. For example:

```
IOX>FILES11 DK1:
```

This command selects unit DK1: for exercising. When IOX executes, only DK1: is exercised. You can use either the SELECT command or the FILES11 command to select a unit for exercising. The FILES11 command is used in this example because it protects the file structure on a FILES-11 volume during an I/O exercise.

E.3.6 Starting An IOX Exercise Execution

After setting up your configuration and selecting the units you want exercised, begin the I/O exercise by entering the START command:

```
IOX>START
```

IOX then performs pattern-checking on DK1: for the length of time you specified with the RUNTIME command (10 minutes). When IOX completes execution, it generates a summary report and the interval reports, and directs them to the log file IOX.LOG. You can gain access to and examine the log file after you exit from IOX by entering the command:

```
>PIP TI:=IOX.LOG
```

This command directs the PIP utility to display the file at your terminal.

E.3.7 Exiting From IOX

You can exit from IOX and return control to the operating system by using the EXIT command as follows:

```
IOX>EXIT
```

E.4 IOX Operational Modes

IOX operates in the following three modes:

- Command
- Execution
- Interactive

Control operational modes by pressing [Ctrl/C] and [Return] on your terminal keyboard.

In most cases, [Ctrl/C] causes IAS to interrupt a task and give control to the system interface. You then receive the system prompt (for example PDS>> or MCR>) at your terminal.

However, by entering the CONTROL_C command, IOX enables you to use [Ctrl/C] in an alternate way: to interrupt an IOX I/O exercise and give control to IOX rather than to the system.

> **NOTE:** On a timesharing system, [Ctrl/C] always suspends IOX and causes PDS to prompt. If you have enabled interruptions, use [Return] instead of [Ctrl/C] to interrupt an IOX exercise and return to the IOX prompt.

This section describes the operational modes and how to use the IOX CONTROL_C command to control them.

E.4.1 Command Mode

When you invoke IOX by entering the command line described in Section E.3.2, IOX is in command mode waiting for you to enter a command. When you press carriage return in command mode, IOX prompts you as follows:

```
IOX>
```

When you enter [Ctrl/C] in command mode, IAS issues the system prompt; the normal rules for entering commands to the system apply.

In command mode, you can use the CONTROL_C command to enable or disable interruptions to an I/O exercise. CONTROL_C YES enables interruptions, enableing you to enter commands interactively; CONTROL_C NO disables interruptions to an I/O exercise.

E.4.2 Execution Mode

When you begin an I/O exercise (by means of the START command), IOX enters execution mode.

If you have enabled interruptions to the I/O exercise by means of CONTROL_C YES, IOX does not accept any input from your terminal except:

- Return, or
- Ctrl/C.

If you try to enter any other input, IOX responds with the following message:

```
IOX -- ONLY ^C ENABLED
```

If you have disabled interruptions to the I/O exercise by means of CONTROL_C NO, Ctrl/C causes IAS to issue the system prompt and execution of the I/O exercise continues on a multiuser system.

On a timesharing system, the system prompt and execution of the I/O exercise are suspended.

In this case, execution of the I/O exercise cannot be interrupted.

E.4.3 Interactive Mode

When CONTROL_C YES is set, you can interrupt an I/O exercise to enter commands interactively. When you enter:

- Return or
- Ctrl/C

IOX prompts you as follows:

```
IOX>
```

When you receive this prompt, you can enter any IOX command except the following:

BUFFERSIZE
PATTERN (PATTERN with no arguments is permitted)
RUNTIME
RESTART
START

These five commands must be entered in command mode.

Ctrl/C places IOX in interactive mode until you enter PROCEED. In interactive mode, the I/O exercise continues to execute, but error reports and activity reports directed to your terminal (if LOGFILE NO is set) are suspended until you enter PROCEED. If you set LOGFILE YES, error reports and activity reports are generated normally.

Interrupting an exercise, by <Ctrl/C> or Return, halts the IOX clock so that even though the I/O exercise is continuing it is not being timed by the IOX clock. Consequently, the exercise might run longer than you specify in the RUNTIME command.

E.5 Setting Execution-Time Parameters

Many IOX commands set IOX execution-time conditions, such as how long IOX executes, time between summary reports, and so forth. The PARAMETERLIST command displays all execution-time parameters and the value of their current setting. For example:

```
IOX>PARAMETERLIST

BUFFERSIZE= 1024.                 CONTROL_C=YES
ERRORLIMIT= 10.                   LOGFILE= NO
INTERLEAVE= 4.                    RANDOM= YES
ITERATIONS= 10.                   REPORTERRORS= YES
PATTERN= 4.                       RETRIES= NO
RUNTIME= 5.                       VOLUMECHECK= YES
SUMMARYTIME= 1.                   WAIT= YES
TEMPORARYFILE= 500.               WRITECHECK= NO

DYNAMIC MEMORY USAGE= 3278.:2248.:2248.:1.
```

The five categories of execution-time parameters are controlled by commands that perform the following operations:

- Set general execution-time conditions
- Set execution-time conditions for units with FILES-11 volumes
- Set execution-time conditions for units with non-file-structured volumes
- Set execution-time conditions for measuring system activity
- Direct the operating system to execute non-IOX tasks

The following sections describe the commands in each of these categories.

E.5.1 Setting General Execution-Time Conditions

General execution-time commands affect the execution of all I/O exercises. You are likely to want to set these commands each time you run IOX; therefore, it might be benificial to put them in an indirect file.

LOGFILE	Enables you to direct IOX activity reports to a log file, IOX.LOG
RUNTIME	Enables you to set the length of time an I/O exercise is to execute
SUMMARYTIME	Sets the time interval between interval summary reports
ERRORLIMIT	Sets the number of I/O errors IOX tolerates before it deselects a unit from the test configuration
REPORTERRORS	Enables or disables error reporting
CONTROL_C	Enables or disables interruptions to an I/O exercise
PATTERN	Sets the test pattern IOX writes and reads when exercising a unit

E.5.2 Setting Execution Conditions For Units With FILES-11 Volumes

To set execution conditions for units with FILES-11 volumes, use the FILES-11 and TEMPORARYFILE commands. The FILES11 command selects a unit to be exercised; the TEMPORARYFILE command specifies the default size of temporary file IOX uses when it exercises the unit. You can override this default with the FILES11 command.

FILES-11 volumes are volumes formatted for use by the IAS file system, FILES-11. These volumes have data structures such as home blocks, directories, and boot blocks on them. The FILES11 command selects a unit with a FILES-11 volume and enables you to exercise the unit without destroying the data structures or files on the volume.

IOX uses a temporary file when it exercises a unit with a FILES-11 volume. IOX creates the temporary file on the volume to be tested and marks the file for deletion. IOX then selects a random block in the temporary file to exercise. Because the temporary file is marked for deletion, it is erased from the disk file structure when the exercise completes or IOX terminates.

E.5.3 Setting Execution Conditions For Non-File-Structured Volumes

To exercise units with non-file-structured volumes, use the SELECT command. The SELECT command does not check for any of the FILES-11 data structures; it exercises the entire volume, destroying all data on the volume.

Use the VOLUMECHECK command (VOLUMECHECK YES) to ensure that the volume on the unit you select does not contain any FILES-11 data structures. VOLUMECHECK directs IOX to warn you if it finds such structures on what you think is a non-file-structured volume.

You can set four execution conditions when you exercise units with nonfile-structured volumes:

1 Default buffer size.

2 Pattern checking method.

3 Bad block locations.

4 Whether IOX performs certain I/O operations during the exercise.

E.5.3.1 Setting the Default Buffer Size

Use the BUFFERSIZE command to change the default size for the buffer IOX uses when it does pattern checking. When you invoke IOX, the default buffer size is 1024. You can override any default you set when you enter SELECT with another buffer size.

E.5.3.2 Choosing a Pattern-Checking Method

IOX can exercise a unit by writing to and reading from either random blocks or sequential blocks on a volume. You can control the pattern-checking method with the RANDOM and INTERLEAVE commands.

If you specify RANDOM YES, IOX exercises randomly selected blocks on the volume.

