

# IAS

## System Generation Guide

Order No. DEC-11-OISGA-A-D

IAS Version 1

Order additional copies as directed on the Software  
Information page at the back of this document.

digital equipment corporation • maynard, massachusetts

First Printing, December 1975

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

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

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

Copyright © 1975 by Digital Equipment Corporation

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

The following are trademarks of Digital Equipment Corporation:

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

## CONTENTS

		Page	
CHAPTER	1	INTRODUCTION TO SYSTEM GENERATION	1-1
	1.1	OVERVIEW OF SYSTEM GENERATION	1-1
	1.2	THE DISTRIBUTION SYSTEM	1-1
	1.3	STAGE 1	1-2
	1.3.1	Phase 1	1-2
	1.3.1.1	Terminal Handler Parameters	1-2
	1.3.1.2	System Generation Directives	1-3
	1.3.1.3	The Results of Phase 1	1-3
	1.3.2	Phase 2	1-3
	1.4	STAGE 2: GENERATING THE TIMESHARING EXECUTIVE	1-4
CHAPTER	2	PREPARING FOR STAGE 1 SYSTEM GENERATION	2-1
	2.1	EDITING SYSTEM GENERATION FILES	2-1
	2.2	POOL REQUIREMENTS OF SGN1	2-1
	2.3	PHASE 2 OF SYSTEM GENERATION	2-3
	2.4	IAS BOOTSTRAPPING	2-3
	2.4.1	System Generation Phase 1 (SGN1)	2-4
	2.4.2	System Generation Phase 2 (SGN2)	2-5
	2.4.3	Saving a Generated System	2-6
	2.4.4	Subsequent Rebooting of a Saved Image	2-7
CHAPTER	3	SYSTEM DEFINITIONS	3-1
	3.1	TERMINAL HANDLER PARAMETERS	3-1
	3.2	SYSTEM GENERATION DIRECTIVES	3-3
	3.2.1	Target Directive	3-4
	3.2.1.1	Console Entry	3-4
	3.2.1.2	Description	3-4
	3.2.2.	PDP-11 Directive	3-5
	3.2.2.1	Console Entry	3-6
	3.2.3	Memory Allocation	3-7
	3.2.3.1	EXEC Directive	3-8
	3.2.3.2	SCOM Directive	3-9
	3.2.3.3	POOL Directive	3-11
	3.2.3.4	PAR Directive	3-12
	3.2.3.5	DEV Directive	3-13

# CONTENTS

			Page
	3.2.4	System Default Specifications	3-16
	3.2.4.1	SY Directive	3-17
	3.2.4.2	DPAR Directive	3-18
	3.2.4.3	CKPNT Directive	3-19
	3.2.5	INS Directive	3-20
	3.2.5.1	Console Entry	3-20
	3.2.5.2	Description	3-20
	3.2.5.3	Required Task Installations	3-21
	3.2.6	Phase 2 Directive (*DELAY)	3-22
CHAPTER	4	TIMESHARING EXECUTIVE PARAMETERS	4-1
	4.1	SET-UP PARAMETERS	4-2
	4.2	SCHEDULING PARAMETERS	4-3
	4.3	SWAPPING PARAMETERS	4-4
CHAPTER	5	TRANSFERRING THE DISTRIBUTED SYSTEM	5-1
	5.1	HARDWARE BOOTSTRAP	5-1
	5.1.1	MR11DB Bootstrap	5-1
	5.1.2	BM792YB Bootstrap	5-2
	5.1.3	BM873YA Bootstrap	5-3
	5.1.4	BM873YB Bootstrap	5-4
	5.2	TRANSFERRING THE DISTRIBUTION MEDIUM FROM MAGNETIC TAPE	5-6
	5.2.1	Bootstrapping the Distribution Tape	5-6
CHAPTER	6	STAGE 1 PROCEDURES	6-1
	6.1	PHASE 1 OF SYSTEM GENERATION	6-2
	6.2	PHASE 2 OF SYSTEM GENERATION	6-9
CHAPTER	7	PERFORMING STAGE 2	7-1
APPENDIX	A	SYSTEM GENERATION FILES	A-1
	A.1	STAGE 1 FILES	A-1
	A.2	STAGE 2 FILE	A-4
APPENDIX	B	DEVICE CHARACTERISTICS WORDS	B-1
APPENDIX	C	SYSTEM GENERATION FOR TERMINALS ON DH11 OR DJ11 LINES	
	C.1	PUD CHARACTERISTICS WORDS	C-2
	C.2	PARTITION	C-5

## CONTENTS

			Page
	C.3	LOGICAL NAMES AND NUMBERS	C-5
	C.3.1	EIA and 20 ma Lines (DH11 Only)	C-5
	C.3.2	How to Assign Names	C-5
APPENDIX	D	MCR COMMANDS	D-1
	D.1	BOOTSTRAP COMMAND (BOO)	D-2
	D.2	DISMOUNT VOLUME COMMAND (DMO)	D-6
	D.3	FIX-IN-MEMORY COMMAND (FIX)	D-8
	D.4	INITIALIZE VOLUME COMMAND (INI)	D-9
	D.4.1	MAN Option for /BAD keyword	D-13
	D.5	INSTALL TASK COMMAND (INS)	D-16
	D.6	LIST COMMON BLOCKS COMMAND (COM)	D-19
	D.7	LOAD COMMAND (LOA)	D-20
	D.8	MOUNT COMMAND (MOU)	D-21
	D.8.1	Format for Mounting Multivolume Magnetic Tape	D-24
	D.9	PARTITIONS COMMAND (PAR)	D-26
	D.10	REDIRECT COMMAND (RED)	D-27
	D.11	SAVE COMMAND (SAV)	D-28
	D.12	SET COMMAND (SET)	D-30
	D.13	TIME COMMAND (TIM)	D-32
	D.14	UNFIX COMMAND (UNF)	D-33
	D.15	UNLOAD COMMAND (UNL)	D-34
	D.16	USER FILE DIRECTORY COMMAND (UFD)	D-35
APPENDIX	E	ERROR MESSAGES	E-1

## FIGURES

Figure	2-1	SGN1 Pool Usage	2-2
Figure	5-1	Output When Distribution Medium is Booted	5-8
Figure	D-1	Sample COMMON BLOCK Listing and Description of Listed Items	D-19

# CONTENTS

Page

## TABLES

Table	3-1	Terminal Handler Parameters	3-2
Table	3-2	Device Information	3-14
Table	5-1	Device Addresses for the MR11DB Bootstrap	5-2
Table	5-2	Device Addresses for the BM792YB Bootstrap	5-2
Table	5-3	Device Addresses for the BM873YA Bootstrap	5-4
Table	5-4	Device Addresses for the BM873YB Bootstrap	5-5
Table	C-1	Characteristics Word 2 for Terminal Handler	C-2
Table	C-2	Characteristics Word 3 for Terminal Handler	C-3
Table	C-3	Receiver and Transmitter Speeds	C-4
Table	C-4	Characteristics Words for Devices on DH11 Lines	C-4
Table	C-5	Characteristics Words for Devices on DJ Lines	C-5
Table	D-1	Defaults in BOO File Specifications	D-4
Table	D-2	Defaults in INS File Specifications	D-18
Table	D-3	UFD Switches	D-36

## PREFACE

### 0.1 MANUAL OBJECTIVES AND READER ASSUMPTIONS

The IAS System Generation Guide provides the user with information needed to perform an IAS System Generation. It assumes a knowledge of system information provided in the IAS System Release Notes and an understanding of the material covered in the IAS System Management Guide.

### 0.2 DOCUMENT STRUCTURE

The Manual consists of the following chapters and appendixes:

Chapter	1	Introduction to System Generation
Chapter	2	Preparing for Stage 1
Chapter	3	System Directives
Chapter	4	Timesharing Executive Parameters
Chapter	5	Transferring the Distributed System
Chapter	6	Stage 1 Procedures
Chapter	7	Stage 2 Procedures
Appendix	A	System Generation Files
Appendix	B	Device Characteristics Words
Appendix	C	System Generation for Terminals on DH11 or DJ11 Lines
Appendix	D	MCR Commands
Appendix	E	Error Messages

### 0.3 ASSOCIATED DOCUMENTS

Refer to the IAS Documentation Directory, Order Number DEC-11-OIDDA-A-D, for a description of all the associated IAS documents and their readerships.



## CHAPTER 1

### INTRODUCTION TO SYSTEM GENERATION

#### 1.1 OVERVIEW OF SYSTEM GENERATION

System generation is the process of building a system tailored both to an installation's hardware configuration and its software requirements. The IAS system distributed to the user on magnetic tape is an operational single user system which is then modified according to the installation's needs.

IAS contains a system generation program that accepts user-provided definitions of the system to be built. The program executes in two stages to create an IAS system based on the definitions provided. The first stage defines the hardware configuration, and memory allocation. The second stage defines the Timesharing Executive, which controls all batch and interactive processing.

System generation is normally performed for one of the following reasons:

1. To generate IAS for the first time
2. To revise IAS when the hardware configuration has changed
3. To revise IAS in order to improve performance

This manual provides sufficient information to generate a system for any of the above reasons. Chapter 1 describes system generation in general. Chapter 2 discusses in more detail both phases of System generation and system bootstrapping. Chapters 3 and 4 describe in detail the directives and parameters that the user must specify. Directions for generating the IAS system distributed on magnetic tape are contained in Chapter 5. Chapters 6 and 7 contain step-by-step instructions for modifying the distributed or an existing system.

#### 1.2 THE DISTRIBUTION SYSTEM

Digital distributes IAS on magnetic tape. Users must then transfer the distributed IAS system from the tape onto the system disk. Chapter 5 provides step-by-step instructions on how to effect the transfer.

Once IAS has been transferred to the system disk, the installation has a workable single-user system. This system is then used to generate IAS according to local requirements. The user specifies the local requirements by editing several files that are read by the system generation programs. The following two sections describe the files to be edited.

### 1.3 STAGE 1

Stage 1 of system generation defines the installation's hardware configuration and memory allocation and then installs all the tasks needed to run a complete IAS system. The target device is a disk (possibly the current system disk) which contains the necessary files (see Chapter 3, section 3.2.3.5). This procedure consists of two phases:

1. Phase 1 defines the hardware and the allocation of memory
2. Phase 2 installs the necessary system tasks.

#### 1.3.1 Phase 1

There are two files relating to Phase 1 of the Stage 1 system generation that must be considered:

- A MACRO source file containing terminal handler parameters
- An indirect command file containing system generation directives

An indirect command file is a sequential file containing a list of commands. To execute the sequence, the user issues an "at" sign (@) followed by the file name instead of the first command in the sequence. The system then retrieves the indirect file and executes the commands contained therein. See the IAS User's Guide for a more detailed explanation of indirect files.

1.3.1.1 Terminal Handler Parameters - The MACRO source file contains information required to build the terminal handler. It specifies the number of terminals and the hardware interfaces used to attach the terminals to the system. The user must edit the parameters file to reflect the local hardware configuration.

1.3.1.2 System Generation Directives - The system generation directives contained in the indirect command file fall into six categories:

1. The TARGET directive which defines the target system device
2. The PDP-11 directive which defines the computer to be used
3. The EXEC, SCOM and PAR directives which partition memory
4. The DEV directives which define the system peripherals and pseudo devices
5. The DPAR, CKPNT and SY directives which supply system default information
6. The INS directives which install tasks needed for phase 2.

At Stage 1 of system generation the user must supply these directives to define the installation's hardware configuration and software requirements. The real-time and timesharing activities to be run on the computer determine how the system manager decides to allocate memory. See the IAS System Management Guide for a discussion of memory allocation.

The user either edits the existing command file containing the directives or creates a new file. The directives are then issued by a call to the indirect command file. The user also has the option of typing the directives individually at a terminal. See Chapter 3 for a detailed description of each directive.

1.3.1.3 The Results of Phase 1 - When the two files described above accurately reflect the installation's requirements, the user can perform phase 1. Using the terminal handler parameters and system generation directives, phase 1 produces a software system capable of driving all the installation's peripherals and of controlling its memory. See Chapter 6 for detailed instructions on performing phase 1.

### 1.3.2 Phase 2

Phase 2, a task called SGN2, is installed automatically during phase 1. When the system established by phase 1 is activated (see Chapter 6, section 6.2), phase 2 begins to run and performs the following actions:

1. It loads the terminal handler
2. It issues a MOUNT command for the system device (SY)
3. It opens the file SY:[11,17]SYSBLD.COMD and, if the open is successful, begins to process the file and print it at the console.

The file SY:[11,17]SYSBLD.COMD installs all the tasks needed to complete the IAS system, including the current version of the Timesharing Executive. Appendix A lists the contents of the distributed version of SY:[11,17]SYSBLD.COMD. The user should make any necessary changes to the file before initiating phase 2.

The phase 2 directive \*DELAY is inserted to cause a 1-second delay in the processing of commands contained in SY:[11,17]SYSBLD.COMD. Some command sequences may fail if a delay is not specified. See Chapter 3, Section 3.2.6.

#### 1.4 STAGE 2: GENERATING THE TIMESHARING EXECUTIVE

Phase 2 of Stage 1 system generation installs the current version of the Timesharing Executive. To perform Stage 2, the revision of the Timesharing Executive, the user must first edit the file [311,107]SGDATA.MAC. This file contains the IAS interactive and batch parameters used to create two libraries which the system requires to control timesharing activities. These parameters are described in detail in Chapter 4.

When each parameter has been set to the required value the user then assembles and builds two libraries. The libraries IASCOM and IASBUF are Shareable Global Areas (SGAs) that contain data structures and common routines used by Timesharing Executive tasks. The libraries are built according to the parameters specified in [311,107]SGDATA.MAC.

Once the libraries have been built, the user must relink all the tasks that use them. The next step is to remove the old libraries and tasks in order to install the new ones. The new versions of IASCOM and IASBUF are installed first, followed by the tasks.

The user completes Stage 2 by dismounting the target disk and saving the system. A new Timesharing Executive has then been generated.

A step-by-step description of performing Stage 2 is contained in Chapter 7. The procedure is greatly simplified by the use of indirect command files for assembling, linking, removing and installing tasks and libraries.

## CHAPTER 2

### PREPARING FOR STAGE 1 SYSTEM GENERATION

This chapter contains information to help the user prepare for both phases of Stage 1 system generation, including a detailed description of IAS bootstrapping procedures.

#### 2.1 EDITING SYSTEM GENERATION FILES

Several indirect command files for system generation are provided in IAS. These indirect files can be used to define the system in phase 1 instead of typing directives in response to phase 1 requests. (Phase 1 system directives are described in Chapter 3.) SYSBLD.CMD is always used to perform phase 2. If the user wishes to tailor the system to a particular installation, he can either edit the existing files or create new ones. See the IAS Editing Utilities Reference Manual for a description of the editor.

All commands files are stored under UFD [11,17] on the system volume. Type the following commands to request the editor for the modification of existing generation files or the creation of new ones.

```
MCR>SET /UIC=[1,1]
MCR>EDI [11,17]command file
```

#### 2.2 POOL REQUIREMENTS OF SGN1

Phase 1 of system generation (SGN1) uses the dynamic pool for storage of input data and for communication with the version of install (...INV) that it requests. SGN1 returns nodes to the pool as soon as it has finished with them. If a fatal error occurs, it returns all nodes using its own internal accounting system.

As a result, pool usage is quite dynamic. The important consideration, however, is the maximum usage at any time. To compute the maximum pool usage consider the phase 1 command input while referring to Figure 2-1.

# PREPARING FOR STAGE 1 SYSTEM GENERATION

## NODE USAGE

1 node = 16 contiguous words.

This graph does not include MCR buffers of nodes used by the file system.

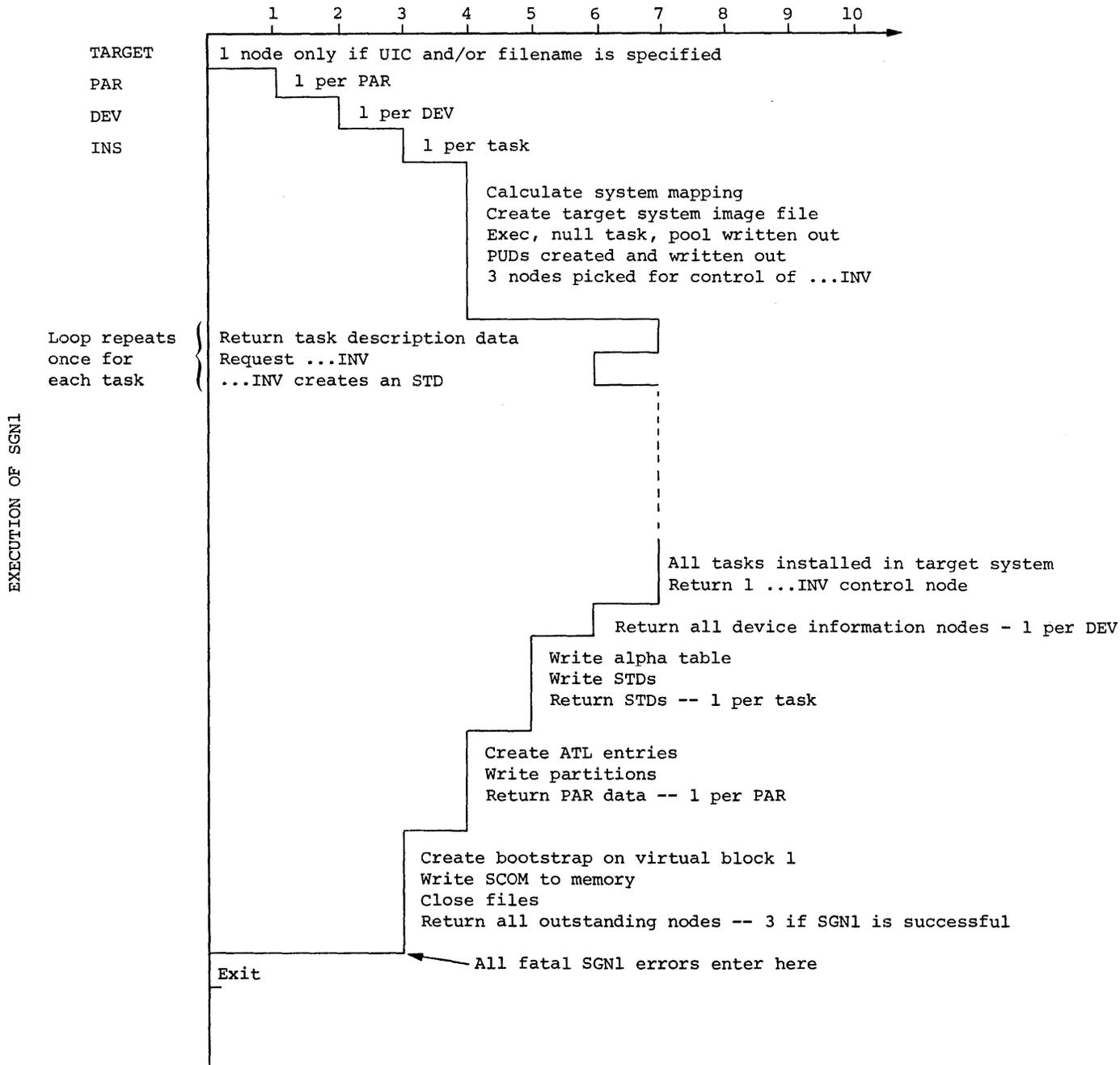


Figure 2-1  
SGN1 Pool Usage

### 2.3 PHASE 2 OF SYSTEM GENERATION

Phase 2 of system generation is installed during phase 1. On bootstrap from a phase 1 target disk as described in Chapter 5, section 5.1, phase 2 is activated and proceeds as follows:

1. Loads the terminal handler,
2. Issues a MOUNT command for the system device (SY),
3. Opens the file SY0:[11,17]SYSBLD.COMD, and if the open is successful, begins to process the file and print it on the console.

The file SYSBLD.COMD is provided in the IAS system. If the user wishes to modify it, he should do so before bootstrapping phase 2.

SYSBLD.COMD can contain the phase 2 directive, \*DELAY, described in Chapter 3, section 3.2.6, and any MCR request except SAVE or BOOT.

A task must be installed before it can be used. The task can be installed either during phase 2 of system generation or during phase 1 by means of the INS directive. If the task refers to a shared region, the referenced region must be installed before the task is installed.

(Note that shareable global areas or tasks which refer to shareable global areas can not be installed during phase 1. They must be installed during phase 2.)

When phase 2 is complete, it prints the following message on the console.

END OF SYSTEM GENERATION PHASE 2

At this point, perform the steps in Chapter 6, Section 6.2 starting with step 4. This process properly saves the system for continued use.

### 2.4 IAS BOOTSTRAPPING

The PDP-11 family of computers is supplied with a hardware bootstrapping program in read only memory. The function of this program is to read one block (physical block 0) of a specified device into main memory starting at real location zero. On successful completion of the read, the processor jumps to real location zero with memory management disabled and starts to execute the contents.

Under IAS, block zero of the booted device must contain a program to read in the whole operating system and user-defined partitions, enable memory management, and start the Executive.

During the generation and subsequent management of an IAS system, the operations related to the bootstrap go through several phases. Although these operations are relatively transparent to the user, the system manager should be aware of the actions taken and their implications. The operations can be divided into the following four categories:

1. Phase 1 of system generation
2. Phase 2 of system generation
3. Saving the generated system
4. Subsequent bootstrapping and saving of a stable system

#### 2.4.1 System Generation Phase 1 (SGN1)

The purpose of the first phase of system generation is to create a file, IAS.SAV, for example, on a target disk. This file is a bootable IAS system image configured for the target hardware; that is, the contents of IAS.SAV on completion of phase 1 are independent of the hardware configuration on which it was created. The only bootstrap-related function performed by SGN1 is to include the bootstrap at the beginning of the IAS image file.

Due to differences among mass storage device controllers, the bootstraps for various devices differ in their control logic. Therefore, IAS provides a bootstrap program for each different device supported as a system device. The bootstrap programs are named using the convention xxxxBOOT.TSK where xxxx is the commonly-used name for the device. At present, bootstraps are supplied for RK05, RP03, and RP04 disk devices.

Using the SY directive and the related device specification, SGN1 determines which bootstrap task is to be written into the beginning of the memory image file that it is creating.

Before it writes this task into the image file, however, SGN1 must insert the following information into the program.

1. The amount of memory to be filled.
2. The disk address of the IAS.SAV file.
3. The real base address of the Executive.
4. The external page address of the disk controller as specified by the corresponding DEV directive.
5. The kernel virtual address of the power recovery routine within the Executive. This routine is used to start the Executive as well as to recover from a power failure.

The first four items are available to SGN1. The last (power recovery routine address) is obtained by building the appropriate bootstrap task with the Executive symbol table, EXEC.STB. SGN1, therefore, inserts items one through five above into the bootstrap task before writing it to the IAS.SAV file.

To insert the required information, SGN1 must have access to the symbol table of the bootstrap task. Consequently, SGN1 also reads xxxxBOOT.STB to determine the bootstrap program's data locations. Since the bootstrap is linked with the EXEC.STB, SGN1 is able to locate all other symbols for the target Executive from the bootstrap task symbol file.

During phase 1 of system generation, the user has the option of defining base addresses for the Executive, the system communication area, and partitions. If base addresses are not specified, SGN1 performs a memory allocation and places the Executive on the next 32 word boundary after the bootstrap program. SGN1 determines the length of each bootstrap from the corresponding .STB file.

If the user specifies the base address, he must take care to leave sufficient space for the bootstrap program. The user can determine the bootstrap size by obtaining a task builder listing of the appropriate xxxxBOOT.STB. The following command is used.

```
MCR>TKB ,LP0:=[11,17]xxxxBOOT.STB
```

The difference between the global symbols .BO.BB (begin bootstrap) and .BO.ND (end bootstrap) rounded up to the next 32 word (100 octal) boundary provides the lowest real address at which the Executive can be placed.

It is important to note that IAS does not write block 0 of the target system device; rather, it places the bootstrap task at the beginning of the memory image file created by phase 1.

#### 2.4.2 System Generation Phase 2 (SGN2)

Some of the functions necessary for a complete system generation must be performed on the target hardware configuration (creation of the checkpoint file, for example). Others are more conveniently performed under the target software configuration (installation of a large number of tasks, for example). Therefore, phase 1 of system generation creates a runnable IAS system with SGN2 installed and loaded into its partition. When the Executive starts, it activates SGN2.

SGN2 makes no modifications to the bootstrap program either in memory or on disk. The only bootstrap-related aspect of SGN2 is the initial bootstrapping of the output file from SGN1. This bootstrapping causes SGN2 to become active.

There are two methods of booting the output of SGN1:

1. By using the BOOT function,
2. By using a combination of the BOOT function and the hardware bootstrap (ROM).

Whichever method is used, the output of SGN1 must always be booted from the device that was specified as SY during phase 1.

When the first method is used, all devices except the bootstrap device must be dismantled to ensure that all activity within the system is stopped. Then the BOOT function is requested and given the file specification of the image file to be booted. The BOOT task reads the first block of the image file into memory starting at real zero and begins to execute it. This process simulates the ROM bootstrap. BOOT does not require block zero of the device to have any special content. It uses the first virtual block of the image file specified.

With the second method, BOOT is used to create a bootstrap block 0 on the device. Then the device can be booted with the appropriate hardware ROM. BOOT accepts the file qualifier /WB following the file specification of the IAS image file to be booted. If this qualifier is present, the first block of the file is copied to block 0 of the same volume. This procedure does not perform a system reload, but does make the volume hardware bootable.

The inclusion of the BOOT function allows the complete generation and testing of a new IAS system without changing the target device substantially. The only modification to the target device is the creation of a new memory image file and installation of tasks into that system. Note that the disk files containing handlers and other system tasks are modified when they are installed into the new system. Therefore, even though it is possible to reboot a disk on which a new IAS image file has been created, care must be taken with such tasks as INS, ISMOU, and ISFCP, because their headers will reflect the new system. This problem can be circumvented by making separate copies of such tasks before running SGN1.

For further information about the BOOT command, refer to Appendix D.

### 2.4.3 Saving a Generated System

When SGN2 runs successfully, all the facilities of the generated IAS system are available. Once the user is satisfied that the system does perform as intended, it is necessary to dismount all devices and save the updated memory-resident system on the bootstrap device. If extensive testing of the new system is intended, it is advisable to save the system first.

The purpose of the SAVE function is to rewrite the file that was originally booted. SAVE uses the bootstrap program that is permanently resident in low memory to perform this function. This approach insures the following:

1. Only as much memory as specified during SGN1 is written,
2. Memory is saved at the disk address (i.e., in the image file) from which it is to be booted.
3. The device on which the file is saved is independent of any redirection of SY or other devices. That is, the save takes place on the device from which it was booted.

Because the output of SAVE is an exact replica of memory, the IAS image file appears as a system with the SAVE task running. In order for SAVE to exit, it makes several modifications to the bootstrap in

low memory after copying it into its own buffer to perform the disk write and before performing the actual save. These modifications are as follows.

1. Store the content of user memory management register APR0.
2. Store the re-entry address in SAVE which is relative to user APR0.
3. Modify an internal branch within the bootstrap so that a sequence of instructions to return to SAVE in user mode is executed after memory management is enabled.

At this point, two versions of the bootstrap program exist in memory: the original within SAVE and a modified version beginning at real location 0.

SAVE now changes the I/O function code of its version from a read to a write, inhibits task switching and interrupts by raising the processor priority to 7, and executes its version of the bootstrap. Upon completion of the save to disk, SAVE executes the same code that it executes when rebooting the saved memory image, as described in the next section.

#### 2.4.4 Subsequent Rebooting of a Saved Image

Regardless of which method is used to boot a saved IAS system, the bootstrap program that performs the read originally came from virtual block 1 of the file being booted. This bootstrap overlays itself with the beginning of the file it is reading. Since the only differences between bootstraps is in data, the instructions continue to execute.

On completion of the read of the saved memory image, which is performed in 1K sections, the overlay copy of the bootstrap executes a sequence of instructions that returns to SAVE in user mode with task switching and interrupts inhibited. SAVE then restores all volatile registers, restores the power recovery vector which was used to direct the bootstrap to SAVE, and performs a number of other functions not related to the bootstrap. Upon completion, task switching and interrupts are enabled and the Executive is started by simulating a power fail AST. Finally, SAVE prints the IAS sign-on message and exits.

One of the other functions performed by SAVE is to check the existence of the system clock. Since IAS can use either a KW11-L or a KW11-P clock, SAVE first checks for the clock for which the system was generated. If this test fails, SAVE attempts to start the other type of clock. Therefore, it is possible to boot an IAS system that was generated and saved on a configuration with a different type of clock.



## CHAPTER 3

### SYSTEM DEFINITIONS

This chapter describes the terminal handler parameters and system generation directives that the user must specify for Stage 1 of system generation. The IAS Release Notes contain system information needed to specify these parameters and directives. See the IAS Editing Utilities Reference Manual for information about the IAS text editors needed to modify system generation files.

#### 3.1 TERMINAL HANDLER PARAMETERS

The terminal handler parameters specify the number of terminals attached to the system, the hardware interfaces used to attach them and other required special servicing such as dial-up. The parameters are contained in a parameter file called [311,14]ISTT64PAR.MAC, which the user must edit before starting Stage 1.

#### NOTE

Decimal numbers in MACRO-11 source files are specified with a trailing decimal point. Otherwise the number is assumed to be octal. For example, 12. represents 12 decimal (14 octal).

Table 3-1 on the next page describes the terminal handler parameters in [311,14] ISTT64PAR.MAC. See Appendix A for a listing of the file as it is distributed.

SYSTEM DEFINITIONS

Table 3-1  
Terminal Handler Parameters

<u>Parameter</u>	<u>Description</u>
TTYNUM	The total number of terminals for which the system is to be generated (including the console terminal).
DHUN	The number of DH11 (and DM11-BB) (16-line) units.
DJUN	The number of DJ11 (16-line) units.
DCUN	The number of DC11 (single line) units.
DL11E	The number of DL11E (single line) units.
DMPEA	The external page address of the DM11-BB unit attached to the first DH11, otherwise set to 0 if no DM11-BB. The second DH11 is assigned to the DM11-BB at DMPEA+10, etc.
DIAL	Non-zero if dial-up support is required; 0 if it is not.
PGEN	Non-zero if any DL11 terminals need parity; 0 if none need parity.
SECOND	The interval in number of TMUNITs between modem (DL11E, DM11-BB) inspections, that is, the interval at which a check is made for any dial up requests. If there is a request for a dial up connection, the "ring bit" is set.
TMUNIT	<p>Either 1 or 2 to indicate the time unit used to measure the interval between modem inspections where</p> <p>1 = Clock ticks (normally equivalent to line frequency)</p> <p>2 = Seconds</p>

More information about hardware interfaces can be obtained from the PDP-11 Peripherals Handbook.

## SYSTEM DEFINITIONS

### 3.2 SYSTEM GENERATION DIRECTIVES

Phases 1 and 2 of Stage 1 system generation are accomplished by issuing a series of commands to the system. The commands issued during phase 1 are called directives and are described in this chapter.

Phase 2 commands install all the tasks needed to complete the IAS system; these commands are described in Chapter 5. One phase 2 directive, \*DELAY (see section 3.2.6), exists to cause a 1-second delay in the processing of the command that follows the directive. Some command sequences may fail if a \*DELAY directive is not specified.

Phase 1 system generation directives are divided into six categories:

1. The TARGET directive that defines the target system device.
2. The PDP11 directive that defines the computer to be used.
3. The EXEC, SCOM, and PAR, directives that divide the computer memory into the Executive, the system communication area, and partitions, respectively.
4. The DEV directives that define the system peripherals.
5. The DPAR, CKPNT and SY directives that supply system default information.
6. The INS directives that install tasks needed for phase 2.

Regardless of whether the directives are to be typed in response to system generation requests or an indirect file is to be used, each category of directives is separated from the preceding category by a record containing only a slash (/). The end of input to phase 1 is indicated by a record containing two slashes (//). See the examples in Appendix A.

Many system generation directives can have multiple parameter sets for a single occurrence of the directive name. Multiple sets are separated from each other by backslash (\) characters. The examples below illustrate two PAR directives first and then the same two directives using the convention for multiple parameter sets.

```
PAR=JIM,,,U
PAR=JOHN,,,U
```

or

```
PAR=JIM,,,U\JOHN,,,U
```

To reach the point where the directives are entered, perform the appropriate steps as detailed in Chapter 6, section 6.1 through step 8. At this point, the following message is printed on the terminal.

```
SYSGEN PHASE1
SPECIFY TARGET DEVICE AND FILENAME
```

## TARGET

### 3.2.1 Target Directive

TARGET is an optional directive used to define the target device, that is, the device on which phase 1 is to create its output file and obtain all required tasks for installation. The TARGET directive has the following format.

```
TARGET=xyn:[ufd] filename.ext
```

where

xyn: is the device mnemonic and unit number. The default is SY0.

[ufd] is the UFD under which phase 1 is to store the system file. The default is [11,17].

filename.ext is the filename and extension used to name the file created by phase 1. The default is IAS.SAV.

If the TARGET directive is omitted, phase 1 uses the following file specification.

```
SY0:[11,17] IAS.SAV
```

3.2.1.1 Console Entry - When the message SPECIFY TARGET DEVICE AND FILENAME is printed on the console, the user can type the TARGET directive or the name of the indirect command file to be used. Alternatively, the user can type both or neither; see Chapter 6 section 6.1, steps 12 and 13. To omit the TARGET directive, simply proceed to input the PDP-11 directive described in section 3.2.2.

3.2.1.2 Description - If the TARGET directive is used, it completely describes where the output file of SGN1 is to be created and with what name. The following example creates a file named 64KSYS.SAV on DB1 under UFD [11,17].

```
SGN>TARGET=DB1:[11,17]64KSYS.SAV
```

If TARGET directive has been entered from the console, SGN1 prompts for the CPU specification, as demonstrated in the following example.

```
SGN>TARGET=DB1:[11,17]64KSYS
```

```
ENTER CPU SPECIFICATION
```

```
SGN>
```

3.2.2 PDP11 Directive

The PDP11 directive is used to define the target system processor configuration to system generation. One PDP11 directive is required. The directive has the following format:

```
PDP11=cpu-type,[mem-size],[fpt-optn],[clock-freq]
```

cpu-type = 45 to specify a PDP-11/45, or 70 to specify a PDP-11/70.

mem-size = physical memory size specified as nnnK words. nnn must be a decimal integer. The maximum value is 124. If this parameter is omitted, 64K is used as the memory size. The mem-size specified must be equal to or less than the amount of memory on the target hardware.

fpt-optn = FP to indicate that the floating point option is available in the hardware; otherwise, the parameter is omitted.

clock-freq = system clock ticks per second for the standard clock (KW11-L) or interrupts per second, clock identification, and clock ticks per interrupt for the programmable clock (KW11-P).