If you specify RANDOM NO, IOX exercises sequential blocks on the volume, beginning with block 0. IOX adds the interleave factor to the location of the last block exercised and uses the result of the addition to determine the next block to exercise. For example:

```
RANDOM NO
INTERLEAVE 1000.
```

These commands direct IOX to begin at block 0 and perform pattern checks on blocks 0, 1000, 2000, and 3000.

E.5.3.3 Entering Bad Block Locations

Use the BAD BLOCKS command to enter the locations of known bad blocks (that is, bad blocks located by the BAD utility) on a volume. IOX maintains a list of bad blocks and ignores them during an I/O exercise. IOX enables you to enter the locations of bad blocks manually; you must know where such blocks are on the disk to use this command effectively. See the description of the BAD BLOCKS command in Section E.10 for information on how to enter the locations of bad blocks.

E.5.3.4 Controlling I/O Operations During the I/O Exercise

Use the RETRIES and WRITECHECK commands to disable certain I/O operations during an I/O exercise. RETRIES NO directs an I/O handler to not retry an I/O operation when it receives an error. WRITECHECK NO directs an I/O handler to not perform write checks on WRITE I/O operations.

E.5.4 Setting Execution Conditions For Measuring System Activity

IOX provides two commands for measuring system activity: WAIT and ITERATIONS.

IOX executes an idle loop while it waits for units to complete I/O operations. It can execute either one of the following in this loop:

1. The wait-for-event-flag algorithm
2. The instruction *BICB -(SP),(SP)+*

The BICB instruction takes a large amount of CPU time and can be used to measure I/O throughout.

The wait-for-event-flag algorithm is the default, but you can disable it and execute the BICB instruction by entering the following command:

```
WAIT NO
```

You can determine the exact execution speed of the BICB instruction on your processor by referring to the appropriate PDP-11 Processor Handbook. Using this information, along with the number of times IOX executes the instruction, as set by the ITERATIONS command, you can estimate the level of system I/O throughput.

See the ITERATIONS command description in Appendix F.

E.5.4.1 The IOX Idle Loop Algorithm

The following code is provided for you to measure the instruction times in the IOX idle loop. This listing is the actual code in the IOX idle loop.

```
BCKGND::TST     STOPFG          ;TIME FOR INTERVAL REPORT?
        BMI     30$             ;IF MI YES
        TST     EXITFG          ;TIME TO EXIT?
```

```
        BMI     50$             ;IF MI YES
1$:     TST     ASTCFG          ;STAY IN INTERACTIVE MODE?
        BEQ     5$              ;IF EQ NO
        CLR     ASTCFG          ;YES, ASSUME WE WILL LEAVE IT
        CALL    $DEQUE          ;DEQUEUE ALL PENDING DESELECTS
        CALL    CMDAST          ;CALL INTERACTIVE COMMAND INTERPRETER
        BR      BCKGND          ;LOOP AGAIN
5$:     TST     RUNFLG          ;ARE WE RUNNING?
        BMI     40$             ;IF MI NO, ABORT
10$:    MOV     $ITCNT,CNTI     ;SETUP ITERATION COUNT
15$:    CALL    $DEQUE          ;DEQUEUE ANY PENDING DESELECTS
        TST     $WAIT           ;USE "WTSE$S" INSTRUCTION?
        BEQ     20$             ;IF EQ NO
        WTSE$S  #WAIEFN         ;WAIT FOR IOX TO DO SOMETHING
        CLEF$S  #WAIEFN         ;NOW CLEAR THE EVENT FLAG
        TST     STOPFG          ;TIME FOR INTERVAL REPORT?
        BMI     30$             ;IF MI YES
        TST     TTACTV          ;^C ACTIVE?
        BNE     1$              ;IF NE YES
        BR      25$             ;GO COUNT ITERATIONS
20$:    BICB    -(SP),(SP)+     ;WAIT
25$:    DEC     CNTI            ;ALL FINISHED?
        BGT     15$             ;IF GT NO
        ADD     #1,CNTA         ;UPDATE IDLE LOOP COUNT
        ADC     CNTB            ;RIPPLE CARRY
        ADC     CNTC
        BR      BCKGND          ;LOOP AGAIN
```

E.5.5 Directing The Operating System To Execute Non-IOX Tasks

Use the EXECUTE command to direct commands to the operating system for execution. These tasks execute concurrently with an I/O exercise. For example:

```
EXECUTE ...BAD DK1:/LI
```

This command specifies that you want the operating system to execute the BAD utility.

You can use the PARAMETERLIST command to display the status of tasks you initiate using EXECUTE. See Section E.7 for details on these status reports. You can use the ABORT command to terminate one (ABORT taskname) or all (ABORT /ALL) tasks initiated by EXECUTE.

E.6 I/O Exercise Output

IOX generates two types of output: activity reports and error reports. Activity reports are summaries of IOX processing information, such as how long an I/O exercise has run and how much time remains for the I/O exercise to complete. Error reports show such information as the block where an I/O error was encountered.

E.6.1 Activity Reports

The three types of activity reports are as follows:

- Summary report—An activity report summarizing IOX activity for the entire I/O exercise
- Interval report—An activity report summarizing IOX activity in the time intervals set by the SUMMARYTIME command

- Print summary—A report displayed at your terminal summarizing the IOX activity in the interval between the last summary report and the time you entered the PRINTSUMMARY command.

Using the LOGFILE command, you can direct IOX to display summary reports, interval reports, and error reports at your terminal or you can direct them to the log file IOX.LOG. The following lines show the general format of IOX activity reports.

```
18-SEP-78   15:44:42 ******** EXERCISER STARTED ********
  TIME -- RUN:  1 MIN.   ELAPSED:  1 MIN.  REMAINING:  0 MIN.
  DK0: -- 1,183.  REQUESTS TOTALING 1,183.  BLOCKS WITH NO ERRORS
  IDLE-LOOP COUNT:  724.  ITERATIONS
18-SEP-78   15:45:56 ******** EXERCISER TERMINATED ********
```

As shown above, the output of activity reports consists of lines telling when exerciser events occured and lines containing status information.

The exerciser event lines tell you when the exerciser began execution, terminated execution, aborted execution, started a log file, and terminated entries to the log file. The following line is an example of a line describing an exerciser event:

```
18-SEP-78   15:45:56 ******** EXERCISER STARTED ********
```

The five types of status information lines are as follows:

- General user information: date and time of report
- Timing information: time between reports, total execution time (to the nearest minute), and remaining execution time
- Unit information: number of I/O requests to the unit, number of blocks exercised, and number of errors encountered on the unit
- Idle loop information: number of iterations through the idle loop
- Error reports (see Section E.6.2)

Both summary reports and interval reports can be directed to either your terminal or the log file by means of the LOGFILE command. LOGFILE YES specifies that the summary information is directed to the log file; LOGFILE NO specifies that the summary information is displayed at your terminal. LOGFILE NO is the default.

To generate a print summary, use the PRINTSUMMARY command. This report cannot be directed to the log file. The print summary shows the amount of IOX exercise activity in the interval between the last interval report and the time you entered the PRINTSUMMARY command.

E.6.2 Error Reports

The following lines show the form of an IOX error report. At the same time IOX is reporting this information, if error logging is enabled, the error logging program is also logging the error. For information on how the error logger handles the errors, refer to the *IAS System Management Guide*.

```
DK1: -- UNRECOVERABLE ERROR AT BLOCK 358 (DECIMAL) 546 (OCTAL)
DK1: -- DATA ERROR AT BLOCK 358 (DECIMAL) 546 (OCTAL)
        GOOD DATA:  165555   BAD DATA:  167155 (OCTAL)
        WORD POSITION:  308 (DECIMAL) 464 (OCTAL)
```

The error report comprises up to three lines.

- The first line tells you the block number (in decimal and octal) where the error was encountered.
- The second line tells "good data" and "bad data."
- The third line indicates the position of the word within the block (in decimal and octal) where the error was encountered. Only the first error encountered in the block is reported.