For the standard clock, specify a decimal number that is the line frequency (normally 50 or 60). The system runs with the line frequency clock at the specified frequency. If the parameter is omitted, 50 hz is used.

For a programmable clock, three decimal numbers are specified.

1. Frequency of clock interrupts per second; e.g., 1000.

2. Clock rate identification

```
0 = 100 KHz
1 = 10 KHz
2 = line frequency
3 = external
```

3. Number of clock ticks per interrupt

The three numbers must be enclosed in angle brackets and separated by commas as in the following example:

```
<100,0,1000>
```

The above example indicates that the programmable clock is to be used as the system clock. It is interrupting 100 times per second using the 100

## SYSTEM DEFINITIONS

KHz clock with a count of 1000 loaded into the count register.

3.2.2.1 Console Entry - When the message ENTER CPU SPECIFICATION is printed on the console, the user types the PDP11 directive. Enter the PDP11 directive followed by a record containing only a slash (/) as in this example.

```
SGN>PDP11=70,96K  
SGN>/
```

### 3.2.3 Memory Allocation

The memory allocation directives make it possible for the user to place the Executive and system common area in memory and to establish partitions. The allocation of memory is accomplished using the following directives:

- . EXEC (optional)
- . SCOM (required)
- . POOL (optional)
- . PAR (required)

These directives can be entered either from within an indirect command file, or in any order in response to the following request that is printed on the console.

#### SPECIFY DIVISION OF MEMORY

The memory allocation directives must be followed by a record containing only a slash (/).

## EXEC

3.2.3.1 EXEC Directive - The EXEC directive is used to load the IAS Executive into memory starting at a specified 32-word boundary. An address higher than the bootstrap program must be used. See Chapter 2, section 2.4.1 for bootstrap sizes.

The EXEC directive has the following format.

```
EXEC=base
```

where

base is a 32-word boundary at which the Executive is to start. The octal number specified in the EXEC directive is multiplied by 100 (octal) by system generation to determine the starting address. For example, if base is specified as 540, the starting address is memory location 54000 (octal).

The EXEC directive is optional. If the EXEC directive is not included, the Executive starts at the next 32-word boundary after the bootstrap program and base addresses specified in subsequent directives are ignored.

If the EXEC directive is included, a base address must also be specified in every directive that contains base as an optional parameter.

**SCOM**

3.2.3.2 SCOM Directive - The SCOM directive allocates space for the system common area. The system common area comprises the system subroutines, a communication region, the system tables (including the system task directory) and the node pool. The system common area cannot exceed 12K words.

One SCOM directive is required.

The SCOM directive has the following format:

SCOM=[base],size,std-entries

where

base is the starting address for the system common area. This location is specified using the same conventions as for the base address in the EXEC directive. If the EXEC directive is included in system generation, the base parameter must be part of the SCOM directive. If this parameter and the EXEC directive are omitted, the system common area follows the Executive in memory.

size is the total length of the system common area. The size can be specified as an octal number of 32-word blocks or as nnK. When the form nnK is used, the number (nn) is multiplied by 1024 words to determine the desired size. The factors involved in determining the size of SCOM are detailed below in Section 4.4.1.

std-entries is a decimal number equivalent to the number of system task directory entries. This parameter establishes the maximum number of simultaneously-installed tasks (user and system tasks) that the system is to support.

SCOM Size - The size of the system common area is the sum of the following elements,

1. The length in words of the system subroutines and the communication area,
2. The number of tasks installed at any given time multiplied by 16 words,
3. The number of system task directory (STD) entries specified by std-size, above,

## SYSTEM DEFINITIONS

4. The number of DEV directives multiplied by 26 words,
5. The number of PAR directives multiplied by 10 words,
6. Size of the variable-length node pool.

To determine the approximate amount of remaining space in the pool, subtract items 1 through 5 from the SCOM size parameter as follows.

$$\text{Pool size} = \text{SCOM size} - (\text{sum of items 1 through 5})$$

The node pool is used by the Executive and many of the system utilities as dynamic storage. Each node consists of a variable number of 8-word blocks. If the system runs out of pool space, its performance degrades and results may be unpredictable. Until experience is gained with the system, leave as much space as possible for the node pool.

**POOL**

3.2.3.3 POOL Directive - The POOL directive specifies the number of 16-word nodes to be prepicked for the terminal handler. These nodes are picked by phase 2 of system generation and placed in a linked list. When the terminal handler picks a node from the pool at the interrupt service routine level, the node allocation routines supply such nodes from the prepicked list.

The directive has the following format.

POOL=xxx

where

xxx is an octal number representing the maximum number of interrupt service level nodes to be made available to the Interrupt Service Routines (ISR). xxx can be in the range from 0 through 377 (octal).

The POOL directive is optional. If it is omitted, system generation prepicks one 16-word node for each device named Ttn.

If xxx is 0, no nodes are prepicked for the terminal handler.

**PAR**

3.2.3.4 PAR Directive - The partition directive establishes the name, base address, size, and type of every partition in the target system. Every partition in the system must be defined during system generation.

Multiple parameter sets are allowed.

The PAR directive has the following format.

PAR=par-name,[base],size,[par-type]

where

par-name is an up to 6-character ASCII name for the partition. It is converted to a radix 50 name by the system.

base is an optional starting address for the partition. This parameter is specified only if a base address is included in the EXEC directive and follows the same rules as those for the EXEC directive.

If base is omitted, the system places the partition optimally in memory.

size is the size of the partition specified as an octal number of 32-word blocks or as nnK. When nnK is used, nn is multiplied by 1024 to determine the desired size.

par-type is the partition type:

U for user-controlled,  
S for system-controlled,  
T for time-sharing.

If par-type is omitted, U is assumed.

See the IAS System Management Guide for information about partition types.

## NOTE

The partition which contains the system disk handler (SYDISK) must not be a timesharing partition. The partition which will be used to run timesharing (interactive and batch) tasks must be defined as a timesharing partition.

**DEV**

3.2.3.5 DEV Directive - The DEV directive is used to characterize completely each device in the system. Any device not defined by a DEV directive does not exist as far as IAS is concerned. Information required for the DEV directive is contained in Table 3-2. Appendix B contains a listing of device characteristic words. Appendix C describes how to define the characteristic words of teleprinters connected to DH11 or DJ11 lines.

The format of the DEV directive follows.

```
DEV=xyn,type,vect,pri,ext-page[,devacp]
```

where

xy                    a 2-character ASCII mnemonic to be used to refer to the device. (See Table 3-2).

n                     a 1-or 2-digit octal unit number.

type                  either a device type that phase 1 uses to determine the device characteristics or the actual device characteristics. Table 3-2 lists the device types recognized by phase 1. Appendix B lists the associated characteristics.

If the four device characteristics are to be specified they must appear between angle brackets as in the following example. Also see Appendix C.

```
DEV=XY0,<3,17,21,1000>,210,5,177000
```

vect                  address of the device interrupt trap vector. See Table 3-2.

pri                   The software priority at which the device's interrupts are to be serviced. The user should ensure that the software priority is equal to or higher than the hardware level at which the device interrupts unless the interrupt service routine of the handler is re-entrant. See Table 3-2.

ext-page             the external page address of the device controller. In most cases, ext-page is the lowest address when a device uses several external page addresses. See Table 3-2.

devacp                is an optional file primitives task name (or ACP name) for file-oriented devices.

An ACP (ancillary control processor) is a task that assists a device handler in managing a file-structured volume. An ACP also is referred to as a file primitives task under IAS.

If an ACP is not specified, phase 1 defaults to F11ACP for directory devices and MTAACP for magnetic tape.

SYSTEM DEFINITIONS

If an ACP task is specified, its name must be 6 characters ending in ACP. The first three characters are stored in the PUD and used at mount time. The ACP task must be installed before the device can be mounted.

Table 3-2  
Device Information

Mnemonic	Type	Normal Vector	Priority	External Page Address
DK	RK03	220	5	177400
	RK05	220	5	177400
DF	RFn(1)	204	5	177460
DP	RP03B	254	5	176710
	RP02	254	5	176710
DS	RS03	204	5	172040
DS	RS04(2)	204	5	172040
DB	RP04(2)	254	5	176700
TT	KSR33	Float(4)	4	Float
	KSR35	Float	4	Float
	VT05	Float	4	Float
	LA30P	Float	4	Float
	LA30S	Float	4	Float
	LA36	Float	4	Float
LP	LP11A(80Col)	200	4	177514
	LP11B(132Col)	200	4	177514
CR	CR11	230	6	177160
CR	CR11(3)	230	4	172460
MT	TU10	224	5	172520
MM	TU16	224	5	172440
PR	PTR	70	4	177550
PP	PTP	74	4	177554
DT	DT11	214	6	177340
CT	TAll	260	6	177500
AF	AF11	134	4	172570
AD	AD01	130	6	176770
LS	LPS11	Float	5	170400
UD	UDC11	234	6	171000
MO	None	None	None	None

(1) n = number of RF11 platters (1 to 8)

(2) These are disks on the RH11 massbus controller.

The DB unit number must correspond to the number assigned by physical connection to the RH11.

(3) Use this information for the CD11.

(4) The console teleprinter does not float. Its vector is 60 and its external page address is 177560.

## SYSTEM DEFINITIONS

### NOTE

For various combinations of the above devices, vector addresses and external page addresses may differ from those in Table 4-1. The correct vector and external page address for any configuration can be determined by Field Service Engineers. Useful information can also be found in PDP11 Peripherals Handbook.

Console Entry - Type the DEV directives in response to the following message.

### SPECIFY DEVICES

When all directives are entered, type a record containing a slash (/).

## SYSTEM DEFINITIONS

### 3.2.4 System Default Specifications

The next group of directives to be specified determine system defaults. The default directives may be specified in any order, either at the console or from within an indirect command file. These directives are:

SY  
DPAR  
CKPNT

A record containing a slash (/) must follow the last directive to be specified.

If the user is entering the directives at the console, they follow the prompt

SPECIFY DEFAULTS

which appears after the last device directive has been entered.

**SY**

3.2.4.1 SY Directive - The SY directive establishes the system disk to be used when phase 2 of system generation runs. The SY directive has the following format.

SY=xyn

where

xy = the 2-character mnemonic of the device.

n = the unit number of the device.

Phase 1 of system generation checks that the handler task for the device designated as SY is installed. For example, if DP0 is to be the system device, the user must install the handler DP.... during phase 1 of system generation (See section 3.2.6).

Care must be taken to ensure that the correct device is assigned to SY. Phase 1 of system generation assumes that all tasks it installs are to be resident on the SY of the target system and creates STDs accordingly.

Since some of these tasks are required during execution of Phase 2, it is necessary to bootstrap phase 2 from the target SY device. See Chapter 2, section 2.4 on software bootstrapping of IAS.

## **DPAR**

3.2.4.2 DPAR Directive - The DPAR directive specifies the default partition. Tasks and SGAs use the default partition if none other is specified at installation, runtime or task build.

The DPAR directive is required.

The DPAR directive has the following format.

DPAR=par-name

where

par-name is the name of the default partition. It must be a name defined by a PAR directive.

**CKPNT**

3.2.4.3 CKPNT Directive - The CKPNT directive enables checkpointing of real time tasks and specifies the disk and storage area to be allocated for checkpointing.

The directive is optional.

The CKPNT directive has the following format.

```
CKPNT=dev,size
```

where

dev is the name of the device on which the checkpoint area is to reside. The device must be named in a DEV directive.

size is the size of the checkpoint area. The size can be specified as nnK, where nn is the decimal number of 1024-word units, or as an octal number of 32-word blocks.

Description - The device named for checkpointing must contain a Files-11 structured disk. This disk volume must be mounted during phase 2 of system generation. To create a Files-11 structured disk, use the INI command in the SYSBLD.COMD command file that is executed during phase 2 of system generation or have a previously created volume on the device. The volume must be mounted during phase 2 of system generation.

The use of checkpointing is optional. However, if it is used, real memory space is allocated to support it. The amount of memory used depends on the checkpoint area of the disk. The relationship is one 32-word memory block for every 512K words of disk space specified for checkpointing. The checkpoint bitmap is at the end of the Executive. This memory allocation must be taken into account when specifying the size of partitions.

**INS**3.2.5 INS Directive

The INS directive installs the system tasks needed to provide the target system with the capability to do useful work. All installs for a given partition must be on the same line. The INS directive has the following format.

INS=par-name,file-specifier,file-specifier,...,file-specifier

where

par-name is a partition name.

file-specifier is an IAS file containing a task to be installed in the named partition. Up to five files can be specified in one INS directive. The file specifier must not include a device mnemonic and must be fewer than 32 (decimal) characters.

3.2.5.1 - Console Entry - After the user has entered the default directives, the system prompts the message

SPECIFY INSTALLS

The install directives are then entered, followed by a record consisting of 2 slashes (//).

3.2.5.2 Description - Minimally, the disk driver (DK...., DP...., or DB....), the teletype handler (TT....), the file system (F11ACP), and phase 2 of system generation (SGN2...) must be installed. The items in parentheses indicate the names of the specified tasks. See the INS directives in the system generation command files in Appendix A for examples.

To obtain a list of system tasks that can be installed, print a directory of UFD [11,1]. See Required Task Installations below.

All tasks installed during phase 1 are assumed to be on the device specified in the TARGET directive. It is further assumed that they are to be on the target SY when phase 2 executes.

No tasks installed during phase 1 can be bound to a shareable global area, nor can such tasks be multiuser.

No more than 15 tasks can be installed during Phase 1 of system generation. During Phase 2, the number of installs is limited only by the number of entries allowed in the system task directory (STD).

## SYSTEM DEFINITIONS

3.2.5.3 Required Task Installations - At the completion of phase 1, a target system exists that can operate as an independent IAS system. The final portion of phase 1 installs the tasks required to accomplish this end.

When a task is installed, it is given an entry in the system task directory (STD). This entry allows the system to load the task without making use of the file system.

Selection of tasks for installation depends on the following two factors:

1. The capabilities defined for the target system,
2. The procedures to be used in creating the operational target system.

If the target system is to operate as an independent IAS system following bootstrap, it must have tasks installed in it that are capable of doing the required work.

While processing the INS directives, phase 1 of system generation checks for the names of two system tasks: the disk handler for the system disk and phase 2 of system generation. These tasks are installed and are running when phase 2 is bootstrapped. The device handler is either DK....., DP....., DB.... depending on the device assigned to SY. Phase 2 of system generation is called SGN2. If these tasks are not named in the INS directive, the system is not operational when bootstrapped.

Normally other tasks must be installed in addition to the disk handler and SG2... . Typically the following additional tasks (identified by their filenames) are installed:

- . Install (INS),
- . Mount (ISMOU),
- . File system primitives (ISBIGFCP or ISFCP)
- . Teletype handler (ISTT64),

up to 15 tasks can be installed during phase 1.

File Primitives - Since phase 2 of system generation automatically mounts the system device, the appropriate file primitive tasks must be installed during phase 1. Two versions of the directory device file primitives are provided. Both have the task name FllACP. One has the filename ISFCP.TSK. It is a small, heavily overlaid version of the task. The other is named ISBIGFCP.TSK and is minimally overlaid.

If the user has his own file primitives, he should set the default ACP name in the DEV directive and install that task during phase 1.

Additional Tasks - With the minimum number of required tasks installed, the installation of any additional tasks can occur during phase 2. The decision to install additional tasks during phase 2 is partly procedural and partly due to the 15-task installation limitation of phase 1.

**\*DELAY**3.2.6 Phase 2 Directive (\*DELAY)

The \*DELAY directive causes a 1-second delay in the processing of phase 2 commands. Phase 2 sequentially retrieves and executes MCR commands from a file named [11,17]SYSBLD.CMD. Some command sequences may fail if a delay is not interposed between them during their execution. For example, the following sequence issued to replace a device handler initially loaded by phase 2 of system generation.

1. Request the new device handler.
2. Unload the existing device handler.
3. Specify several delays to give the system time to install and run the new device handler.

Without the delays, the next MCR command may fail due to the nonexistence of a necessary handler.

The \*DELAY directive has no parameters.

## CHAPTER 4

### TIMESHARING EXECUTIVE PARAMETERS

The purpose of Stage 2 system generation is to revise the Timesharing Executive so that it reflects current installation interactive and batch processing requirements. The user specifies the installation requirements by editing a MACRO source file called [311,107]SGDATA.MAC. The file contains the parameters needed by IAS to control timesharing activity.

This chapter defines the individual timesharing parameters. The IAS System Management Guide describes how to set the parameter values to produce a Timesharing Executive tailored to the needs of the installation. The parameters define the IAS data structures contained in the libraries IASCOM and IASBUF. The data structures are described in Chapter 9 of the IAS Executive Reference Manual-Volume II. They include

1. The User Job Node (UJN)
2. The User Terminal Node (UTN)
3. The Command Interpreter Table (CIT)
4. The Device Table (DVT)
5. The User Task List (UTL)
6. The Swap File Table (SFT)
7. The Job Node Pool (JNP)
8. The Terminal Node Pool (TNP)

Stage 2 determines the size and contents of all the tables, lists and node pools. Parameters that determine the contents may be overridden at System Startup, but Stage 2 system generation must be performed in order to change the size of tables, lists and node pools and the number of nodes within a pool. Each description of the parameters below includes the equivalent Startup command if one exists. See the IAS System Management Guide for System Startup information.