E.7 Displaying IOX Status Information

IOX provides four commands for displaying status information:

- PARAMETERLIST—Displays parameter settings
- SPY—Displays the status of a unit being tested
- CONFIGURE—Displays all devices in your configuration
- PATTERN—Displays all 11 IOX patterns

E.7.1 Displaying Parameter Settings

The PARAMETERLIST command displays the settings of IOX exercise parameters at the time you enter the command. The following lines show the format of a PARAMETERLIST display.

```
TASK    STATUS          COMMAND LINE
----    ------          ------------
...BAD  QUEUED          DK0:/LI

DEVICE  LUN     BUFSIZE         TYPE      FILESIZE
------  ---     -------         ----      --------
  DB1:   5.       512.          F11          500.
  DB2:   6.       512.          F11          500.
  MM1:   7.      1024.          NFS

BUFFERSIZE= 1024.               CONTROL_C= YES
ERRORLIMIT= 10.                 LOGFILE= NO
INTERLEAVE= 4.                  RANDOM= YES
ITERATIONS= 10.                 REPORTERRORS= YES
PATTERN= 4.                     RETRIES= NO
RUNTIME= 5.                     VOLUMECHECK= YES
SUMMARYTIME= 1.                 WAIT= YES
TEMPORARYFILE= 500.             WRITECHECK= NO

DYNAMIC MEMORY USAGE= 3278.:2248.:2248.:1.
```

For tasks initiated by the EXECUTE command, the PARAMETERLIST command displays the following information:

- The installed task name (...BAD)
- The execution status (QUEUED, ACTIVE, SUCCESS, WARNING, ERROR, SV ERROR)
- The command line elements you entered (DK0:/LI)

ACTIVE indicates that IOX is being executed. The remaining status indications relate to the Exit with Status (EXST$) Executive directive. For more information on these status indications, refer to the *IAS System Directives Reference Manual*.

For selected units, PARAMETERLIST displays the following information:

- Device specifications (DB1:, DB2:, and MM1:)
- Logical Unit Number (LUN) (5, 6, and 7)
- Buffer size IOX uses for reads and writes (512_{10} for units with FILES-11 volumes; 1024_{10} for units with non-file-structured volumes)
- Type of volume mounted on unit (F11 for FILES-11; NFS for non-file-structured volumes; VFY for units being verified)
- File size of the temporary file IOX uses for processing units with FILES-11 volumes mounted

The line describing dynamic memory usage is in the form:

```
totalmemory:freeblock:freebytes:fragments
```

Where:

- totalmemory = Size of IOX dynamic memory in bytes.
- freeblock = Size of the largest free block of IOX dynamic memory in bytes.
- bytes = Total free bytes available in IOX dynamic memory.
- fragments = Number of memory fragments in IOX dynamic memory.

E.7.2 Displaying Unit Status

The SPY command displays the status of the unit or units you are exercising. The following line is an example of a SPY command display line:

```
DB0:  -- TYPE=F11  VBN=305.  I/O=AST QUEUED   FUNCTION=WRITE
```

This line indicates:

- The unit being exercised (DB0:)
- The current virtual block being exercised (305_{10})
- The type of volume mounted on the unit (FILES-11)
- I/O request status: AST QUEUED, IN PROGRESS, or IDLE
- The type of I/O function being performed (write)

E.7.3 Displaying The Units In Your Configuration

The CONFIGURE command displays all devices that are part of your system's configuration.

For example:

```
IOX>CONFIGURE
```

This command produces the following display:

```
DB0 DD0 DF0 DK0 DL0 DM0 DP0 DR0 DS0 DX0
DY0 CT0 DT0 MF0 MM0 MS0 MT0
```

E.7.4 Displaying The Pattern Setting

The PATTERN command displays all 11 possible patterns that IOX writes and reads during testing.

For example:

```
IOX>PATTERN
```

This command produces the following display:

```
 1 = 000000,000000
 2 = 177777,177777
 3 = 125252,052525
 4 = 165555,133333*
 5 = 163126,163126
 6 = 055555,155555
 7 = 022222,122222
 8 = 111111,111111
 9 = 007417,007417
10 = 021042,021042
11 = 104210,104210
```

The asterisk indicates the default pattern that the system builds into IOX.

E.8 Starting and Controlling IOX Execution

The START command begins execution of an I/O exercise on units selected by the FILES11 command and the SELECT command. Section E.3.6 describes how to use the START command.

The CONTROL_C command enables or disables the use of Ctrl/C (or RET on a timesharing system) to interrupt an I/O exercise (see Section E.4). CONTROL_C YES enables interruptions to an I/O exercise; CONTROL_C NO disables interruptions to an I/O exercise.

E.8.1 Disabling Interruptions To An I/O Exercise

When you set CONTROL_C NO the I/O exercise cannot be interrupted. That is, when you enter Ctrl/C, the system prompts you and the rules for entering commands to the system apply. The only way to terminate the I/O exercise in this case is to use the ABORT command:

```
>ABO IOX
```

E.8.2 Interrupting IOX To Enter Commands

When you set CONTROL_C YES, you can interrupt IOX to enter commands interactively by entering Ctrl/C (or RET on a timesharing system) on your keyboard. For example, you can execute other tasks with the EXECUTE command or abort the I/O exercise with the IOX ABORT command. During this interactive session, however, IOX suspends display of error reports and activity reports to your terminal.

PROCEED directs IOX to resume displays of error reports and activity reports at your terminal. When IOX is in interactive mode, these reports are discontinued even though the exercise continues to execute.

E.8.3 Terminating An I/O Exercise And Restarting IOX

The ABORT command terminates the execution of an I/O exercise and of a task or tasks initiated by the EXECUTE command.

ABORT with no command qualifier terminates the execution of an I/O exercise before normal completion.

ABORT with a task name terminates the execution of a task initiated by the EXECUTE command. For example:

```
ABORT ...BAD
```

This command aborts the Bad Block Locator Utility (BAD).

ABORT /ALL terminates all the tasks initiated by the EXECUTE command and removes the tasks from the list of tasks to be executed by IOX.

The RESTART command directs IOX to perform the following operations:

- Deselect and detach all units
- Remove all user-configured units
- Reset all parameters to default values
- Return IOX to its original memory size

E.9 Using Indirect Command Files with IOX

You can use indirect command files with IOX in command mode. IOX accepts only one level of indirect command file; nested indirect files are not enabled.

You can use indirect command files to set up execution parameters for exercises. For example, the file RUN10.IOX contains the following IOX commands to set execution parameters:

```
LOGFILE YES
RUNTIME 10
SUMMARYTIME 1
CONTROL_C NO
```

To execute this indirect command file, you enter the command:

```
@RUN10.IOX
```

IOX accesses the file RUN10.IOX and executes the commands, setting the execution-time parameters.

You can place comments on command lines.

E.9.1 IOX Task Builder Command File

IOX is built with the defaults that it needs to perform its required functions. The Overlay Description Language (ODL) file for IOX resides in UFD [11,16] on the system disk. If you want to rebuild IOX with different defaults, you can create a Task Builder command file like the one the system uses, which is shown below, and put it in UFD [11,16].