## TIMESHARING EXECUTIVE PARAMETERS

Note that:

1. Timesharing terminals include any device from which a Command Language Interpreter (CLI) can receive input, for example, a card reader (CR), a teleprinter terminal (TT) or spooled input (SP).
2. Decimal numbers in MACRO-11 source files are specified with a trailing decimal point. Otherwise the number is assumed to be octal. For example, 12. represents 12 decimal (14 octal).

### 4.1 SET-UP PARAMETERS

**SGLUN** The maximum number of luns that each user can have assigned at anyone time. In PDS terms this means the maximum number of lun assignments which can be made with the ASSIGN command.

SGLUN generates the lun table section of a terminal node.

**SGUVL** The maximum number of devices that each user can have allocated at any one time.

SGUVL generates the device map table in the terminal node.

**SGDEV** The maximum number of devices which may be available for interactive and batch jobs. Can include terminals if they are not timesharing terminals.

SGDEV generates the available device section of the DVT table with the specified number of entries for non-terminal devices.

**SGTMM** The maximum number of timesharing terminals to be supported at any one time.

SGTMM generates additional entries in the DVT for terminal devices and a User Terminal Node (UTN) for each terminal.

**SGXBF** The number of timesharing buffers. Should normally be the sum of the active terminals (SGATT) and the number of non-CLI timesharing tasks (SGXTSK) + 1.

SGXBF generates the buffer pool (IASBUF) used by Command Language Interpreters (CLIs) and Timesharing Control Primitives (TCP).

**SGATT** The number of active timesharing terminals at any one time. This parameter generates one job node (UJN) for each CLI task and will normally be set equal to SGTMM.

## TIMESHARING EXECUTIVE PARAMETERS

**SGXTSK** The number of non-CLI timesharing tasks. This parameter generates the specified number of job nodes (UJNs) in addition to those allocated for CLIs. In a PDS system where at present each active PDS CLI requires one additional job node this parameter would be set equal to SGATT.

**SGTKSZ** The maximum swap size allowed for a timesharing task in units of 32 decimal words or 100 octal bytes. A task with a swap size exceeding the specified size will not be allowed to execute.

Equivalent start-up command is SET SWAP.

**SGBUF** The number of bytes available in a timesharing buffer. The system will ensure that this value is even and not less than 80 bytes.

**SGCIT** The maximum number of CLIs which can be concurrently declared to the system via the System Control Interface. Should be a minimum of 2 to allow SCI and PDS to be concurrently declared.

SGCIT generates the Command Interpreter Table (CIT) with the specified number of entries.

**SGUTL** The number of timesharing scheduling levels. Must be 2 or more if interactive jobs are to be given preference over batch jobs.

SGUTL generates the User Task List (UTL).

### 4.2 SCHEDULING PARAMETERS

**SGSPR** Ticks between scheduler promotions. The number of ticks of elapsed time after which the scheduler will promote a task up one level if no task at that level has been scheduled.

Equivalent Startup command is SET QUANTUM.

**SGSTK** Scheduler idle time. Should be set to 1.

**SGCON** Quantum constant. The minimum time quantum given to a job when it is scheduled.

Equivalent Startup command is SET QUANTUM.

**SGTSL** Maximum time slice. The maximum number of ticks a task can run before a timesharing reschedule will occur.

Equivalent Startup command is SET QUANTUM.

## TIMESHARING EXECUTIVE PARAMETERS

- SGAFM Allocation factor memory size. The number of 32 decimal word blocks (or 100 octal bytes) of memory used to allocate quanta in relation to task size.
- Equivalent Startup command is SET QUANTUM.
- SGAFT Allocation factor ticks/memory size. The number of ticks allocated per unit of memory as specified by SGAFM. For example: if SGAFM = 32 (32 \* 32 = 1024 words) and SGAFT = 2 then 2 ticks of CPU time will be allocated for each 1K words of the task.
- SGBSH The number of ticks between batch schedules.
- Equivalent Startup command is SET BATCH.
- SGBQT The number of ticks of CPU time allocated to batch each time it is scheduled.
- Equivalent Startup command is SET BATCH.
- SGT1PR The priority of TSS1 in the Active Task List (from 2. to 240.).
- Equivalent Startup command is SET PARTITION.
- SGPIPR The priority of the TCP handler in the Active Task List (from 2. to 240.). This value must be higher than SGT1PR.
- Equivalent Startup command is SET PARTITION.

### 4.3 SWAPPING PARAMETERS

- SGSPR The maximum number of swap files which will be used to create the swapping space.
- SGLBSZ The logical swap block size. This is the number of physical disk blocks (256 words) which comprise one logical swap block. For example, if SGLBSZ = 4 the task swap areas will be allocated in 1K word units.
- Equivalent Startup command is SET SWAP.
- SGFLSZ The maximum size of any of the swap files. Specified in terms of logical swap blocks.
- Equivalent Startup command is SET SWAP.

## CHAPTER 5

### TRANSFERRING THE DISTRIBUTED SYSTEM

This chapter lists the steps to be followed to transfer the IAS system distributed on magnetic tape onto the system disk, either an RK05, RP02, RP03 or RP04.

Most of the commands contained in this chapter and Chapters 6 and 7 are issued to the Monitor Console Routines (MCR) (see Appendix D). MCR is the interface used for system generation and startup. MCR is designed to time out after five minutes. When this happens, as indicated by a carriage return and line feed, press CTRL and C simultaneously to reactivate MCR. Because the timeout interval starts when the MCR prompt (MCR>) is printed, it is possible for MCR to time out while the user is typing a command; in this case the entire line must be retyped.

#### 5.1 HARDWARE BOOTSTRAP

In order to transfer the distribution medium onto the system disk, the distribution medium must be bootstrapped. Four models of hardware bootstraps are available on systems used for IAS: MR11DB, BM792YB, BM873YA and BM873YB. The type of bootstrap for a particular PDP-11 can be determined by consulting the equipment order.

Whenever a request to bootstrap the system is encountered in the following text, refer to one of the four sections that follow to perform the appropriate bootstrap.

##### 5.1.1 MR11DB Bootstrap

Perform the following steps to use an MR11DB Bootstrap.

1. On the console switches, set HALT.
2. Set ENABLE.
3. Enter the address of the device from which the bootstrap is to occur into the console switches. Table 5-1 provides the device addresses.
4. Press LOAD ADDR.

5. Press START.

<u>Table 5-1</u> <u>Device Addresses for the MR11DB Bootstrap</u>	
<u>Device</u>	<u>Address</u>
RP03 Disk	173154
RK Disk	173110
RF Disk	173100
TU10 Magnetic Tape	173136
DEctape	173120

5.1.2 BM792YB Bootstrap

Perform the following steps to use a BM792YB Bootstrap.

1. On the console switches, set HALT.
2. Set ENABLE.
3. Enter 173100 into the display switches.
4. Press LOAD ADDR.
5. Enter the address of the device from which the bootstrap is to occur into the console switches. Table 5-2 provides the device addresses.
6. Press START.

<u>Table 5-2</u> <u>Device Addresses for the BM792YB Bootstrap</u>	
<u>Device</u>	<u>Address</u>
RP03 Disk	176716
RK Disk	177406
RF Disk	177462
DEctape	177344

## TRANSFERRING THE DISTRIBUTED SYSTEM

Magnetic tape cannot be booted with the BM792YB Bootstrap. To bootstrap from a TU10 perform the following sequence of operations.

Mount the magnetic tape reel containing the distribution system on magnetic tape unit 0. The tape must be at the load point. Turn the unit ON LINE. (For more detailed instructions see the magnetic tape hardware manual.)

Move the CPU ENABLE/HALT switch to its HALT position.

Deposit the following routine in memory via the console switch registers.

<u>Address</u>	<u>Contents</u>	<u>Symbolic</u>
10000	12700	MOV #172524,R0
10002	172524	
10004	5310	DEC (R0)
10006	12740	MOV #60011,(R0)
10010	60011	
10012	105710	TSTB (R0)
10014	100376	BPL .-2
10016	5710	TST (R0)
10020	100767	BMI .-20
10022	12710	MOV #60003,(R0)
10024	60003	
10026	105710	TSTB (R0)
10030	100376	BPL .-2
10032	5710	TST (R0)
10034	100777	BMI .
10036	5007	CLR PC

1. Set the CPU Switch Register to 010000.
2. Depress the LOAD ADDR switch.
3. Move the CPU ENABLE/HALT switch to its ENABLE position.
4. Depress the CPU START switch.

If the machine hangs in the RUN state at location 10034, a magnetic tape error has occurred. Halt the machine, rewind the magnetic tape manually, and restart the bootstrap procedure.

### 5.1.3 BM873YA Bootstrap

Perform the following steps to use the BM873YA Bootstrap.

1. On the console switches, set HALT.
2. Set ENABLE.

## TRANSFERRING THE DISTRIBUTED SYSTEM

3. Enter the address of the device from which the bootstrap is to occur into the console switches. Table 5-3 provides the device addresses.
4. Press LOAD ADDR.

### NOTE

If a unit other than 0 contains the device to be booted, set the switch register to the unit number of the device to be booted before pressing START.

5. Press START.

<u>Table 5-3</u> <u>Device Addresses for BM873YA Bootstrap</u>	
<u>Device</u>	<u>Address</u>
RF11 (RFn disk)	773000
RP11 (RP03 disk)	773100
RK11 (RK05 disk) -- Unit 0	773010
Unit specified in switch register	773020
TC11 (DECTape)	773030
TM11 (TU10 magnetic tape)	773050

### 5.1.4 BM873YB Bootstrap

Perform the following steps to use the BM873YB bootstrap.

1. On the console switches, set HALT.
2. Set ENABLE.
3. Enter the address of the device from which the bootstrap is to occur into the console switches. Table 5-4 provides the device addresses.
4. Press LOAD ADDR.

### NOTE

If a unit other than 0 contains the device to be booted, set the switch register to the unit number of the

TRANSFERRING THE DISTRIBUTED SYSTEM

device to be booted before pressing  
START.

5. Press START.

<u>Table 5-4</u> <u>Device Addresses for BM873YB Bootstrap</u>	
<u>Device</u>	<u>Address</u>
RH11 (RS03,4 disk) -- unit 0	773000
Unit specified in switch register	773002
RK11 (RK05 disk) -- unit 0	773030
Unit specified in switch register	773032
RH11 (RP04 disk) -- unit 0	773320
Unit specified in switch register	773322
RP11 (RP03 disk) -- unit 0	773350
Unit specified in switch register	773352
RF11 (RS11 disk)	773136
TC11 (DEctape)	773070
TM11 (TU10 magnetic tape)	773110
RH11 (TU16/TM02 magnetic tape)	773150

5.2 TRANSFERRING THE DISTRIBUTION MEDIUM FROM MAGNETIC TAPE

The following paragraphs detail the procedures to be used to transfer a distribution system from magnetic tape onto an RK05, RP02, RP03 or RP04 system device.

5.2.1 Bootstrapping the Distribution Tape

1. Place the distribution tape on unit 0.
2. Bootstrap the distribution tape following the procedures in Section 5.1. The system prints the following on the console:

```
IAS SYSTEM DISTRIBUTION TAPE

SYSTEM DISK?
```

3. Respond with one of the following to indicate the type of system disk:

```
RK05  for RK05 system disk
RP02  for RP02 system disk
RP03  for RP03 system disk
RP04  for RP04 system disk
```

4. Once the name of the system device has been typed, the system prints the following message or its equivalent:

```
LOAD SCRATCH DISK ON xy0 WRITE ENABLED
TYPE 'CR' WHEN READY?
```

where xy is the device type.

5. When the disk is ready type carriage return, and if the disk can be formatted, the system prints the following:

```
FORMAT DISK?
```

6. Respond with one of following:

```
YES  to format the system disk.
NO   if disk formatting is not required.
```

7. The system then formats the disk if so requested and prints the following message:

```
RUN 'BADBLOCKS'?
```

8. Respond with one of the following:

```
YES  to run the bad block utility on the system
      disk
NO   if this utility is not required
```

## TRANSFERRING THE DISTRIBUTED SYSTEM

9. The system tests the system disk for bad blocks if this option was selected and then types:

STANDARD VOLUME INITIALIZATION?

10. Respond with one of the following:

YES if standard volume initialization required

NO to make the initialization task prompt for the initialization parameters.

See Appendix D for a description of the MCR command INITIALIZE VOLUME.

11. The system is then created. The information in Figure 5-1 is printed on the console while the system is created. When the system prints the following message the system building procedure is complete.

\*\*\*END OF SYSTEM GENERATION PHASE 2\*\*\*

12. The system should be saved by typing CTRL/C to invoke MCR and by typing the following commands:

MCR>TIM dd-mm-yy hh:mm:ss

MCR>DMO SY:

MCR>SAV

13. The system is now ready for use. It is a single user system which can be used to generate a system tailored to the installation's hardware and software requirements.

TRANSFERRING THE DISTRIBUTED SYSTEM

IAS SYSTEM DISTRIBUTION TAPE

SYSTEM DISK? RP03  
 LOAD SCRATCH DISK ON DP0 WRITE ENABLED  
 TYPE 'CR' WHEN READY?  
 FORMAT DISK? NO  
 RUN 'BADBLOCKS'? NO  
 STANDARD VOLUME INITIALISATION? YES

RED DP=SY  
 INI SY: IASSYS/BAD=[AUTO]  
 INI -- CHECKING DP0:  
 MOU SY:/OVR  
 MOUNT-\*\*\*VOLUME INFORMATION\*\*  
     DEVICE =DP0  
     CLASS =FILE 11  
     LABEL =IASSYS  
     UIC =[1,1]  
     ACCESS =[RWED, RWED, RWED, RWED]  
     CHARAC =[ ]  
 UFD SY:[1,1]/PRO=[RWED, RWED, R, R]  
 UFD SY:[1,2]/PRO=[RWED, RWED, R, R]  
 UFD SY:[1,3]  
 UFD SY:[1,4]/PRO=[RWED, RWED, RWED, RWED]  
 UFD SY:[1,5]/PRO=[RWED, RWED, R, R]  
 UFD SY:[1,6]/PRO=[RWED, RWED, RWED, RWED]  
 UFD SY:[1,100]/PRO=[RWED, RWED, RWED, RWED]  
 UFD SY:[1,101]  
 UFD SY:[11,1]  
 UFD SY:[11,14]  
 UFD SY:[11,15]  
 UFD SY:[11,17]  
 UFD SY:[11,100]  
 UFD SY:[11,101]  
 UFD SY:[11,102]  
 UFD SY:[11,103]  
 UFD SY:[11,105]  
 UFD SY:[11,106]  
 UFD SY:[11,107]  
 UFD SY:[111,14]  
 UFD SY:[111,15]  
 UFD SY:[111,17]  
 UFD SY:[111,100]  
 UFD SY:[111,101]  
 UFD SY:[111,102]  
 UFD SY:[111,103]  
 UFD SY:[111,105]  
 UFD SY:[111,106]  
 UFD SY:[111,107]  
 UFD SY:[200,200]  
 UFD SY:[211,14]  
 UFD SY:[211,107]  
 UFD SY:[311,14]  
 UFD SY:[311,107]  
 CPY

CPY -- STARTING COPY

TRANSFERRING THE DISTRIBUTED SYSTEM

CPY -- COPY COMPLETE (ALLOW TAPE TO REWIND)

RED DP=SY

MOU SY:/OVR

MCOUNT-\*\*\*VOLUME INFORMATION\*\*

DEVICE =DP0  
 CLASS =FILE 11  
 LABEL =IRSSYS  
 UIC =[1,1]  
 ACCESS =[RWED,RWED,RWED,RWED]  
 CHARAC =[ ]

INX SYSRES/LI/ACC=RO/UIC=[1,1]

INX [11,1]INS

UNF ... RED,... INX,... MOU,... UNF

INS [11,1]CREUPF/TASK=... CRE

CRE

INS [11,1]LBR

INS [11,1]PIP

LBR SYSLIB/CO:300.:2000.:750.=SYSLIB

PIP SYSLIB.OLB/PU

INS [11,1]B00

INS [11,1]INV

INS [11,1]SGN1/TASK=... SG1/UIC=[11,17]

SG1 @SY:[11,17]RPF03L64K

PDP11=45,64K

/

SCOM=,340,96

PAR=SYSDSK,,35,U

PAR=SYS,,114,T

PAR=FILE,,121,U

PAR=TTY,,130,U

PAR=GEN,,2447,T

/

DEV=DK0,RK05,220,5,177400

DEV=DK1,RK05,220,5,177400

DEV=DP0,RP03B,254,5,176710

DEV=TT0,LA305,60,4,177560

DEV=LP0,LP11B,200,4,177514

DEV=DT0,DT11,214,6,177340

DEV=DT1,DT11,214,6,177340

DEV=MT0,TU10,224,5,172520

DEV=MM0,TU16,224,5,172440

DEV=MO,, , ,

DEV=SP,<10000,0,0,0>,, ,

DEV=PI,, , ,

DEV=TO,, , ,

/

DPAR=GEN

CKPNT=DP0,50K

SY=DP0

/

INS=SYSDSK,[11,1]DP

INS=GEN,[11,17]SGN2,[11,1]ISMOU,[11,17]INZ/UIC=[1,1]

INS=FILE,[11,1]ISFCP

INS=TTY,[11,1]ISTT64

//

SGN>END OF PHASE 1

B00 SY:[11,17]IAS.SAV/WB

B00 SY:

TRANSFERRING THE DISTRIBUTED SYSTEM

\*\*\* SYSTEM GENERATION PHASE 2 \*\*\*

```

MOU DP0:/OVR
MOUNT-***VOLUME INFORMATION**
      DEVICE  =DP0
      CLASS   =FILE 11
      LABEL   =IASSYS
      UIC     =[ 1, 1 ]
      ACCESS  =[ RWED, RWED, RWED, RWED ]
      CHARAC  =[ ]
INZ [ 11, 1 ]MCR
INZ SYSRES/LI/ACC=RO/UIC=[ 1, 1 ]
*DELAY
INZ [ 11, 1 ]INS
*DELAY
INS [ 11, 1 ]FIX
*DELAY
FIX F11ACP
*DELAY
INS IASCOM/LI/ACC=RO/UIC=[ 1, 1 ]
INS IASBUF/LI/ACC=RO/UIC=[ 1, 1 ]
INS PDELIB/LI/ACC=RO/UIC=[ 1, 1 ]
*DELAY
INS [ 11, 1 ]ISDT
INS [ 11, 1 ]JDK
INS [ 11, 1 ]JLP
INS [ 11, 1 ]JMO
INS [ 11, 1 ]JISTU10
INS [ 11, 1 ]JISTU16
INS [ 11, 1 ]JQUE
INS [ 11, 1 ]JISSPR
INS [ 11, 1 ]JSPR2
INS [ 11, 1 ]JISOPR
INS [ 11, 1 ]JTKTN
INS [ 11, 1 ]JABO
INS [ 11, 1 ]JACT
INS [ 11, 1 ]JALT
INS [ 11, 1 ]JBOO
INS [ 11, 1 ]JCOM
INS [ 11, 1 ]JISDEV
INS [ 11, 1 ]JISDMO
INS [ 11, 1 ]JEDI
INS [ 11, 1 ]JFLX
INS [ 11, 1 ]JF11MSG
INS [ 11, 1 ]JINI
INS [ 11, 1 ]JLOA
INS [ 11, 1 ]JLUN
INS [ 11, 1 ]JPURMAC
INS [ 11, 1 ]JOPF
INS [ 11, 1 ]JPAR
INS [ 11, 1 ]JPURPIP
INS [ 11, 1 ]JREA
INS [ 11, 1 ]JISRED
INS [ 11, 1 ]JREM
INS [ 11, 1 ]JRUN
INS [ 11, 1 ]JSAV
INS [ 11, 1 ]JSET
INS [ 11, 1 ]JSLP
INS [ 11, 1 ]JTAS
INS [ 11, 1 ]JTIM

```

TRANSFERRING THE DISTRIBUTED SYSTEM

```
INS [11,1]JUF0
INS [11,1]JUNF
INS [11,1]JUNL
INS [11,1]JUSERS
INS [11,1]JACCOUNTS
INS [11,1]JRESET
INS [11,1]JSCI
INS [11,1]JIAS
INS [11,1]JIASTART
INS [11,1]JTCP
INS [11,1]JTSS1
INS [11,1]JTSS3
INS [11,1]JTSSNUL
INS [11,1]JTKB
INS [11,1]JDMP
INS [11,1]JLBR
*DELAY
REM ... INZ
*DELAY
*DELAY
*** END OF SYSTEM GENERATION PHASE 2 ***
MCR>TIM 5-NOV-75 17:6:0
MCR>DMO SY:
DMO -- DP0: ** DISMOUNT COMPLETE **
MCR>
```

Figure 5-1  
IAS Bootstrap



## CHAPTER 6

### STAGE 1 PROCEDURES

IAS provides two tasks (phase 1 and phase 2) and a set of command files for Stage 1 of system generation. Its purpose is to create a file on disk that is an IAS system capable of running on the target hardware configuration.

Phase 2 of system generation is designed to perform those functions that require the target configuration. Phase 2 is a task that is activated in the system created by phase 1.

Phase 1 (SGN1) should be the only task running in the system.

Both phases accept input from command files. Phase 1 also accepts input from a terminal. (See Chapter 3, section 3.2 for details on console entry.) The distributed system includes several system generation command files under UFD [11,17]. The command files that can be used as input to phase 1 all have the following format for the filename and type.

diskcnnk.CMD

disk is the disk type that is to be the system device, that is, RK05, RP02, RP03 or RP04.

c is the clock type; L for line clock or P for programmable clock.

nn is the amount of memory supported by system generated using this file.

Any one of these files can be used as input to phase 1.

Appendix A provides sample listings of the files.

The phase 2 input file is SYSBLD.CMD. It is located under UFD [11,17]. Phase 2 automatically uses this file to create a running system on the target hardware. SYSBLD.CMD can be edited prior to running phase 2 to reflect local requirements. Appendix A contains a listing of SYSBLD.CMD.

## STAGE 1 PROCEDURES

### 6.1 PHASE 1 OF SYSTEM GENERATION

Phase 1 of Stage 1 of system generation uses the defaults of the system under which it is running and communicates with the user on the console terminal. The purpose of phase 1 is to create a file on a disk. Once the file is created, the disk is called the target system disk.

Care should be taken if the target system disk and the host system disk are the same; several tasks are installed in the target system during phase 1 that are also installed in the host system. Since task headers are modified by the install function, a task installed in one operating system cannot be used in another even though it was installed there previously. Device assignments, for example, may be different.

#### NOTE

Throughout this manual, the use of lower case letters in command strings indicates that the user is to supply the appropriate variables at that point in the command. For example, in the following MCR command, the user is to specify the device type and unit of the target disk.

MCR>MOU target disk

could be typed as

MCR>MOU DP0:

The procedures for executing phase 1 on a running IAS system follow.

1. Install system generation phase 1 and a special version of the install function as follows:

```
MCR>SET /UIC=[1,1]
MCR>INS [11,1]SGN1
MCR>INS [11,1]INV
```

2. If the target disk is the current system disk, go to step 5. If steps 2 through 4 have been done previously, go to step 5. Otherwise, initialize and mount the target disk, as follows:

MCR>INI target disk            Use any appropriate INITVOL options (see Appendix D).

MCR>MOU target disk            Use any appropriate MOUNT options (see Appendix D).

STAGE 1 PROCEDURES

3. Create the required UFD's on the target disk using the following commands:

```
MCR>UFD target disk[1,1]/PRO=[RWED,RWED,R,R]
MCR>UFD target disk[1,2]/PRO=[RWED,RWED,R,R]
MCR>UFD target disk[1,3] (for Verify)
MCR>UFD target disk[1,4]/PRO=[RWED,RWED,RWED,RWED]
                               (to allow spooling)
MCR>UFD target disk[1,5]/PRO=[R,RWED,R,R]
MCR>UFD target disk[1,6]/PRO=[RWED,RWED,RWED,RWED]
                               (for error logging)
MCR>UFD target disk[1,100]/PRO=[RWED,RWED,RWED,RWED]
MCR>UFD target disk[1,101]
MCR>UFD target disk[11,1]
MCR>UFD target disk[11,14]
MCR>UFD target disk[11,15]
MCR>UFD target disk[11,17]
MCR>UFD target disk[11,100]
MCR>UFD target disk[11,101]
MCR>UFD target disk[11,102]
MCR>UFD target disk[11,103]
MCR>UFD target disk[11,105]
MCR>UFD target disk[11,106]
MCR>UFD target disk[11,107]
MCR>UFD target disk[111,14]
MCR>UFD target disk[111,15]
MCR>UFD target disk[111,100]
MCR>UFD target disk[111,101]
MCR>UFD target disk[111,102]
MCR>UFD target disk[111,103]
MCR>UFD target disk[111,105]
MCR>UFD target disk[111,106]
```

STAGE 1 PROCEDURES

MCR>UFD target disk[111,107]

MCR>UFD target disk[211,14]

MCR>UFD target disk[211,107]

MCR>UFD target disk[311,14]

MCR>UFD target disk[311,107]

4. Transfer all files required for the target system. Both the target disk and the current system disk must be write enabled.

MCR>SET /UIC=[1,1]

MCR>PIP target disk =\*.\*

MCR>PIP target disk[1,2]=[1,2]\*.\*

MCR>SET /UIC=[11,1]

MCR>PIP target disk[1,100]=[1,100]\*.\*

MCR>SET /UIC=[11,17]

MCR>PIP target disk=\*.\*TSK

MCR>PIP target disk=\*.\*STB

MCR>PIP target disk=\*.\*CMD

MCR>SET /UIC=[1,1]

MCR>PIP target disk[1,101]=[1,101]\*.\*

MCR>PIP target disk[11,1]=[11,1]\*.\*

MCR>PIP target disk[11,100]=[11,100]\*.\*

MCR>PIP target disk[11,101]=[11,101]\*.\*

MCR>PIP target disk[11,102]=[11,102]\*.\*

MCR>PIP target disk[11,103]=[11,103]\*.\*

MCR>PIP target disk[11,105]=[11,105]\*.\*

MCR>PIP target disk[11,106]=[11,106]\*.\*

MCR>PIP target disk[11,107]=[11,107]\*.\*

MCR>PIP target disk[311,107]=[311,107]\*.\*

## STAGE 1 PROCEDURES

```
MCR>PIP target disk[311,14]=[311,14]ISTT64PAR.MAC
```

```
MCR>PIP target disk[311,14]=[311,14]ISTT64.MAC
```

```
MCR>PIP target disk[311,14]=[311,14]ISTT64MAC.CMD
```

```
MCR>PIP target disk[11,14]=[11,14]ISTT64TKB.CMD
```

5. The files on the target disk are now available to be used to generate the new system. The pseudo device 'SY' should be redirected to the target disk by entering the following commands. The reedirect will cause all subsequent operations which do not explicitly specify a device name to default to the target disk.

```
MCR>DMO SY0:
```

```
MCR>DMO target disk:
```

```
MCR>RED target disk=SY
```

```
MCR>MOU target disk
```

```
MCR>MOU old system disk
```

6. The terminal handler parameter file [311,14]ISTT64PAR.MAC, should be edited to specify the terminals and interfaces of the target system. If it is not necessary to change the terminal handler proceed to step 7. The parameters are described in Chapter 3, section 3.1. The following commands will perform all operations from the target disk.

```
MCR>SET /UIC=[1,1]
```

```
MCR>EDI [311,14]ISTT64PAR.MAC
```

When the file has been edited the new terminal handler can be assembled and built by issuing the following commands.

```
MCR>MAC @[311,14]ISTT64MAC
```

```
MCR>TKB @[11,14]ISTT64TKB
```

A new version of the terminal handler [11,1]ISTT64.TSK, now exists on the target disk. The size of the handler is variable according to the parameter settings. This size should be determined in order to generate the target system with a partition of the correct size to hold the teletype handler (TTY partition).

If the TTY partition is larger than the size of the handler space in the partition may remain unused. If the TTY partition is smaller than the size of the handler a fatal error will be produced by SGN1.

The size of the terminal handler can be easily determined by installing it with a temporary task name, issuing a TASK command and removing the task.

## STAGE 1 PROCEDURES

For example,

```
MCR>INS [11,1]ISTT64/TASK=AAAAAA/PAR=GEN
MCR>TAS
AAAAAA V004A GEN 248 15200 DB 0-00000007202

MCR>REM AAAAAA
```

The printing of the complete task list can be suppressed by typing CTRL/O. In the above example, the handler size is 15200 octal bytes; therefore, the size of the TTY partition should be set to 152.

7. The command file to be used as input to phase 1 should be edited to define the target system. If a suitable file currently exists proceed to step 8. The following commands should be used to run the editor:

```
MCR>SET /UIC=[1,1]
MCR>EDI [11,17]command file
```

It is important to check that the following conditions are satisfied:

- a. The size of the SYDISK partition is large enough to hold the system disk handler.
- b. The SYDISK partition is not timesharing controlled.
- c. The TTY partition is large enough to hold the terminal handler (see step 6).
- d. The SY directive correctly names the target system disk.
- e. The disk handler for the target system disk is installed during phase 1.
- f. The CKPNT directive names the correct device on which to create the checkpoint area (normally the target system disk).

If the sizes of any partitions are changed (for example, TTY) the size of other partitions (for example, GEN) must be altered correspondingly.

8. Depending on the available pool size of the host system, it may be necessary to remove some tasks from the host system. Chapter 2, section 2.2, provides guidelines for calculating node pool usage during system generation.

A file under [11,17] called SYSGENREM.COMD can be used to remove tasks. It is advisable to use this file. Enter the following command to remove installed tasks using SYSGENREM.COMD.

```
MCR>REM @[11,17]SYSGENREM.COMD
```

## STAGE 1 PROCEDURES

9. If FllACP has not been fixed in memory, fix it using the following command.

```
MCR>FIX FllACP
```

This step is recommended for target system generations and is necessary if generating on the current system disk.

10. At this point, phase 1 of system generation can be executed. It is preferable to execute it under UIC [11,17]. Use the following command to set the UIC.

```
MCR>SET /UIC=[11,17]
```

11. Run phase 1 ensuring that the command is terminated by pressing ^^Altmode (ESC).

```
MCR>RUN SGN1          Depress Altmode
```

12. The system responds as follows.

```
SYSGEN PHASE 1
```

```
SPECIFY TARGET DEVICE AND FILENAME
```

```
SGN>
```

13. At this point, the user has the option of naming the device, UFD, and filename to be assigned to the file produced by phase 1. The directive has the following format.

```
SGN>TARGET=td:[ufd]filename
```

td indicates the target disk. If it is omitted, SY is used by default.

[ufd] is the UFD of the system image file. If it is omitted [11,17] is used.

filename is the name to be assigned to the file. If it is omitted, IAS.SAV is used.

If the TARGET directive is not specified, the default is SY:[11,17]IAS.SAV.

## STAGE 1 PROCEDURES

14. Name the command file to be processed by phase 1 of system generation (SGN1). The command file can be any one of those provided with the system, e.g., RP04L64K.CMD, or it can be a user created file. The following command to SGN1 is used.

SGN>@command file specification

All of the commands read from the command file are printed on the terminal. Upon successful completion of the processing of the command file, the following message appears on the terminal.

SGN>END OF PHASE 1

Depress CTRL/C to obtain MCR.

The host system remains usable in the normal way even if fatal errors occurred during execution of phase 1.

15. If any errors occurred during phase 1, correct them (consult Appendix E) and repeat steps 10 through 14.

### NOTE

Fatal errors must be corrected, but diagnostic messages imply that the resulting system may be usable.

16. Phase 1 of system generation does not produce a target system that is hardware bootable, i.e., it does not write physical block 0 of the disk.

## 6.2 PHASE 2 OF SYSTEM GENERATION

Phase 1 of system generation has created a file on the target system disk. This file contains a memory image ready for phase 2.

In this system, two tasks are active. One is the system disk handler, and the other is phase 2 of system generation. Bootstrapping this file into memory causes phase 2 to run.

### NOTE

For phase 2 to execute successfully, it must be booted on a machine with as much or more memory than that specified during phase 1.

Phase 2 of system generation processes a command file named [11,17]SYSBLD.COMD. If any modifications need to be made to this file to reflect system requirements, they must be made by means of an editor prior to execution of phase 2 (see Chapter 2, section 2.1).

The steps necessary to run phase 2 follow.

1. Bootstrap phase 2. There are two ways to bootstrap phase 2 into memory on the target system. Whichever way is used, it must be booted from the unit specified as SY during phase 1. Methods a and b follow.
  - a. Create a bootstrap block 0 on the target disk and use the hardware bootstrap. Issue the following command on the host system to create bootstrap block 0.

```
MCR>BOO target device:filename/WB
```

filename is the name specified in the TARGET directive.

This command takes virtual block 1 of the file created by phase 1 and writes it on physical block 0 of the target device. The disk is now hardware bootable (see Section 2.1). It can be taken to the target hardware and booted.

- b. Use the software BOOT function. The MCR software bootstrap function is described in Appendix D. The software bootstrap boots the file created by phase 1 into memory in the host system. Block 0 of the target system is not modified.

When the target system is bootstrapped there should be no other activity in the system. The host system is overlaid in memory by the target system. The host system remains unaltered.

## STAGE 1 PROCEDURES

Before issuing the BOOT command, ensure that the target system disk is mounted on the unit specified as SY during phase 1. Then issue the following command to the host system.

```
MCR>BOO target disk:filename
```

Notice that the MCR BOOT command can be used to create the bootstrap block on a system disk by including the /WB switch or to perform the bootstrapping from any disk by omitting the /WB switch. The two functions are mutually exclusive.

2. Once the system has been bootstrapped, phase 2 prints the following message.

```
*** SYSTEM GENERATION PHASE 2 ***
```

Phase 2 proceeds to mount the new system disk and then open and read the file SYSBLD.COMD. All commands encountered in SYSBLD.COMD are executed and echoed on the terminal.

### NOTE

Loading handlers or fixing tasks in the partition in which SGN2 is running results in fragmentation of that partition after SGN2 exits.

3. Upon successful completion, phase 2 prints the following message.

```
*** END OF SYSTEM GENERATION PHASE 2 ***
```

4. Set the time and date, load the message output handler, and perform any other similar functions, e.g., loading other handlers.

```
MCR>TIM hh:mm:ss dd-mmm-yy  
MCR>LOA MO  
MCR>LOA LP
```

5. Pseudo devices such as the spooler (SP) and console listing device (CL) should be redirected to physical devices. For example:

```
MCR>RED LP=CL  
MCR>RED SY=SP
```

Any alterations to device characteristics can also be made. For example, the following command sets the line printer as a spooled device:

```
MCR>SET /SP=LP0:
```

## STAGE 1 PROCEDURES

6. If satisfied at this point that the system generation was successful, dismount any mounted devices and perform a save using the MCR SAV command. The following is an example.

```
MCR>DMO SY:  
MCR>SAV
```

Once the system is saved, the following message is printed on the console.

```
nnk IAS V001A    nn is the memory size  
MCR>
```

### NOTE

If at any time before performing the SAV, a reboot of the new system disk is performed, as described in step 1 above, phase 2 of system generation is executed again.