The system command file for building IOX puts the IOX task in system UFD [11,1]. The command file is shown as inline code with commentary. (The IAS Task Builder and the ODL are described in the *IAS Task Builder Reference Manual.*)

```
;
; BUILD I/O EXERCISER
;
[11,1]IOX/FU/CP/MM,[111,16]IOX/-SP/CR/MA/-SH=[11,16]IOXBLD/MP
STACK=96
ASG=TI:1:2:4:5:6,SY0:3
TASK=...IOX
PAR=GEN
;
; NOW DEFINE THE MAXIMUM AMOUNT OF UNITS YOU WILL EVER WANT TO
; TEST. THE MAXIMUM IS 250 AND THE DEFAULT IS 32.
;
UNITS=32
;
; INHIBIT USER OF UNUSED FCS READ/WRITE LONG CODE.
;
GBLDEF=..RWLG:0
;
; THE DEFAULT TEMPORARY FILE STRUCTURED BLOCK SIZE IS
; DEFINED HERE. THE NORMAL DEFAULT IS 500 DECIMAL (764 OCTAL)
;
GBLDEF=F11SIZ:764
;
; NOW WE WILL DEFINE THE DEFAULT 2-WORD PATTERN WHICH WILL BE
; USED TO FILL THE WRITE BUFFER. THE VALUE MUST BE BETWEEN 1 AND 13.
; THE VALUES CORRESPOND TO THE FOLLOWING PATTERNS:
;        1 -- 000000,000000
;        2 -- 177777,177777
;        3 -- 125252,052525
;        4 -- 165555.133333
;        5 -- 163126,163126
;        6 -- 055555,155555
;        7 -- 022222,122222
;       10 -- 111111,111111
;       11 -- 007417,007417
;       12 -- 021042,021042
;       13 -- 104210,104210
;
GBLDEF=PATERN:4
;
; NOW WE WILL DEFINE THE CHARACTERISTICS USED FOR MAGTAPE TESTING.
; THE DEFAULT FOR TU10 IS CORE DUMP MODE; FOR ALL OTHERS IT IS
; 1600 BPI MODE. SEE THE DEVICE HANDLERS REFERENCE MANUAL FOR FURTHER
; INFORMATION CONCERNING THIS VALUE.
;
GBLDEF=MAGSTC:004004
;
; NOW DEFINE THE DEFAULT TEST RUNTIME.
;
GBLDEF=RUNTIM:5
;
; NOW DEFINE THE DEFAULT INTERVAL BETWEEN SUMMARY REPORTS.
;
GBLDEF=SUMARY:1
;
; NOW DEFINE THE ERROR THRESHOLD. THIS IS THE NUMBER OF ERRORS
; PERMITTED BETWEEN SUMMARY REPORTS BEFORE THE UNIT WILL BE DE-SELECTED.
;
```

```
GBLDEF=ERRMAX:12
;
; NOW DEFINE THE NUMBER OF ITERATIONS THAT WILL BE MADE IN THE
; BACKGROUND LOOP BEFORE A ITERATION IS COUNTED.
;
GBLDEF=ITRCNT:12
;
; NOW DEFINE THE INTERLEAVE FACTOR THAT IS USED FOR SEQUENTIAL
; NON-FILE STRUCTURED DISK OPERATIONS.
;
GBLDEF=INTRLV:4
;
; NOW DEFINE THE DEFAULT NAME FOR THE EXERCISER. THIS NAME IS IN
; RAD50 AND IS USUALLY "IOX".
;
GBLDEF=IOXNAM:035260
;
; NOW DEFINE THE DEFAULT NON-FILE STRUCTURED BYTE COUNT. THIS IS USED
; FOR ALL NON-FILE STRUCTURED DEVICES EXCEPT DECTAPE. THE USUAL VALUE
; IS 2 BLOCKS (2000(8) BYTES).
;
GBLDEF=PABCNT:2000
;
; NOW WE WILL DEFINE THE UNSOLICITED AST CHARACTER. THIS IS THE
; CHARACTER WHICH WILL INTERRUPT THE EXERCISER WHEN THE 'CONTROL_C'
; FEATURE IS ENABLED. THE DEFAULT CHARACTER IS CONTROL/C.
; CANNOT BE CHANGED FOR IAS
;
GBLDEF=$CC:3
;
; NOW WE WILL DEFINE SOME DEFAULT YES/NO VALUES.
; A VALUE OF "0" SIGNIFIES "NO"; A VALUE OF "1" SIGNIFIES "YES".
;
GBLDEF=REPORT:1                 ;REPORT ERRORS TO CONSOLE
;
GBLDEF=RANDOM:1                 ;USE RANDOM DISK SEEKS
;
GBLDEF=CONTIN:0                 ;DO NOT RUN CONTINUOUS
;
GBLDEF=CTRLC:1                  ;USE UNSOLICITED AST'S
;
GBLDEF=WAITF:1                  ;"WAIT" INSTRUCTION IN BACKGROUND LOOP
;
GBLDEF=VOLCHK:1                 ;FILE STRUCTURED VOLUME CHECKING
;
//
```

F IOX Command Descriptions

This appendix provides complete descriptions of IOX commands and the rules for entering them on command lines. The commands are arranged in alphabetical order.

F.1 Notational Conventions

The paragraphs below describe the notational conventions used in this section to define IOX command elements and how to enter them on command lines.

Command Line Elements

IOX command lines consist of command lines and qualifiers. You can abbreviate both the command name and the qualifier using the rule described below under "Command Abbreviations."

Two types of command qualifiers are available:

- One type requires numeric settings
- The other type requires YES/NO settings.

Default Values for Command Qualifiers

Where there is a default value for a command qualifier, that value is stated explicitly in the command description.

Use of Uppercase Characters

Where uppercase characters appear in a command line format, the uppercase characters must be entered as they are shown.

Use of Lowercase Characters

Lowercase letters, words, or symbols in command line format specifications represent variables for which values should be substituted. For example:

```
dduu:
```

These letters represent the values that make up a device specification. Values can be substituted for each of these variables as appropriate.

Use of Colons (:)

Colons are used in device specifications as delimiters. In this case, the colon is required. Colons are also used to separate related parts of certain command line elements. For example:

```
BADBLOCKS DK2: 70:3
```

The colon following the device specification is the specification delimiter; the colon in 70:3 is the command element separator; it means enter a sequence of 3 bad blocks beginning at block 70.

Command Abbreviations

Where short forms of commands or qualifiers are enabled, the shortest form acceptable is represented by uppercase letters. The following example shows the abbreviations enableed for the IOX LOGFILE YES command:

```
LOgfile Yes
```

This notation means that LO, LOG, LOGF, LOGFI, LOGFIL, and LOGFILE and Y, YE, and YES are valid specifications for the LOGFILE command.

Use of Braces ([ ])

Braces are used to denote optional entries in a command line.

Use of At Sign (@)

The at sign (@) is used to invoke an indirect command file. The at sign precedes the file specification for an indirect command file:

```
@filename.filetype;version
```

Enter all numeric values are entered in decimal, with the exception of device unit numbers and file version numbers.

F.2 Summary of IOX Commands

Table F–1 lists the IOX commands and their required formats. Where uppercase characters are listed in the command format, you must type those uppercase characters as listed to invoke the command.

Table F–1 IOX Command Summary

Command	Format
ABORT	ABORT *[taskname]*
BADBLOCKS	BAdblocks ddnn:*[blocknum:number,...[number,...]...]*
BUFFERSIZE	BUffersize *bytes*
CONFIGURE	CONFigure *[dduu:]*
CONTROL_C	CONTrol_C *[Yes]* CONTrol_C *[No]*
DENSITY	DENSity *dduu:[nnnn]*
DESELECT	DESelect *dduu:*
ERRORLIMIT	Errorlimit *number*
EXECUTE	EXEcute *taskname commandstring*
FILES11	Files11 *dduu:filesize*
INTERLEAVE	INterleave *number*
ITERATIONS	ITerations *number*
LOGFILE	Logfile *[Yes]* Logfile *[No]*
PARAMETERLIST	PARameterlist
PATTERN	PATtern *[patternnumber]*

Table F–1 (Cont.) IOX Command Summary

Command	Format
PRINTSUMMARY	PRIntsummary
PROCEED	PROceed
RANDOM	RAndom *[Yes]* RAndom *[No]*
REPORTERRORS	REPorterrors *[Yes]*[1] REPorterrors *[No]*[1]
RESTART	REStart
RETRIES	RETries *[Yes]* RETries *[No]*
RUNTIME	RUntime *minutes*
SELECT	SElect *dduu:[buffsize]*
SPY	SPy *[dduu:]*
START	STart
SUMMARYTIME	SUmmarytime *[minutes]*
TEMPORARYFILE	TEmporaryfile *[filesize]*
VERIFY	VErify *dduu:[blocks]*
VOLUMECHECK	VOlumecheck *[Yes]* VOlumecheck *[No]*
WAIT	WAit *[Yes]* WAit *[No]*
WRITECHECK	WRitecheck *[Yes]* WRitecheck *[No]*

[1] If you use this command, you must specify one or the other parameters.

ABORT

The ABORT command terminates the execution of an I/O exercise and any tasks initiated by the EXECUTE command.

FORMAT

ABORT *[taskname]*
[

EXAMPLE(S)

Example 1:

```
ABORT
```

ABORT with no command qualifier terminates the execution of an I/O exercise before normal completion.

Example 2:

```
ABORT ...BAD
```

This command aborts the bad block locator utility.

Example 3:

```
ABORT /ALL
```

This command terminates the execution of all the tasks initiated by the EXECUTE command and removes the tasks from the list of tasks to be executed by IOX.

BADBLOCKS

The BADBLOCKS command enables you to enter the block numbers of bad blocks on the unit IOX is to test.

FORMAT

BAdblocks ddnn: *[blocknum:number,...[number,...]...]*

In the format above, number specifies the number of sequential bad blocks beginning at the specified block number (blocknum in the format). The colon is required when you specify a sequence of bad blocks in this form.

EXAMPLE(S)

Example 1:

```
BADBLOCKS DK2: 70:3
```

This command enters the block numbers 70, 71, and 72 in the IOX bad block list.

Example 2:

```
BADBLOCKS DK2: 3
```

You can also specify the address of a single bad block. This command enters block 3 in the IOX bad block list.

You can use both of these forms on the same command line.

Example 3:

```
BADBLOCKS DK2: 100:2,3,300:100
```

This command enters blocks 100, 101, 3, and 300 through 399 in the IOX bad block list.

Example 4:

```
BADBLOCKS DK2:
```

This example shows that when you enter only the device name, IOX displays the bad blocks in its bad block list. This command causes IOX to display the following:

```
00070:003
00100:002
00003:001
00300:100
```

The first number in the display represents the beginning of the sequence of bad blocks; the second number in the display represents total bad blocks in a sequence of bad blocks.

Bad block numbers entered with the IOX BADBLOCKS command are temporary and are used only by IOX in its testing. That is, IOX ignores the bad blocks recorded by the manufacturer or such utilities as BAD; IOX uses only the blocks you enter with the BADBLOCKS command.

BUFFERSIZE

The BUFFERSIZE command changes the size of the buffer IOX uses when exercising units with non-file-structured volumes. IOX stores the specified buffer size in byte form (rounded up to the next 4-byte boundary) and IOX requests buffers of that size dynamically, as necessary.

This command is not valid if units are currently selected.

FORMAT

BUffersize *bytes*

CONFIGURE

The CONFIGURE command enters a unit in the test configuration. (Units must be configured before they can be selected for testing by the SELECT or FILES11 commands.) The CONFIGURE command also displays the units in the test configuration.

FORMAT

CONFigure *[dduu:]*

EXAMPLE(S)

Example 1:

```
CONFIGURE DK20:
```

CONFIGURE with a qualifier adds a unit to the test configuration.

Example 2:

```
IOX>CONFIGURE
```

This command adds device DK20: to the test configuration.

Example 3:

```
DB0: DB1: DB2: DD0: DF0: DK0: DK20: DL0: DM0: DP0:
DR0: DS0: DX0: DY0: CT0: DT0: MF0:  MM0: MS0: MT0:
```

CONFIGURE with no qualifiers lists all the devices in the test configuration.

Unit 0 is configured by default.

CONTROL_C

The CONTROL_C command enables or disables the use of Ctrl/C (Return on a timesharing system) to interrupt an I/O exercise.

FORMAT

CONTrol_C *[Yes]*
[No]

EXAMPLE(S)

CONTROL_C YES enables interruptions to an I/O exercise; CONTROL_C NO disables interruptions to an I/O exercise.

When you set CONTROL_C YES you can interrupt an I/O exercise to enter IOX commands interactively. For example, you can execute other tasks by means of the EXECUTE command or abort the I/O exercise by means of the IOX ABORT command.

When you interrupt an exercise with Ctrl/C or Return, the exercise continues to execute, but error reports and activity reports are not displayed at your terminal. Further, if an exercise is interrupted the IOX clock stops at the next summary interval, even though the exercise continues to execute. Therefore, the execution times shown in activity reports might not show how long the I/O exercise actually executes.

When you set CONTROL_C NO, the I/O exercise cannot be interrupted. When you enter Ctrl/C, the system prompts you and the rules for entering commands to the system apply. The only way to terminate the I/O exercise in this case is to use the ABORT command:

```
>ABO IOX
```

NOTE: On a timesharing system, Ctrl/C always suspends IOX and causes PDS to prompt. If CONTROL_C interruptions are enabled, press Return to interrupt the exercise and return to the IOX prompt.

DENSITY

The DENSITY command sets the tape density and characteristics word for IOX to use while testing magnetic tape units.

FORMAT

DENSity *dduu: [nnnn]*

EXAMPLE(S)

Three densities are legal for MT: devices:

1 200 bpi

2 556 bpi

3 800 bpi

Two densities are legal for MM: devices:

1 800 bpi

2 1600 bpi

One density is legal for MS: devices:

1 1600 bpi

When you set the density, you must specify the device. For example:

Example 1:

```
IOX>DENS MT: 556
```

This command sets the tape density for MT: testing to 556 BPI. If you do not specify the density, [nnnn], but specify the device unit, as follows:

Example 2:

```
IOX>DENS MM:
```

IOX displays the current tape density setting and characteristics word for MM: devices. For example:

Example 3:

```
DENSITY = 800 BPI CHARACTERISTICS WORD = 04004 (OCTAL)
```

DESELECT

The DESELECT command terminates testing on the specified device. DESELECT terminates I/O operations to the specified unit, deletes the bad block list associated with the unit, deassigns the LUN for the unit, and detaches the unit from the test configuration.

The unit remains in the configuration.

FORMAT

DESelect *dduu:*

ERRORLIMIT

The ERRORLIMIT command sets the number of errors tolerated between interval reports before IOX deselects a device because of too many errors.

FORMAT

Errorlimit *number*

EXAMPLE(S)

```
SUMMARYTIME 2
ERRORLIMIT 10
```

If you request summary reports every two minutes and an error limit of 10 errors, IOX enables up to 10 errors on a unit in that 2-minute interval before deselecting the unit.

EXECUTE

The EXECUTE command directs a command line to IAS for execution. You can enter the EXECUTE command while IOX is in command mode or when IOX is in interactive mode. In both cases, output directed to your terminal from the spawned task is displayed at your terminal whether or not an exercise is executing.

All the rules for entering IAS MCR command lines and the use of the SPWN$ directive apply to the parameters you enter on the EXECUTE command line.

FORMAT

EXEcute *taskname commandstring*

EXAMPLE(S)

A practical example of the use of EXECUTE is to run the BAD utility to list the bad blocks on a volume at your terminal:

```
EXE ...BAD DK0:/LI
```

This command line invokes the BAD utility, which then lists any bad blocks on a volume at your terminal.

EXIT

The EXIT command terminates IOX processing and exits from IOX to the operating system monitor.

FILES11

The FILES11 command selects a unit with a Files-11 volume for an I/O exercise. The unit must first be entered in the test configuration with the CONFIGURE command. The volume must be mounted as a FILES-11 volume.

The FILES11 command performs the same function as the SELECT command. FILES11 is used to select units with Files-11 volumes; SELECT is used to select units with non-file-structured volumes.

FORMAT

Files11 *dduu: filesize*

EXAMPLE(S)

An example of the FILES11 command is:

```
FILES11 DK1: 1000
```

This command selects the unit DK1: for I/O exercising. The number 1000 overrides the file size of the temporary file IOX uses to process units with Files-11 volumes mounted.

IOX uses a temporary file when it processes units with Files-11 volumes. You can set the size of this file with the TEMPORARYFILE command.

HELP

The HELP command displays a summary description of IOX commands at your terminal.

INTERLEAVE

The INTERLEAVE command specifies the interleave factor (in number of blocks) for exercising units.

FORMAT

INterleave *number*

EXAMPLE(S)

When IOX writes the test pattern to disk, it can write the pattern either to random blocks or to a specified sequence of blocks. You can direct IOX to write to random blocks by entering the RANDOM command:

```
RANDOM YES
```

When you direct IOX to write to a sequence of blocks (RANDOM NO is set), use INTERLEAVE to specify the number of blocks to be skipped before the next pattern is written. For example:

```
RANDOM NO
INTERLEAVE 2
```

In this example, the interleave factor is set to two and IOX writes to and reads from every second block on the disk (block 0, block 2, block 4, and so forth) beginning at block 0. For DECtapes, the only interleave factor enabled is 4.