A reboot after a SAV causes the same response as seen after performing the SAV above, i.e., nnk IAS V001A.

7. If the user is satisfied with the system generation and wants the disk to be hardware bootable, the functions of step 1a in this section should be performed if they have not been done already.



## CHAPTER 7

### STAGE 2 PROCEDURES

SGN2, the second phase of Stage 1 system generation installs the current version of the Timesharing Executive. Stage 2 enables the user to produce a new version of the the Timesharing Executive according to the installation's requirements. Stage 2 may be performed at any time, independently of Stage 1.

1. The first step is to edit the file [311,107]SGDATA.MAC, which contains the interactive and batch parameters. These parameters are all described in Chapter 4.

The following commands invoke the IAS Text Editor to edit [311,107]SGDATA.MAC:

```
MCR>SET /UIC=[1,1]
```

```
MCR>EDIT [311,107]SGDATA.MAC
```

When each parameter has been set to the required value, the user then issues the commands described below. Most commands invoke indirect command files, thus simplifying the Stage 2 procedure. Appendix A lists the contents of the various command files.

Note that user-written Command Language Interpreters (CLIs) that link with IASBUF must also be relinked, the old CLIs removed, and new ones installed in steps 3,4, and 5. Either issue individual commands in the appropriate place to do so or edit the command files to include the necessary commands.

2. To assemble and build the IAS timesharing libraries, IASCOM and IASBUF:

```
MCR>SET /UIC=[1,1]
```

```
MCR>MAC @[311.107]MAC107
```

```
MCR>TKB @[11,107]TKB107
```

3. To relink all the tasks that use the libraries IASCOM and IASBUF:

```
MCR>TKB @NEWTSTKB
```

User-written CLIs should be relinked at this point.

## STAGE 2 PROCEDURES

4. The old tasks and libraries must then be removed from the system. The libraries cannot be removed until all tasks which link to them have also been removed. Remove any user-written CLIs before issuing the following command.

```
MCR>REM @OLDTSREM
```

5. To install the new versions of the tasks and libraries:

```
MCR>INS @NEWTSINS
```

The libraries must be installed before the tasks. Do not install any user-written CLIs until the command file above has executed.

6. To save the system with the new Timesharing Executive and installed tasks:

```
MCR>DMO system disk
```

```
MCR>SAV
```

A new version of the Timesharing Executive has now been built and saved into the system.

APPENDIX A  
SYSTEM GENERATION FILES

A.1 STAGE 1 FILES

A.1.1 TT64PA.MAC

```
.SBTTL IAS TT64 TERMINAL HANDLER -- ASSEMBLY PARAMETERS
;
;
; THE FOLLOWING PARAMETERS ARE USED AT ASSEMBLY TIME TO
; DEFINE THE NUMBER AND TYPE OF TERMINAL INTERFACES
; IN THE SYSTEM.
;
; IF THE TERMINAL CONFIGURATION IS CHANGED THE PARAMETERS
; MUST BE CHANGED ACCORDINGLY AND THE HANDLER RE-ASSEMBLED
; AND RE-LINKED.
;
;
; TTYNUM = 1. ; THE TOTAL NUMBER OF TERMINAL INTERFACES
; ; IN THE SYSTEM.
; DHUN = 0 ; THE NUMBER OF DH11 (AND DM11-BB) (16-LINE) UNITS.
; DJUN = 0 ; THE NUMBER OF DJ11 (16-LINE) UNITS.
; DCUN = 0 ; THE NUMBER OF DC11 (SINGLE LINE) UNITS.
; DL11E = 0 ; THE NUMBER OF DL11E (SINGLE LINE) UNITS.
;
; DMEPA = 0 ; THE EXTERNAL PAGE ADDRESS OF THE DM11-BB UNIT
; ; ATTACHED TO THE FIRST DH11; 0 IF NON-EXISTENT.
; ; THE SECOND DH11 IS ASSIGNED THE DM11 AT DMEPA+10 ETC.
; DIAL = 0 ; NON-ZERO IF DIAL-UP LINES NEED SERVICEING.
; PGEN = 0 ; NON-ZERO IF ANY OF THE DL11 TERMINALS NEED PARITY.
;
; SECOND = 3. ; MODEM (DL11E,DM11BB) INSPECTION TIME INTERVAL
; TMUNIT = 2 ; ... .. ... .. RATE
;
;
; THE REMAINDER OF THE IAS TT64 TERMINAL HANDLER
; CODE FOLLOWS IN ISTT64.MAC
;
;
;
```



```
INS IASCOM/LI/ACC=RO/UIC=[1,1]
INS IASBUF/LI/ACC=RO/UIC=[1,1]
INS PDSLIB/LI/ACC=RO/UIC=[1,1]
*DELAY
INS [11,1] ISDT
INS [11,1] DK
INS [11,1] LP
INS [11,1] MO
INS [11,1] ISTU16
INS [11,1] QUE
INS [11,1] ISSPR
INS [11,1] SPR2
INS [11,1] ISOPR
INS [11,1] TKTN
INS [11,1] ABO
INS [11,1] ACT
INS [11,1] ALT
INS [11,1] BOO
INS [11,1] COM
INS [11,1] ISDEV
INS [11,1] ISDMO
INS [11,1] EDI
INS [11,1] FLX
INS [11,1] F11MSG
INS [11,1] INI
INS [11,1] LOA
INS [11,1] LUN
INS [11,1] PURMAC
INS [11,1] OPE
INS [11,1] PAR
INS [11,1] PURPIP
INS [11,1] REA
INS [11,1] ISRED
INS [11,1] REM
INS [11,1] RUN
INS [11,1] SAV
INS [11,1] SET
INS [11,1] SLP
INS [11,1] TAS
INS [11,1] TIM
INS [11,1] UFD
INS [11,1] UNF
INS [11,1] UNL
INS [11,1] USERS
INS [11,1] ACCOUNTS
INS [11,1] VFY
INS [11,1] SCI
INS [11,1] IAS
INS [11,1] IASTART
INS [11,1] TCP
INS [11,1] TSS1
INS [11,1] TSS3
INS [11,1] TSSNUL
INS [11,1] TKB
INS [11,1] DMP
INS [11,1] LBR
*DELAY
REM ...INZ
*DELAY
*DELAY
```

## A.2 STAGE 2 FILE

### A.2.1 SGDATA.MAC

```
.SBTTL EQUATES FOR VARIABLE PARAMETERS
;
; VARIABLE PARAMETERS
;
; THESE PARAMETERS HAVE INITIAL VALUES DEFAULTED TO
; THEM BELOW.
; THOSE PREFIXED 'SG' MAY BE CHANGED DIRECTLY OR
; INDIRECTLY BY USER SPECIFICATION AT SYSGEN TIME.
;
;
;
; SGLUN = 6. ;MAX NO. OF LUNS FOR 1 JOB
;
; SGUVL = 6. ;MAX NUMBER OF DEVICES ALLOCATABLE BY A USER
;
; SGDEV = 8. ;NO. OF NON-TERMINAL DEVICES USED BY TIMESHARING
;
; SGTMM = 6. ;MAX NO. OF TERMINALS
;
; SGATT = 6. ;MAX NO. OF ACTIVE TERMINALS AT ANY TIME
;
; SGXTSK = 6. ;NO. OF TASKS OVER '1 PER TERMINAL' (IE. NON CLI TASKS)
;
; SGXBF = SGATT+SGXTSK+1 ; NO. OF BUFFERS
;
; SGTKSZ = 1024. ; MAX TASK SWAP SIZE(32 WORD BLOCKS)
;
; SGBUF = 80. ;SIZE OF TERMINAL BUFFER (BYTES)
;
; SGCIT = 3. ;NO.OF CLI'S WHICH CAN RUN CONCURRENTLY
;
; SGUTL = 4. ;NO. OF SCHEDULING LEVELS
;
;
; SGSPR = 2500 ; TICKS BETWEEN SCHEDULER PROMOTIONS
;
; SGSTK = 1 ; SIZE OF SYSTEM IDLE TIME
;
; SGCON = 2. ; CONSTANT IN QUANTUM EQUATION
;
; SGTSL = 25. ; TIME SLICE MAXIMUM SIZE
;
; SGAFM = 240 ; ALLOCATION FACTOR MEMORY SIZE
;
; SGAFT = 1 ; ALLOCATION FACTOR TICKS/MEMORY SIZE
;
; SGBSH = 6000. ; TIME BETWEEN BATCH SCHEDULES
;
; SGBQT = SGTSL ; BATCH QUANTUM
; NO WHEN THIS PARAMETER IS NON ZERO A BATCH
; SCHEDULING LEVEL (BOTTOM LEVEL) IS AUTOMATICALLY CREATED
```

```
;
SGBTSP = 0 ; MAX NON-SWAPPABLE SPACE (MOD 32 WORDS)
;
SGT1PR = 100. ; TSS1 PRIORITY
;
SGPIPR = SGT1PR+1 ; TCP PRIORITY
;
;
SGSPF = 2 ; NUMBER OF SWAP FILES
;
SGLBSZ = 4. ; LOGICAL SWAP BLOCK SIZE (NUMBER OF 256. WORD BLOCKS)
;
SGFLSZ = 500. ; MAX LOGICAL BLOCKS PER SWAP FILE
;
```





```

.IIF      B <BLKCNV>      .WORD      0
.IIF      NB <VSIZH>      .WORD      VSIZH
.IIF      B <VSIZH>      .WORD      0
.IIF      NB <VSIZL>      .WORD      VSIZL
.IIF      B <VSIZL>      .WORD      0

;
LNG       = .-Z           ;LENGTH OF CHARACTERISTICS TABLE
.ENDM

.PAGE

.SBTTL   DVSCH -- STANDARD SUPPORTED DEVICE CHARACTERISTICS

;
;
;
DCHST:
DEVCH    LA30P,7,0,1,72.
DEVCH    LA30S,7,20000,1,72.
DEVCH    LA36,7,21000,1,132.
DEVCH    VT05,7,50010,0,80.
DEVCH    KSR33,7,4000,0,72.
DEVCH    PTR,40,0,0,1000
DEVCH    PTP,40,0,0,1000
DEVCH    RK03,140010,503,0,512.,F11,RK05,RK05DA,,4800.
DEVCH    LP11A,3,0,0,80.
DEVCH    LP11B,3,0,0,132.
DEVCH    DT11,140010,0,0,512.,F11,,,,,578.
DEVCH    AFC11,0,1023.,0,0
DEVCH    AD01,0,100,0,0
DEVCH    LPS11,0,16.,0,0
DEVCH    UDC11,0,0,0,0
DEVCH    RK05,140010,503,0,512.,F11,RK05,RK05DA,,4800.
DEVCH    RP02,140010,40000,0,512.,F11,RP03,RP23AB,,40000.
DEVCH    RP03A,140010,100000,0,512.,F11,RP03,RP23AB,1,34200
DEVCH    RP03B,140010,140000,0,512.,F11,RP03,RP23AB,1,34200
DEVCH    KSR35,7,2000,0,72.
DEVCH    RF1,140010,0,0,512.,F11,RF11,RFDSK,,1024.
DEVCH    RF2,140010,0,0,512.,F11,RF11,RFDSK,,2048.
DEVCH    RF3,140010,0,0,512.,F11,RF11,RFDSK,,3072.
DEVCH    RF4,140010,0,0,512.,F11,RF11,RFDSK,,4096.
DEVCH    RF5,140010,0,0,512.,F11,RF11,RFDSK,,5120.
DEVCH    RF6,140010,0,0,512.,F11,RF11,RFDSK,,6144.
DEVCH    RF7,140010,0,0,512.,F11,RF11,RFDSK,,7168.
DEVCH    RF8,140010,0,0,512.,F11,RF11,RFDSK,,8192.
DEVCH    CR11,1,0,0,80.
DEVCH    RP04,140010,160000,0,512.,F11,RP04,RP04DA,2,117426
DEVCH    RS03,140010,100000,0,512.,F11,RS04,RS03DA,,1024.
DEVCH    RS04,140010,140000,0,512.,F11,RS04,RS04DA,,2048.
DEVCH    TU10,140070,0,0,512.,MTA
DEVCH    TU16,140070,100000,0,512.,MTA
DEVCH    TAl1,100070,0,0,128.
DEVCH    ,0,0,0,0

DCHEN:

```

## APPENDIX C

### SYSTEM GENERATION FOR TERMINALS ON DH11 OR DJ11 LINES

Normally during system generation, when a device is defined using the DEV directive, the type parameter in the DEV directive supplies the information needed by the system to fill in the four device characteristics words of the PUD. However, in certain cases, the user must supply the information for the characteristics words in a different format. Such is the case for terminals connected to DH11 or DJ11 lines. The format of the DEV directive used to provide this information follows.

```
DEV=dev,<char1,char2,char3,char4>,trap,pri,ext-page[,devacp]
```

All parameters, except those for the characteristics words, are identical with those described in Chapter 3 of this manual.

char1 = characteristics word 1. It has the following values.

Word	Bit	Meaning If Set (Bit=1)
1	0	Indicates a record-oriented device, e.g., card reader.
	1	Indicates a carriage control device, e.g., line printer.
	2	Indicates a terminal device, e.g., LA30.
	3	Indicates a multiple directory device.
	4	Indicates a single directory device.
	5	Indicates a sequential device, e.g., magnetic tape.
	6	Indicates an output spooled device.

char2 = characteristics word 2. It contains device-specific flags used by the handler. See Table C-1.

char3 = characteristics word 3. It contains additional flags used by the handler. See Table C-2.

char4 = characteristics word 4. It contains the maximum block (buffer) size for the device as an octal number of bytes.

TERMINALS ON DH11 OR DJ11 LINES

C.1 PUD CHARACTERISTICS WORDS

For DL, lines, the type parameter may be used in the DEV directive instead of explicitly stating the individual characteristics words.

Table C-4 contains examples of the four characteristics words for some terminals devices connected to DH11 lines.

Table C-5 contains the four characteristics words for all supported terminals device connected to DJ11 lines.

<u>Table C-1</u> <u>Characteristics Word 2 for Terminal Handler</u>	
<u>Bit</u>	<u>Meaning if Set</u>
15 - 13	Carriage control type 0 = KSR33, KSR35, LA30P 1 = LA30S 2 = VT05
12	Scope (enables VT05 to run at speeds up to 2400 baud)
11	ALT code = 175
10	ALT code = 176
9	Lower case enabled
8	Parity exists
7	Odd parity
6	DH11 Interface for this terminal
5	Hardware form feed
4	Hardware vertical tab
3	Hardware horizontal tab
2	Local echo
1	No printer
0	No keyboard

TERMINALS ON DH11 OR DJ11 LINES

Table C-2  
Characteristics Word 3 for Terminal Handler

<u>Bit</u>	<u>Meaning if Set</u>
15	DL11E interface for this terminal
14	Indicates DM11-BB Controller
13 - 10	Transmitter speed See Table C-3.
9 - 6	Receiver speed
5	Must be 0
4	DJ11 Interface for this terminal
3	Dial-up line
2	DC11 interface for this terminal
1	EIA interface (for DH only)
0	Margin (LA30 only)

TERMINALS ON DH11 OR DJ11 LINES

<u>Table C-3</u> <u>Receiver and Transmitter Speeds</u>					
	<u>Bit</u>				<u>Speed</u>
<u>Transmitter</u>	13	12	11	10	
<u>Receiver</u>	9	8	7	6	
	0	0	0	0	Zero Baud
	0	0	0	1	50 Baud
	0	0	1	0	75 Baud
	0	0	1	1	110 Baud
	0	1	0	0	134.5 Baud
	0	1	0	1	150 Baud
	0	1	1	0	200 Baud
	0	1	1	1	300 Baud
	1	0	0	0	600 Baud
	1	0	0	1	1200 Baud
	1	0	1	0	1800 Baud
	1	0	1	1	2400 Baud
	1	1	0	0	4800 Baud
	1	1	0	1	9600 Baud

<u>Table C-4</u> <u>Characteristics Words for Devices on DH11 Lines</u>	
<u>Device</u>	<u>Characteristics Words</u>
VT05 on dial-up line (110 baud send/ receive with DM11-BB)	<7,50110,46312,120>
VT05 on local DH line (150 receive/ 2400 send) where the DM11-BB is attached to DH11 unit	<7,50110,66500,120>
VT05 on local DH line (110 baud) 2400 send) with no DM11-BB	<7,50110,26500,120>
KSR33 on local DH line (110 baud) with DM11-BB	<7,6100,46300,110>
LA30S on local DH line (300 baud) with DM11-BB	<7,20100,56701,120>
NOTE	
If any DH lines are connected to the DM11-BB, all lines for the DH unit must be connected.	

# TERMINALS ON DH11 OR DJ11 LINES

<u>Table C-5</u> <u>Characteristics Words for Devices on DJ Lines</u>	
<u>Device</u>	<u>Characteristics Word</u>
VT05 on DJ line	<7,50010,20,120>
KSR33 or KSR35 on DJ line	<7,6000,20,110>
LA30S on DJ line	<7,20000,21,120>
LA30P on DJ line	<7,0,21,120>

## C.2 PARTITION

The size of the TTY partition must be large enough for the size of the handler. When the TTY partition is enlarged, some other partition (e.g. GEN) must be decreased accordingly.

## C.3 LOGICAL NAMES AND NUMBERS

IAS terminals have device names in the format TT0,TT1,..., TTn; n is the octal unit number. A DH11 or DJ11 hardware unit multiplexes 16 lines. The lines are distinguished by subline numbers.

At initialization, the Teletype handler must establish a correspondence between each hardware number and the corresponding PUD unit number. The handler also must allow for the possible existence of gaps in the hardware numbers for the DH11.

### C.3.1 EIA and 20 ma Lines (DH11 only)

Generally, four line adaptor units are connected to a DH11. Each of these units performs line adaption for four terminal lines.

### C.3.2 How to Assign Names

The terminal unit numbers are assigned to DH11 subline numbers in order. The only exception is that there can be a gap between the EIA lines, which must come first, and the 20 ma lines. If a gap occurs, the DH11 subline number is assumed to jump to the next group of four lines. The following is an example.

IF: TT7 is the first Teletype PUD to indicate that it is connected to a DH11 unit. The PUD characteristics words also indicate that it is an EIA line.

AND: TT10 is connected to the same DH11 unit, but the PUD characteristics words indicate that the line adaptor is a 20 ma connection.

TERMINALS ON DH11 OR DJ11 LINES

THEN: TT7 is assigned to DH subline number 0 and TT10 is assigned to DH subline number 4.

## APPENDIX D

### MCR COMMANDS

This appendix describes some MCR commands used during system generation. The commands described are:

- BOOT
- DISMOUNT
- FIX-IN-MEMORY
- INITIALIZE VOLUME
- INSTALL TASK
- LIST COMMON BLOCKS
- LOAD
- MOUNT
- PARTITIONS
- REDIRECT
- SAVE
- SET
- TIME
- UNFIX
- UNLOAD
- USER FILE DIRECTORY

**BOO**D.1 BOOTSTRAP COMMAND (BOO)FUNCTION

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

1. Stop the present system and bootstrap another system from any system device.
2. Copy a bootstrap block on to block 0 of the specified device from a specified IAS system image file on the same device.

FORMAT

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

## Format 1

BOO[T] CR

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

BOO>

The user must now enter another carriage return.

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

## Format 2

BOO[T] filespec

## MCR COMMANDS

where:

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

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

Format 3

BOO[T] /WB

where:

/WB is the write bootstrap switch.

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

Format

BOO[T] filespec/WB

where:

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

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

MCR COMMANDS

<u>Table D-1</u> <u>Defaults In BOO File Specifications</u>	
<u>Specification</u>	<u>Default</u>
dev:	SY:
ufd	The UFD which corresponds to the current UIC under which the user is running
filename	IAS
.extension	.SAV
;version	Highest version

TECHNICAL NOTES

1. All devices, except the one from which the bootstrap operation is to be accomplished, should be dismounted.
2. When a re-bootstrap is to take place, no tasks should be running on the current system.
3. The device from which a task is installed is recorded in that task's STD entry. Therefore, after bootstrapping a new system, any installed tasks must be present at the same place on their original devices; e.g. ,a system pack cannot be moved from DK0 to DK1 and then bootstrapped using the BOOT command.
4. Only as much memory as was specified at system generation time will be loaded during the bootstrap operation.
5. The file specified to BOO must have been created by system generation and must not have been removed or copied after creation.

EXAMPLES:

1. MCR>BOO CR

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

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

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

## MCR COMMANDS

### 3. MCR>BOO /WB

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

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

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

**DMO**D.2 DISMOUNT VOLUME COMMAND (DMO)FUNCTION

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

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

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

FORMAT

```
DMO dev:[volumelabel] [/UIC=[uic]] [/LOCK]
```

where:

dev:	is the device specification for the device on which the volume is mounted.
volumelabel	is the label of the volume being dismounted. If specified, the value entered is compared with the corresponding value in the VCB. If they are not the same, the volume is not dismounted.
/UIC=[uic]	is the Volume UIC. Like the volume label, it is compared to its corresponding entry in the VCB. If unequal, the volume is not dismounted.
/LOCK	If this option is specified, the volume is marked for dismount regardless of whether files are currently accessed on it. Further accesses are then disallowed and the volume is actually dismounted only when all current file accesses are terminated.

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

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

```
DMO -- DKØ: ** DISMOUNT COMPLETE **
```

EXAMPLES

1. MCR>DMO DK1:

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

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

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

**FIX**D.3 FIX-IN-MEMORY COMMAND (FIX)FUNCTION

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

FORMAT

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

where:

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

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

EXAMPLES

1. MCR>FIX XKE
2. MCR>FIX JAG,240Z

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

**D.4 INITIALIZE VOLUME COMMAND (INI)**FUNCTION

## WARNING

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

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

FORMAT

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

where:

dev: is the device specification of any file structured media.

volumelabel is the label being assigned to the volume.

## NOTE

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

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

/UIC=[group number, member number]

Default

/UIC=[1,1]

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

This keyword allows the user to establish volume access privileges. Each entry consists of from

## MCR COMMANDS

one to four letters which have the following meanings:

- R - for READ access
- W - for WRITE access
- E - for EXTEND access
- D - for DELETE access

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

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

Example

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

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

Default

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

### NOTES

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

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

Example

```
/MXF=100
```

## MCR COMMANDS

Default

One fourth the number of blocks on the volume.

/EXT=default file extension size in blocks.

Default

/EXT=5

/FPRO=[default file protection]

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

Default

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

/CHA=[characteristic words]

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

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

DCF - device control functions permitted; Read/Write logical, for example.

Example

/CHA=[ATCH,DCF]

Default

No ATCH and no DCF

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

Default

/INF=16

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

Default

/WIN=7

/LRU=number of directories to keep pre-accessed while volume is in use. For optimized directory access this value should be

## MCR COMMANDS

slightly higher than the expected number of concurrent users of the volume.

Default

/LRU=3

/INDX=Option - Index file position.

Options are:

BEG - beginning of volume

MID - middle of volume

END - end of volume

BLK:number - logical block number

Example

/INDX=BLK:100

Default

/INDX=MID

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

Options are:

AUTO - use bad blocks data left on volume by BAD command.

MAN - enter bad blocks.

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

Example

/BAD=[AUTO]

or

/BAD=[AUTO,MAN]

Default

None

D.4.1 MAN Option for /BAD keyword

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

```
INI>BAD=
```

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

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

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

where:

- u is the logical block number, in octal, of the initial bad block in the group.
- n is the number, in octal, of consecutive blocks contained in the group. If omitted, a value of one is assumed.

## NOTE

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

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

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

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

TECHNICAL NOTES

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

1. INI must be initiated as follows:

```
MCR>INI CR
```

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

```
INI>
```

## MCR COMMANDS

2. At this point, multiple lines can be typed in to INI. Each line except the last must contain a hyphen (-) as the last character entered. When INI recognizes the hyphen, it assumes that the command line is incomplete, and reissues the INI> prompt for more parameters. When the last line is entered (nohyphens), INI begins processing the command.

### EXAMPLES

1. MCR>INI DB1:TESTPACK/CHA=[ATCH,DCF]/BAD=[AUTO]

In this example, a disk volume on DB1: is initialized with the following characteristics:

Volume label	- TESTPACK
UIC	- [1,1]
Volume Protection	- System = RWED Owner = RWED Group = RWED World = RWED
Maximum number of files	- 42949
Default file extension	- 5
File protection	- System = RWED Owner = RWED Group = RWE World = R
Characteristic word	- ATCH and DCF
Index file position	- MID
Bad blocks	- Automatically accounted for.
Number of headers in index file	- 16
Window size	- 7
Number of pre-accessed directories	- 3

MCR COMMANDS

2. MCR>INI DB1:TESTPACK/PRO=[RWED,RWED,RW,R]/CHA=[ATCH,DCF]-  
/BAD=[MAN]

INI>BAD=6.,10.

INI>BAD=12.

INI>BAD=

In the above example, a disk volume on DB1: is initialized with the following characteristics:

Volume label	- TESTPACK
UIC	- [1,1]
Volume protection	- System = RWED Owner = RWED Group = RW World = R
Maximum number of files	- 42949.
Default file extension	- 5
File protection	- System = RWED Owner = RWED Group = RWE World = R
Characteristic word	- ATCH and DCF
Index file position	- MID
Bad blocks	- 6 through 17, and 12
Number of headers in index file	- 16
Window size	- 7
Number of pre-accessed directories	- 3

**INS**D.5 INSTALL TASK COMMAND (INS)FUNCTION

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

FORMAT

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

or

```
INS[TALL] @indirect
```

where:

`filespec` is the file specification for the image (task or shareable global area) being installed. Table D-2 contains a list of defaults in `INS` file specifications.

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

`/PAR=partition name`

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

Example

```
/PAR=GEN
```

Default

Value specified at system generation.

`/PRI=priority-number (decimal)`

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

Example

```
/PRI=200
```

## MCR COMMANDS

Default

/PRI=50

/TASK=taskname

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

Example

/TASK=RICK

/POOL=pool-limit (decimal)

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

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

Example

/POOL=100

Default

/POOL=40 (established by task builder)

/UIC=[uic]

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

Example

/UIC=[11,11]

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

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

Example

/ACC=RO

Default

/ACC=NA

/LI and /CM

These options are used to specify that the entity being installed is either a library (/LI) or a common area (/CM).

Default

/CM

<u>Table D-2</u> <u>Defaults in INS File Specifications</u>	
<u>Field</u>	<u>Default</u>
dev:	If omitted in the first or only file specification, SY: is used.  If omitted in the second through n file specifications, the device specified or defaulted from the previous file specification is used.
[ufd]	If omitted in a file specification, the UFD that corresponds to the UIC under which the user is running.
filename	Must be specified.
.extension	.TSK
;ver	The highest version for the file.

EXAMPLES:

1. MCR>INS SCAN

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

2. MCR>INS DB1:[11,2]SCAN;3/PRI=103,RICK/TASK=HELP

In this example, the tasks to be installed reside in directory file [11,2] on DB1:. Version 3 of task image file SCAN.TSK, and the latest version of task image file RICK.TSK are selected. The task images contained in these files are installed into the system, with the following modifications being made to them. Task image SCAN will have its priority changed to 103, and task image RICK will have its name changed to "HELP".

**D.6 LIST COMMON BLOCKS COMMAND (COM)****FUNCTION**

The COMMON BLOCKS (COM) command lists on the user terminal a description of each installed shareable global area (i.e., dynamic libraries and common blocks).

**FORMAT**

COM[MON BLOCKS]

**EXAMPLE**

MCR>COMMON BLOCKS

SYSRES 200400 016200 1,1 RO LIB PI 02/04/75

Figure D-1 describes the information listed by the COMMON BLOCKS command.

	1	2	3	4	5	6
	SYSRES	200400	016200	1,1	RO	LIB PI 02/04/75
1.	Common Blocks name, Common Block base and Common Block size with numeric values in octal notation					
2.	UIC					
3.	Non-owner access (NA = no access, RO = read only, RW = read/write)					
4.	Library or Common Block indicator (i.e., LIB or COM)					
5.	Position Independent Code indicator (i.e. PI or blank)					
6.	Date of Library or Common Block creation in the form mm/dd/yy					

Figure D-1  
Sample COMMON BLOCK Listing and  
Description of Listed Items

# LOA

## D.7 LOAD COMMAND (LOA)

### FUNCTION

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

### FORMAT

LOA[D] handler-name[:]

where

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

### EXAMPLE

MCR>LOAD LP

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

**MOU****D.8 MOUNT COMMAND (MOU)****FUNCTION**

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

**FORMAT**

The format listed below is the general format for the MOUNT command. MOUNT for multivolume magnetic tape is somewhat different in format, and is discussed in section D.4.1.

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

where:

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

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

/keyword(s)

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

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

\* Applies to disk and DECtape only

= Overrides the corresponding option specified at INITVOL

+ Applies to magnetic tape only

/CHA=[characteristic word]

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

## MCR COMMANDS

\*+ FOR -a non-Files-11 structured volume (FOREIGN). When this parameter is specified DCF is automatically assumed.

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

DCF -device control FUNCTIONS permitted; removes restrictions.

### NOTE

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

### Example

```
/CHA=[FOR,ATCH]
```

\* /UNL

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

+ /DENS=magnetic tape density

Possible parameters are:

800 for 800 BPI

1600 for 1600 BPI

### Example

```
/DENS=800
```

```
/ACP=task name
```

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

### Example

```
/ACP=DSKFI
```

\*= /EXT=default file extend increment in blocks

### Example

```
/EXT=5
```

\*= /FPRO=[default file protection]

This option allows the user to change the default file protection assigned to new files created on the volume.

## MCR COMMANDS

Each entry consists of from one to four letters which have the following meanings.

- R - for READ access
- W - for WRITE access
- E - for EXTEND access
- D - for DELETE access

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

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

Example

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

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

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

Example

```
/LRU=3
```

```
/OVR
```

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

+ /OVRFSID

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

+ /OVREXP

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

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

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

- R - for READ access
- W - for WRITE access

## MCR COMMANDS

E - for EXTEND access  
D - for DELETE access

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

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

Example

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

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

```
/UIC=[uic]
```

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

Example

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

### D.8.1 Format For Mounting Multivolume Magnetic Tape

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

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

where:

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

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

#### NOTE

A separate unit number need not be specified for each volume in the set.

## MCR COMMANDS

The file processor processes volumes sequentially down the list of specified units, until the last unit is reached. If more volumes are to be processed, a mount request, for the next sequential volume to be mounted is issued for one of the units listed.

/keyword(s)

See keywords provided for the general class of MOUNT command.

### TECHNICAL NOTES

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

1. Initiate MOU as follows:

```
MCR>MOU
```

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

```
MOU> CR
```

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

### EXAMPLES

1. MCR>MOU[NT] DK1:SYS004

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

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

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

# PAR

## D.9 PARTITIONS COMMAND (PAR)

### FUNCTION

The PARTITIONS command (PAR) lists a description of each memory partition in the system.

### FORMAT

PAR[TITIONS]

### EXAMPLE

MCR>PAR

SYDISK	071400	003400	UC
SYS	075000	015000	SC
GEN	112000	620000	TC
TTY	732000	011100	UC

### NOTE

"UC" means "user-controlled" partition;  
 "SC" means "system-controlled"  
 partition; "TC" means "timesharing".

### TECHNICAL NOTES

1. The PARTITIONS command lists the:
  - a. Partition Name
  - b. Partition Base (octal address)
  - c. Partition Size (octal)
  - d. Control Code UC, SC or TC.
2. The PARTITIONS command can be used to check that all available memory has been allocated. The highest memory address allocated can be determined by adding the Base and Size of the last partition in the list. In the example above, the highest memory address for the TTY partition would be 732000 + 011100 which equals 743100. Therefore, 743100 is the highest address used. Memory higher than that address should be allocated by increasing the size of a partition (normally GEN). If the system had 124K (760000 octal bytes), then 14700 (that is, 760000-743100) bytes will not be allocated. Therefore, one partition should be increased by 14700 bytes to use all available memory.

D.10 REDIRECT COMMAND (RED)FUNCTION

The REDIRECT command (RED) allows the operator to redirect all I/O requests made for a pseudo device to the physical device unit with which it is associated or allows a spooled device to be redirected to another spooled device of the same type.

FORMAT

RED[IRECT] new-dev:=old-dev:

where

new-dev is the new device unit symbol, followed by an optional unit number (zero is assumed).  
 = is notation signifying the redirect action.  
 old-dev is the old device unit symbol, followed by an optional unit number (zero is assumed).

EXAMPLES

1. MCR>REDIRECT LP0:=CL:

Redirect all I/O requests for CL to device TT3; where both the devices are set output spooled

2. MCR>RED LP1:=LP0:

Redirect all I/O requests from device LP0 to device LP1.

TECHNICAL NOTES

1. The RED command does not redirect any I/O already in the queue. Previous I/O requests are not transferred.
2. If, through a sequence of Redirect commands, the user establishes a redirect chain which returns to old-dev, the following message is issued, and the RED command is rejected:

RED -- CIRCULAR REDIRECT CHAIN

**SAV****D.11 SAVE COMMAND (SAVE)****FUNCTION****WARNING**

The SAVE command should be executed only when no tasks are running.

The SAVE command is used 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.

**FORMAT**

SAV[E] [/switch]

where:

/switch is the following optional switch:

/MOU=dev:[dev:...:dev]

This switch is used to allow the system to be saved with the specified devices mounted. See Technical Note 3.

**NOTE**

If a system still has volumes mounted on it and the /MOU switch to the SAVE command has not been specified, the save will not occur.

**EXAMPLE**

1. MCR>SAVE

In this example, the current status of the system is saved on the disk from which it was originally bootstrapped.

2. MCR>SAVE /MOU:DF0:

In this example, the system is saved with DF0: mounted.

**TECHNICAL NOTES**

1. The SAVE command should be requested only when the system is inactive. This command is intended to provide the user with development stages rather than crash recoveries.