ITERATIONS

The ITERATIONS command specifies the number of times IOX executes its idle loop before it counts an iteration through the idle loop.

FORMAT

ITerations *number*

EXAMPLE(S)

```
ITERATIONS 10
```

IOX counts one iteration for every 10 times it executes its idle loop.

Use the ITERATIONS command with the WAIT command to measure system activity.

LOGFILE

The LOGFILE command enables you to direct activity reports and error reports to either your terminal or to a log file called SY0:IOX.LOG. LOGFILE YES directs reports to the log file; LOGFILE NO directs reports to your terminal. LOGFILE NO is the default.

Other types of output (from the PRINTSUMMARY command or SPY command, for example) are always directed to your terminal.

IOX notes the time you started the log file and the time you completed entering data to it in the activity reports. Once IOX.LOG is created, IOX appends data to the end of the file without creating new versions of the file.

Refer to Section E.6 for more information on how to use the LOGFILE command.

FORMAT

Logfile *[Yes]*
[No]

PARAMETERLIST

The PARAMETERLIST command displays the settings of the IOX execution parameters at the time you entered the command. It also displays information on currently selected devices, active tasks, and the amount of dynamic memory in use.

FORMAT

PARameterlist

EXAMPLE(S)

When you enter the PARAMETERLIST command, IOX displays information in the following form:

```
IOX>PARAMETERLIST


TASK   STATUS       COMMAND LINE
----   ------       ------------
...BAD QUEUED        DK0:/LI

DEVICE LUN   BUFSIZE     TYPE  FILESIZE
------ ---   -------     ----  --------
DB1:   5.     512.     F11    500.
DB2:   6.     512.     F11    500.
MM1:   7.    1024.     NFS

BUFFERSIZE= 1024.          CONTROL_C= YES
ERRORLIMIT= 10.          LOGFILE= NO
INTERLEAVE= 4.         RANDOM= YES
ITERATIONS= 10.          REPORTERRORS= YES
PATTERN= 4.           RETRIES= NO
RUNTIME= 5.           VOLUMECHECK= YES
SUMMARYTIME= 1.          WAIT= YES
TEMPORARYFILE= 500.        WRITECHECK= NO

DYNAMIC MEMORY USAGE= 3278.:2248.:2248.:1.
```

For tasks initiated by the EXECUTE command, the PARAMETERLIST command displays the following information:

- The installed task name (...BAD)
- The execution status (QUEUED, ACTIVE, SUCCESS, WARNING, ERROR, SV ERROR)
- The command line elements you entered (DK0:/LI)

ACTIVE indicates that IOX is being executed. The remaining status indications relate to Exit with Status (EXST$) Executive directive. For more information on these status indications, refer to the *IAS System Directives Reference Manual*.

For selected units, PARAMETERLIST displays the following information:

- Device specifications (DB1:, DB2:, and MM1:)
- Logical Unit Number (LUN) (5, 6, and 7)

- Buffer size IOX uses for reads and writes (512(10) for units with Files-11 volumes mounted; 1024(10) for units with non-file-structured volumes)
- Type of volume mounted on unit (F11 for Files-11; NFS for non-file-structured volumes; VFY for units being verified) a blank goes here
- File size of the temporary file IOX uses for processing units with Files-11 volumes mounted

The line describing dynamic memory usage is in the form:

```
totalmemory:freeblock:freebytes:fragments
```

Where:

- totalmemory = Size of IOX dynamic memory in bytes.
- freeblock = Size of the largest free block of IOX dynamic memory in bytes.
- bytes = Total free bytes available in IOX dynamic memory.
- fragments = Number of memory fragments in IOX dynamic memory.

PATTERN

The PATTERN command sets the pattern IOX writes and reads during testing.

FORMAT

PATtern *[patternnumber]*

EXAMPLE(S)

Eleven patterns exist, each specified by one of the following numbers:

```
Pattern
Number       Pattern

 1        000000,000000
 2        177777,177777
 3        125252,052525
 4        165555,133333 *
 5        163126,163126
 6        055555,155555
 7        022222,122222
 8        111111,111111
 9        007417,007417
10        021042,021042
11        104210,104210
```

For example:

```
PATTERN 3
```

This command directs IOX to use the pattern 125252,052525 when it writes to units for testing. IOX places an asterisk (*) next to the selected pattern when you display the patterns.

PATTERN 4 is the default pattern.

PRINTSUMMARY

The PRINTSUMMARY command prints the totals for I/O exercise activity between the time of the last interval summary report and the time you entered the PRINTSUMMARY command. These totals are printed in the form of a summary report displayed at your terminal. Section E.6 shows the form of a print summary report.

FORMAT

PRIntsummary

PROCEED

The PROCEED command is used to resume display of IOX activity reports and error reports at your terminal and to restart the IOX clock after these functions have been suspended by [Ctrl/C].

[Ctrl/C] suspends activity and error reporting and stops the IOX clock. PROCEED resumes reporting to your terminal and restarts the IOX clock.

FORMAT

PROceed

RANDOM

The RANDOM command directs IOX to use a random number generator to choose the blocks on the disk being exercised. RANDOM YES directs IOX to use the random number generator; RANDOM NO directs IOX to use the number of sequential blocks specified in the INTERLEAVE command.

Refer to the command description of the INTERLEAVE command for information on how to use INTERLEAVE.

FORMAT

RAndom *[Yes]*
[No]

REPORTERRORS

The REPORTERRORS command enables or disables IOX error reports. REPORTERRORS YES directs IOX to report errors; REPORTERRORS NO disables error reports. REPORTERRORS YES is the default.

When you direct IOX to report errors, IOX directs the error reports either to your terminal or to the log file IOX.LOG, depending on how you have set the LOGFILE command. (See the command description of LOGFILE for information on how to use LOGFILE.)

IOX error reports have the following form:

```
DK1: -- UNRECOVERABLE ERROR AT BLOCK 358 (DECIMAL) (OCTAL)
DK1: -- DATA ERROR AT BLOCK 358 (DECIMAL) (OCTAL)
    GOOD DATA: 165555  BAD DATA: 167155 (OCTAL)
    WORD POSITION: 308 (DECIMAL) 464 (OCTAL)
```

FORMAT

REPorterrors *[Yes]* *[No]*

You must specify either Yes or No if you use this command.

RESTART

The RESTART command directs IOX to deselect and detach all units, remove all user-configured units, reset all parameters to their default values, and return the exerciser task to its original memory size.

FORMAT

REStart

RETRIES

The RETRIES command directs an I/O driver to either retry or not retry an I/O operation after it receives an error on the first attempt at the operation. RETRIES is used when exercising units with non-file-structured volumes.

FORMAT

RETries *[Yes]*
[No]

RUNTIME

The RUNTIME command specifies the amount of time in minutes that IOX is to test a unit or units.

FORMAT

RUntime *minutes*

EXAMPLE(S)

```
RUNTIME 2
```

This command directs IOX to exercise selected devices for two minutes. If you specify RUNTIME 0, IOX tests units continuously.

The default run time is five minutes.

SELECT

The SELECT command selects a unit with a non-file-structured volume for an I/O exercise. The volume must be mounted as a foreign volume, as follows:

```
MCR>MOU ddn:/CHA=[FOR]
```

or:

```
PDS>MOU/FOR/NOOP ddn: vol-id
```

FORMAT

SElect *dduu: [buffsize]*

EXAMPLE(S)

An example of the SELECT command is:

```
SELECT DK1: 2000.
```

This command selects unit DK1: for an I/O exercise and specifies a buffer size of 2000(10) bytes for IOX processing.

The unit must be configured with the CONFIGURE command before you can select it.

The SELECT command enables an I/O exercise to destroy all the data structures on the mounted volume. To ensure that the mounted volume is not a Files-11 volume, enter:

```
VOLUMECHECK YES
```

If you do not specify the buffsize parameter, the size specified in the last BUFFERSIZE command applies.

The default buffer size is 1024 bytes.

SPY

The SPY command displays information describing the current status of a unit or units. When you specify no device on the command line, the status of all units is displayed.

SPY displays status information in the following form:

```
DBO: -- TYPE=F11 VBN=305. I/O=AST QUEUED  FUNCTION=WRITE
```

This line indicates:

- The unit being exercised (DB0:)
- The type of volume mounted on the unit (Files-11)
- The Virtual Block Number being tested 305(10). (This would be the Logical Block Number (LBN) for an unformatted volume)
- I/O request status: AST QUEUED, IN PROGRESS, or IDLE
- The type of I/O function being performed (WRITE)

FORMAT

SPy *[dduu:]*

START

The START command starts execution of an I/O exercise on units selected by the FILES11 command and the SELECT command. Refer to Section E.3 for information on how to use the START command.

FORMAT

STart

SUMMARYTIME

The SUMMARYTIME command specifies the time interval in minutes between interval summary reports.

FORMAT

SUmmarytime *[minutes]*

EXAMPLE(S)

```
SUMMARYTIME 15
```

This command specifies that you want a summary report generated every 15 minutes.

A value of 0 specifies that you want no summary reports.

The default for the interval between summary reports is 1 minute.

TEMPORARYFILE

The TEMPORARYFILE command is used when you want to exercise a unit with a Files-11 volume mounted. TEMPORARYFILE specifies the size in blocks of the temporary file IOX uses when it tests file-structured devices. This file is marked for deletion when IOX exits. Thus, if IOX terminates prematurely, your file structure is not corrupted with temporary files used by IOX.

FORMAT

TEmporaryfile *[filesize]*

VERIFY

The VERIFY command is used with units that have non-file-structured volumes mounted. VERIFY ensures that a device is performing I/O correctly by reading the number of blocks (specified by the blocks parameter) from the specified unit.

IOX performs no pattern checking with VERIFY; that is, no I/O exercise is performed.

FORMAT

VErify *dduu: [blocks]*

VOLUMECHECK

The VOLUMECHECK command enables or disables a check by IOX to determine whether or not a unit being tested as non-file-structured has a home block. If you check for the home block and IOX finds a home block on the unit, IOX issues a message warning you that the unit has a Files-11 volume mounted and you may be destroying vital information.

The default is VOLUMECHECK YES.

FORMAT

VOlumecheck *[Yes]*
[No]

WAIT

The WAIT command is used with the ITERATIONS command to measure system activity. IOX executes an idle loop while it waits for you to enter a command. It can execute either one of the following in this loop:

1 The wait-for-event-flag algorithm

2 The instruction BICB -(SP),(SP)+

The BICB instruction takes up a large amount of CPU time and can be used to measure I/O throughput.

The wait-for-event-flag algorithm is the default, but you can disable it and execute the BICB instruction by entering the command:

```
WAIT NO
```

Refer to Section E.5.4 for information on how to use the WAIT command.

FORMAT

WAit *[Yes]*
[No]

WRITECHECK

THE WRITECHECK command enables or disables write checks by I/O drivers when they perform I/O during an I/O exercise. WRITECHECK is used to exercise units with non-file-structured volumes. WRITECHECK YES directs the I/O driver to perform write checks; WRITECHECK NO directs the I/O driver to disable write checks.

When an I/O driver performs an I/O operation, it issues three instructions: WRITE, WRITE CHECK, and READ. By eliminating one of these I/O operations, you can increase the speed of the I/O exercise.

The default is WRITECHECK NO.

FORMAT

WRitecheck *[Yes]*
[No]

G IOX Error Messages

This appendix lists the IOX error messages. When IOX can determine that a specific device unit is involved, it also displays the device unit.

dduu:, BAD BLOCK, XXX (DECIMAL) XXX (OCTAL)

Explanation: The error code returned to IOX was IE.BBE (bad block error).

User Action: Use the BADBLOCKS command to enter that bad block into the IOX bad block list.

dduu:, DATA ERROR AT BLOCK XXX (DECIMAL) XXX (OCTAL)
GOOD DATA: XXXXXX BAD DATA: XXXXXX (OCTAL)
WORD POSITION: XXX (DECIMAL) XXX (OCTAL)

Explanation: IOX found an error when comparing the read data with the written data. The position of the word within the IOX data buffer is shown along with the good and bad data.

User Action: None. If the error persists, enter that block in the bad block list by using the BADBLOCKS command (for non-file-structured devices only.)

dduu:, DATA OVERRUN AT BLOCK XXX (DECIMAL) XXX (OCTAL)

Explanation: IOX received an error return of IE.DAO. The record size given was greater than that read. This error occurs on tape devices only.

User Action: None.

dduu:, END OF VOLUME SEEN

Explanation: IOX attempted to write a record past the end of tape marker.

User Action: None.

dduu:, ERROR THRESHOLD EXCEEDED

Explanation: The number of errors set with the ERRORLIMIT command has been exceeded.

User Action: None. This is an informational message only. IOX deselects the unit and continues to test the other devices.

dduu:, FAILED TO DETACH UNIT

Explanation: This error is extremely rare because IOX detaches all units that it has attached and only those units.

User Action: None. This is a possible operating system problem.

dduu:, FATAL HARDWARE ERROR

Explanation: The IE.FHE (fatal hardware error) I/O error code was returned to IOX. IOX could not recover from the error.

User Action: Check the hardware.

dduu:, I/O REQUEST ABORTED

Explanation: The IE.ABO I/O error code was returned to IOX. The QIO to the device failed.

User Action: None. If the problem persists, check the system code (Executive or driver).

dduu:, I/O UNSATISFIED AT TIMEOUT—FUNCTION: XXXX

Explanation: When IOX finishes processing, it waits five seconds for I/O to stop. If a long tape is rewinding, this message may occur. XXXX is the function that timed out: READ, WRITE, or REWIND.

User Action: None.

dduu:, NO RESPONSE

Explanation: Unit did not respond to IOX command within one minute. The device may have been accidentally put offline or a tape drive may have lost its column vacuum.

User Action: Check the device.

dduu:, NOT READY

Explanation: The unit is not ready.

User Action: Check the device.

dduu:, OFF-LINE

Explanation: The unit is offline.

User Action: Check the unit.

dduu:, PRIVILEGE VIOLATION

Explanation: The device was not allocated or was mounted by someone else.

User Action: Check ownership of the device.

dduu:, UNIT CONTAINS FILE STRUCTURED VOLUME

Explanation: The VOLUMECHECK command found that the unit is file structured because the file contains a home block. You cannot use the SELECT command on this unit.

User Action: IOX destroys information on non-file-structured volumes. Make certain that you really want this unit tested as a non-file-structured volume.

dduu:, UNIT NOT MOUNTED FOR NFS OPERATIONS

Explanation: You tried to issue a command for a non-file-structured device to a file-structured device.

User Action: Check the device configuration and reissue the command.

dduu:, UNEXPECTED ERROR XX

Explanation: IOX received an error code that it could not process. XX is the error code.

User Action: None.

dduu:, UNEXPECTED REQUEST TO QUEUE I/O—REQUEST ABORTED

Explanation: IOX receives an AST from a device after it has deselected the device.

User Action: Check the device.

dduu:, UNRECOVERABLE ERROR AT BLOCK XXX (DECIMAL) XXX (OCTAL)

Explanation: IOX received an IE.VER code from the driver. This code indicates an unrecoverable error.

User Action: Check the device and the media.

dduu:, VERIFICATION COMPLETE

Explanation: This is an information message. The operation of the VERIFY command has completed.

User Action: None.

dduu:, WRITE-LOCKED

Explanation: Device is write-locked.

User Action: Check the device write-lock/write-enable switch.

dduu:, WRITECHECK ERROR AT BLOCK XXX (DECIMAL) XXX (OCTAL)

Explanation: The write check operation failed at block XXX.

User Action: None.

IOX, "@" SYNTAX ERROR

Explanation: An error occurred in the indirect command file.

User Action: Check the indirect command file.

IOX, "@" NESTING LEVEL EXCEEDED

Explanation: Explanation: An indirect command file can have only one level of nesting.

User Action: Check that the indirect command file has only one level of nesting.

IOX, ABORTING DUE TO MCR/PDS COMMAND OR DIRECTIVE

Explanation: IOX was aborted from an external source, but did the necessary housekeeping before exiting.

User Action: Restart IOX.

IOX, BUFFER SIZE GREATER THAN DEFAULT

Explanation: The buffer size used in the SELECT command is greater than the default buffer size in the BUFFERSIZE command.

User Action: Respecify the SELECT command.

IOX, COMMAND I/O ERROR

Explanation: IOX detected an error when it tried to read the command that you were typing in at the terminal.

User Action: Retype the command.

IOX, COMMAND VALID ONLY AFTER ABORT

Explanation: You tried to enter a command that cannot be entered while IOX is running. These commands are: RUNTIME, BUFFERSIZE, PATTERN, RESTART, and START.

User Action: Enter these commands in command mode.

IOX, COMMAND NOT UNIQUE

Explanation: The command abbreviation you entered was not sufficient to distinguish it from another command.

User Action: Reenter the command with more letters.

IOX, DRIVER NOT LOADED

Explanation: IOX tried to attach a device with an unloaded driver.

User Action: Load the driver.

IOX, DUPLICATE BLOCK NUMBER

Explanation: The block number that you entered with the BADBLOCKS command is already in the bad blocks list.

User Action: Reenter the command.

IOX, ERROR ATTEMPTING TO READ HOME BLOCK

Explanation: When the VOLUMECHECK command has been given, this error occurs if IOX reads a home block or has an error trying to read a home block. The error indicates that IOX has found a file-structured volume and might destroy the information on it.

User Action: Stop IOX and check the disk that you have mounted on that device unit.

IOX, FAILED TO OPEN TEMPORARY FILE ON dduu:

Explanation: IOX could not find the necessary number of blocks on the disk as specified in the FILES11 command or the TEMPORARYFILE command.

User Action: Specify a smaller file size.

IOX, FAILED TO OPEN "@" FILE

Explanation: GCML failed to open the indirect command file.

User Action: Make sure that the file exists and reenter the indirect command file.

IOX, FAILED TO OPEN LOG FILE

Explanation: IOX could not open the log file.

User Action: The disk might be write protected or might not have enough blocks.

IOX, FAILED TO CLOSE LOG FILE

Explanation: IOX could not close the log file.

User Action: Check the device.

IOX, FAILED TO ATTACH DDUU:

Explanation: IOX could not attach unit.

User Action: The device might be mounted as a Files-11 volume or someone else may be attached the unit. Make sure that the device is mounted foreign.

IOX, FATAL FILES ERROR. F.ERR= XXX PC+2= XXX

Explanation: An error occurred while IOX was trying to issue a READ$ or a WRITE$ command. F.ERR is the error code in the FDB. PC+2 is the location of the error in IOX. IOX immediately exits.

User Action: Investigate the error code. See the *IAS I/O Operations Reference Manual*.

IOX, FATAL ERROR. DSW= XXX PC+2= XXX

Explanation: A directive issued by IOX did not work. DSW is the directive status word. PC+2 is the location of the failure in IOX. IOX immediately exits.

User Action: Investigate the source of the error in the code.

IOX, FILES OPERATIONS NOT SUPPORTED ON dduu:

Explanation: You entered a FILES11 command to a volume that is not a Files-11 volume.

User Action: Reenter the command for the correct device or use a different command.

IOX, ILLEGAL DENSITY

Explanation: The density that you specified for the device is incorrect. The device does not support that density.

User Action: Reenter the command using a different density or device.

IOX, ILLEGAL UNIT SPECIFICATION

Explanation: The syntax of the unit specification is incorrect.

User Action: Reenter the command.

IOX, INVALID BLOCK NUMBER

Explanation: The block number that you entered for the BADBLOCKS command was not a valid block number for that device.

User Action: Reenter the command using a different block number.

IOX, INVALID TASK NAME

Explanation: The task name that you entered is invalid. That is, the task name is incorrect, the task does not exist, or the task is not installed.

User Action: Reenter the task name or determine whether the task exists or is installed.

IOX, INVALID PATTERN NUMBER

Explanation: You tried to set a data pattern for IOX to use for I/O testing that IOX did not recognize.

User Action: Check the PATTERN command and reenter the correct data pattern.

IOX, INVALID COMMAND—TYPE H FOR HELP

Explanation: You entered a command that IOX did not recognize.

User Action: Enter H (for help) and IOX displays all of its commands.

IOX, NO BUFFER SPACE

Explanation: IOX could not extend to make room for a needed larger buffer.

User Action: Reinstall IOX with a larger increment.

IOX, NO LUNS AVAILABLE

Explanation: This is an unlikely message because IOX enables 250 devices. Unless you have more than 250 devices, this message may indicate a system problem.

User Action: If you have less than 250 devices, call your software specialist.

IOX, NO ACTIVITY REMAINS—ABORTING

Explanation: All disk testing and all tasks are finished before the run time has expired.

User Action: None.

IOX, NO ACTIVITY TO START

Explanation: You issued the START command without defining the configuration or units to exercise.

User Action: Configure and define the devices that you want to exercise.

IOX, ONLY ^C ALLOWED

Explanation: You previously entered the CONTROL_C YES command to enable interruptions to IOX. Therefore, IOX does not accept any other input but [Ctrl/C] (or <Return> on a timesharing system).

User Action: Enter [Ctrl/C] or [Return], then enter the command.

IOX, SYNTAX ERROR

Explanation: The command line that you entered contained a syntax error.

User Action: Reenter the command line.

IOX, TASK NEVER EXECUTED

Explanation: The task that you tried to abort is not on the list of tasks initiated by the EXECUTE command.

User Action: Check your list of tasks to be initiated by the EXECUTE command.

IOX, TASK "XXXXXX" CURRENTLY ACTIVE

Explanation: You tried to execute a task that was already active. XXXXXX is the task name.

User Action: None.

IOX, TASK "XXXXXX" EXECUTION COMPLETE. STATUS= NNNNNN

Explanation: The task completed execution. XXXXXX is the task name. NNNNNN is the task status.

User Action: None.

IOX, TASK "XXXXXX" NOT INSTALLED

Explanation: The task name "XXXXXX" is not installed and cannot be executed as requested.

User Action: Install the task.

IOX, TASK "XXXXXX" UNABLE TO EXECUTE

Explanation: IOX could not execute the task within a specified time.

User Action: Try to reexecute the task.

IOX, UNABLE TO ASSIGN LUN TO dduu:

Explanation: You tried to access a unit that is not on the system.

User Action: Check the system configuration.

IOX, UNIT NOT A MAGTAPE

Explanation: You tried to issue a DENSITY command to a device that is not a magnetic tape device.

User Action: Reissue the command.

IOX, UNIT NOT SELECTED

Explanation: You issued a device-dependent command to a device that you have not selected with the SELECT command or the FILES11 command.

User Action: Determine which device you want to affect and select it with the SELECT or FILES-11 command.

IOX, UNIT IS FILE STRUCTURED

Explanation: You used the BADBLOCKS command to try to create a bad blocks list on a file-structured volume. The command did not execute.

User Action: Make certain that you really want IOX to write on this device.

IOX, UNIT ALREADY CONFIGURED

Explanation: You tried to configure a unit with the CONFIGURE command that is already in the configuration.

User Action: Check the IOX device configuration.

IOX, UNIT NOT CONFIGURED

Explanation: You tried to access a unit that is not in the IOX unit configuration.

User Action: Configure the unit using the CONFIGURE command.

IOX, UNIT ALREADY SELECTED

Explanation: You tried to select a unit that has already been selected.

User Action: Check the printout to find which units you have previously selected.

IOX, UNITS ARE SELECTED—COMMAND IGNORED

Explanation: The BUFFERSIZE command is valid only if no units are selected by either the FILES11 or SELECT command.

User Action: Deselect all selected units.

Index

A

B

C

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

X

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