## MCR COMMANDS

2. The SAVE command will attempt to record (on the system device) all memory specified at system generation time. If more memory exists than was declared at system generation, only the declared memory will be saved. If, however, less memory exists than was declared at system generation, SAVE will loop and not complete its operation successfully.
3. SAVE will not permit the system to be saved with volumes mounted. When specifying the /MOU switch, the user must consider the following information about Files-11 volumes:
  - a. When a volume is mounted, volume control data is established in memory to reflect the volume's current file status. This data is updated with every file operation.
  - b. When a volume is dismounted, the volume's control data is reset.
  - c. If the user selects to save the system with volumes mounted the volume control data for each mounted volume is saved, reflecting the current status of the volume. It is imperative that no file activity occurs on the volume between the time the system was saved, and the next bootstrapping of the system. Otherwise, the integrity of the volume will be destroyed.
  - d. When the system is re-bootstrapped, the status of the volume must be exactly the same as it was when the system was saved, or the integrity of the volume will be destroyed on the very first file operation.

### NOTE

After the system is re-bootstrapped, and before any file activity has occurred, the user can execute the following commands to ensure the integrity of any volume:

MCR>DMO dev:

MCR>MOU dev:

This will reset the volume's control data to reflect the current volume status.

**SET**D.12 SET COMMAND (SET)FUNCTION

The SET command provides a facility for changing the MCR timeout, the terminal's characteristics and for setting or clearing output spooling on a device.

FORMAT

SET [/keyword(s)]

where

/keyword is one or more optional keywords which, when specified, alter or set the default values they represent. The following keyword options are provided:

/UIC=[group,member]

This option allows the user to change the default UIC to the one specified to the right of the equal sign (=).

Example

/UIC=[200,200]

/TMO=nnnnn

This option allows the user to change the MCR timeout value. The number nnnnn is a decimal value (any number up to 32767) representing the number of seconds MCR is to wait for a command before timing out.

Example

/TMO=90

/LA30S=dev:[dev:...:dev:]

This option defines the specified device(s) as having LA30S characteristics.

Example

/LA30S=TT1:TT2:

## MCR COMMANDS

`/LA30P=dev:[dev:...:dev:]`

This option defines the specified devices as having LA30P characteristics.

Example

`/LA30P=TT1:TT2:`

`/KSR33=dev:[dev:...:dev:]`

This option defines the specified device(s) as having KSR33 characteristics.

`/SP=dev:[dev:.....:dev:]`

This option defines the specified device(s) as being available to the output despooler for the spooling of spooled output files.

Example

`/SP=LP:TT1:`

`/-SP=dev:[dev:...:dev:]`

This option defines the specified device(s) as no longer being available to the output despooler for the spooling of output files.

Example

`/-SP=TT1:TT2:`

### EXAMPLES

1. `MCR>SET /SP=LP0:`

Sets the line printer LP0: as an output spooled device.

2. `MCR>SET /LA30S=TT1:TT2:/SP=TT3:`

In this example, the following actions are performed:

- a. TT1:, TT2:, and TT3: are defined as having LA30S characteristics.
- b. TT3: is made available to the output despooler for the spooling of output files.

# TIM

## D.13 TIME COMMAND (TIM)

### FUNCTION

The TIME command (TIM) allows the operator to list the time and date, or to alter the time and date values in the system clock calendar.

### FORMAT

TIM[E] [time] [date]

where

time            is specified in the format hh:mm:ss

date            is specified in the format dd-mm-yy

### NOTE

If the TIME command is executed without parameters, the current system time and date will be listed. Also, if either or both of the parameters (time or date) are specified, the corresponding field in the system will be altered to the value specified by the parameter.

### EXAMPLE

1. MCR>TIM 9:10:20 26-JUN-75

in this example the system time and date are changed to 9:10:20 and 26-JUN-75 respectively.

**D.14 UNFIX COMMAND (UNF)****FUNCTION**

The UNFIX command (UNF) frees "fixed" tasks from memory.

**FORMAT**

```
UNF[IX] taskname[/TI=dev][,taskname[/TI=dev],...]
```

where

taskname is the name of the task being unfixed.

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

**EXAMPLES**

1. MCR>UNFIX XKE

Unfix task XKE, freeing it from memory.

2. MCR>UNFIX 240Z, NK111

Unfix task 240Z and task NK111.

# UNL

## D.15 UNLOAD COMMAND (UNL)

### FUNCTION

The UNLOAD command (UNL) causes the indicated device handler to exit, releasing its memory and causing the device to become inaccessible.

### FORMAT

UNL[OAD] handler-name[:]

### EXAMPLE

1. MCR>UNLOAD LP

In the example above the request unloads the line printer handler task.

**D.16 USER FILE DIRECTORY COMMAND (UFD)****FUNCTION**

The User File Directory command (UFD) creates a user file directory (UFD) on the specified volume and enters its file name into the master file directory (MFD). The UFD command accepts the [group,owner] number as both the UFD name and the file owners' UIC.

**FORMAT**

UFD device:[uic][/switch]

where:

device: specifies the device that contains the volume to be acted upon.

[uic] is the UIC of the UFD being created.

**NOTE**

The brackets which enclose the UIC are a required part of the parameter.

/switch is one or more of the optional UFD switches described in Table D-3.

**EXAMPLE**

**MCR>UFD DK1:[1,1]/ALLOC=100.**

In the example above, the command created the UFD on disk device DK1 with the UIC [1,1] as the directory name; space is allocated for 100 directory entries.

Table D-3  
UFD Switches

<u>Switch</u>	<u>Description</u>
/PRO	<p>This switch allows the owner of the directory to selectively permit access to his file. This switch is specified in the following format:</p> <p style="text-align: center;">/PRO=[system,owner,group,world]</p> <p style="text-align: center;">NOTE</p> <p>If the /PRO switch is specified, the default file protection for the volume is used.</p>
/ALLOC	<p>This switch allows the owner of the directory to pre-allocate space for a specified number of directory entries. This switch is specified in the following format:</p> <p style="text-align: center;">/ALLOC=number-of-entries</p> <p style="text-align: center;">NOTE</p> <p>The number specified is assumed to be octal; the number may be specified with a trailing decimal point to represent a decimal value. The default is 32 octal.</p>

APPENDIX E  
ERROR MESSAGES

During system generation, errors detected cause a message to be printed on the console. Phase 1 messages are preceded by a SGN1; phase 2 messages are preceded by SGN2. System generation produces three types of error messages.

1. Diagnostic messages -- These messages are informative and do not result in termination of system generation. The user should not force termination if a diagnostic message is printed. Diagnostic messages are prefixed as follows.

SGn -- \*DIAG\*

2. Diagnostic message dependent on console input -- These message are nonfatal if console intervention is possible. If input is from an indirect file, the error is declared fatal.
3. Fatal error messages -- These errors are not recoverable. System generation terminates the current attempt to build the system. Fatal error messages are prefixed as follows.

SGn -- \*FATAL\*

Error messages in this appendix are divided into one section for phase 1 messages and another for phase 2 messages. Within each section, messages are presented in alphabetic order.

1. PHASE 1 ERROR MESSAGES

If an error occurs during phase 1 and the message output handler is not loaded, a number is printed on the console rather than the error message. In the following section, the number associated with each message is provided to the right of the message.

CANNOT REQUEST ...INV

40

Failure of the REQUEST directive or corruption of the communication data sent by SGN1 to ...INV (the special version of INSTALL) has occurred. If the latter is true, remove and reinstall ...INV and then try rerunning phase 1.

ERROR MESSAGES

DUPLICATE NAME	3
<p>A device name, partition name, or shared region name has appeared more than once.</p>	
ERROR IN READING TASK IMAGE	27
<p>An attempt to read a task has failed.</p>	
ERROR IN STD TABLE CONSTRUCTION	20
<p>SGN1 has failed internally.</p>	
ERROR ON COMMAND INPUT	9
<p>A hardware failure has occurred on attempting to read command input</p>	
FILE POSITION ERROR	0
<p>An internal system generation failure.</p>	
I-O ERROR ON FILE filename	10
<p>I/O error has occurred on reading or writing the named file.</p>	
ILLEGAL CHECKPOINT DEVICE	28
<p>The checkpoint device was not defined by a DEV=directive.</p>	
ILLEGAL ERROR -- SEVERITY number	5
<p>SGN1 has failed internally.</p>	
ILLEGAL 'SY' INPUT	6
<p>This message can be caused by any one of the following conditions.</p>	
<p>Undefined device (no DEV command associated with it)</p>	
<p>Invalid mnemonic</p>	
<p>Device for which no bootstrap program exists (see Section 5.4)</p>	
ILLEGAL MULTI-PARAMETER SETS	32
<p>Multiple parameter sets have occurred in a directive that permits only one parameter set.</p>	
IMPROPER "@" FILE SPECIFICATION	8
<p>File syntax specification incorrect.</p>	

ERROR MESSAGES

INSUFFICIENT PARAMETERS	33
All the parameters required by the directive have not been submitted.	
INVALID ACP TASKNAME	39
The specified name of the file primitive tasks for this device was not nnnACP; i.e., the last three characters are erroneous.	
INVALID DEVICE TYPE <type>	16
Device type not recognizable. The device type is not defined internally in system generation.	
INVALID KEYWORD IDENTIFIER	28
Keyword in a directive cannot be recognized.	
INVALID PARTITION NAME	35
An attempt has been made to install a task in a nonexistent partition.	
INVALID TARGET DEVICE SPECIFICATION	34
A syntax error was made in specifying the target device and filename.	
INVALID TASK NAME	22
Bad data was found in the header of a task INV has installed. Possibly a task has become corrupted. Start phase 1 again.	
MARK TIME OR WAIT FOR DIRECTIVE FAILURE	19
SGN1 failed to issue one of these directives successfully.	
NO DYNAMIC STORAGE AVAILABLE	4
No more nodes are available. Indicates nested generations may be required. Thus, the user must generate successively larger systems until he reaches his target system. Alternatively, some running on or installed tasks should be removed; then try again.	
NON-EXISTENT "@" FILE SPECIFIED	7
The indirect file specified cannot be found.	
NO SPACE FOR POOL	12
SCOM size insufficient to provide for any pool.	

ERROR MESSAGES

NO TELETYPE HANDLER (TT....) INSTALLED	18
<p>After all the installs for phase 1 were processed, SGN1 found that no task with the name TT.... was installed. Insert the directive to install TT.... in the phase 1 command file. This message is a warning; the system is operable.</p>	
NO TELETYPE ZERO DEFINED CI, CO, AND CL NOT REDIRECTED	17
<p>TT0 was not assigned. CI, CO, and CL have no assignments and should be redirected when the target system is initiated.</p>	
OPEN FAILURE ON FILE filename	11
<p>File cannot be opened. The failure has occurred in attempting to open EXEC.TSK, one of the bootstrap .TSK or .STB files, or the system image output file. Usually either the file does not exist, or no room exists on the disk to create the output file.</p>	
OPTION LINE SYNTAX ERROR	30
<p>A syntax error exists in an directive line.</p>	
OUTPUT I-O ERROR ON FILE filename	25
<p>An I/O error has occurred when writing the save file.</p>	
OVERLAP IN REAL ADDRESS SPACE	21
<p>The base specifications and size specifications result in a space allocation conflict.</p>	
REAL MEMORY SIZE EXCEEDED	2
<p>After summing memory requests, those requests are found to exceed the amount specified by the PDP11 directive.</p>	
STD ENTRY NOT FOUND FOR A TASK AFTER INSTALL	36
<p>INSTALL has failed to create an STD entry. INSTALL may issue a message in addition to the one issued by SGN1. Alternatively, INV may not have run to completion.</p>	
SYMBOL NOT FOUND symbol	1
<p>A symbol needed by SGN1 cannot be found in the appropriate bootstrap .STB file. The file either has not been built or has been built incorrectly.</p>	
SYSGEN PHASE 2 NOT INSTALLED	26
<p>After all phase 1 installs were processed, SGN1 determined that SGN2 was not installed.</p>	

ERROR MESSAGES

SYSTEM DISK HANDLER NOT ALLOWED IN 'T' CONTROLLED PARTITION	41
The system disk handler has been installed into a timesharing controlled partition. It must be put into a User of System controlled partition.	
SYSTEM DISK HANDLER NOT INSTALLED	24
No task with the name xy.... was installed. xy is the device specified in the SY directive, which was valid. See Section 3.4.2.1.	
SYSTEM DISK NOT DEFINED, 'SY' NOT REDIRECTED	23
SY has not been defined explicitly. Therefore, SGN1 is attempting, by default, to redirect SY to DK0. However, no such device has been defined by a DEV directive; SY cannot be redirected.	
TASK ...INV NOT IN SYSTEM	15
SGN1 requires the task ...INV to operate but the task cannot be found.	
TOO MANY PARAMETERS	31
More than the allowable number of parameters have appeared in the directive.	
VIRTUAL ADDRESS SPACE OF SCOM OUT OF RANGE	13
The size specified for SCOM makes the virtual starting address less than 1000000; Administrator must reduce the size of SCOM. The maximum size of SCOM is 12K.	

2. PHASE 2 ERROR MESSAGES

ASSIGN LUN ERROR

SGN2 cannot assign a LUN to SY.

CHECKPOINT FILE ALLOCATION ERROR

There is not sufficient contiguous space on the disk to allocate the checkpoint file.

ERROR ON LOGGING DEVICE

Could not successfully log.

IO ERROR

I/O ERROR has occurred.

MARK TIME ERROR

SGN2 has attempted a MARK TIME and the request failed.

OPEN ERROR ON SYSBLD.CMD

SYSBLD.CMD could not be opened.

OPTION LINE SYNTAX ERROR

A syntax error exists in an SGN2 directive.

READ ERROR ON COMMAND FILE

Command file cannot be read.

REQUEST ERROR

SGN2 has attempted to REQUEST a task and the request failed.

WAITFOR ERROR

A WAITFOR request has failed.

## INDEX

- \ (backslash), 3-3
- / (slash), 3-3
  
- [311, 3-1, A14] ISTD64PAR.MAC,  
3-1, A-1
- [311, 107] SGDATA.MAC, 1-4, 4-1,  
A-1
  
- Allocating memory, 3-7
  
- Backslash (\), 3-3
- BM792YB bootstrap, 5-2
- BM873YA bootstrap, 5-3
- BM873YB bootstrap, 5-4
- BOOT command, D-2
- Bootstrapping, 2-3
  - distribution tape, 5-6
  - hardware, 2-3, 5-1
  
- Checkpointing, 3-19
- CKPNT directive, 1-3, 3-19
- CLI (Command Language Interpreter), 4-2
- Clock ticks, 3-5
- Command files,
  - see indirect command files,
- Command Language Interpreter (CLI), 4-2
- Common areas, 3-9
- CPU specification, 3-5
  
- Data structures, 4-1
- Defaults, system, 3-16
- \*DELAY directive, 1-4, 3-2
- DEV directive, 1-3, 3-13
- Device,
  - characteristics, 3-13, B-1, C-1, to C-6
  - specifications, 3-13
- Directives, 1-3, 3-3, to 3-22
  
- DISMOUNT command, D-6
- Distribution system, 1-1
  - transferring, 5-1 to 5-7
- DPAR directive, 1-3, 3-18
  
- Editing files, 2-1
- Error messages, E-1 to E-6
- EXEC directive, 1-3, 3-8
- Executive, 1-3, 3-8
  - IAS, 3-8
  - timesharing, 1-4
  
- File primitives, 3-21
- Files, system, A-1
- FIX-IN-MEMORY command, D-8
- Floating point option, 3-5
  
- Hardware bootstrapping, 2-3, 5-1
  
- IASBUF, 1-4, 4-1
- IASCOM, 1-4, 4-1
- Indirect command files, 1-2
  - editing, 2-1
  - phase 1, 6-1
  - use of in Stage 2, 1-4
- INITIALIZE VOLUME command, D-9
- INS directive, 1-3, 3-20
- Installing system tasks, 3-20
  
- Libraries,
  - IASCOM and IASBUF, 1-4, 4-1
- LIST COMMON BLOCKS command, D-19
- LOAD command, D-20
  
- MCR commands, D-1 to D-36
- Memory,
  - allocation, 3-7
  - size, 3-5
- MOUNT command D-21

INDEX (Cont.)

MR11DB bootstrap, 5-1

Node pool usage, 2-1, 3-11

PAR directive, 1-3, 3-12

Parameter file,  
 terminal handler, 3-1  
 timesharing executive, 1-4, 4-1

Parameters,  
 scheduling, 4-3  
 set-up, 4-2  
 swapping, 4-4  
 terminal handler, 3-1  
 timesharing executive, 1-4, 4-1

Partitioning memory, 3-12

PARTITIONS command, D-26

PDP11 directive, 1-3, 3-5

Phase 1 of Stage 1, 1-2  
 bootstrapping, 2-4  
 procedures, 6-2 to 6-8

Phase 2 of Stage 1, 1-3  
 bootstrapping, 2-5  
 directive (\*DELAY), 3-22  
 system disk, 3-17

POOL directive, 1-3, 3-11

Pool requirements for SGN1, 2-1

Procedures,  
 Stage 1, 6-1 to 6-11  
 Stage 2, 7-1 to 7-2

Processor configuration, 3-5

Rebooting saved system, 2-7

REDIRECT command, D-27

SAVE command, D-28

Saving generated system, 2-6

SCOM directive, 1-3, 3-9

SET command, D-30

SGAs (Shareable Global Areas),  
 1-4

SGN1, 2-4  
 bootstrapping, 2-4

SGN2, 1-3  
 bootstrapping, 2-5

Shareable Global Areas (SGAs),  
 1-4

Slash (/), 3-3

Stage 1, 1-2  
 procedures, 6-1 to 6-11

Stage 2, 1-4  
 procedures, 7-1 to 7-2

SY directive, 1-3, 3-17

SY:[11,17]SYSBLD.CMD, 1-4, 6-1,  
 6-9

Startup, 4-1

System,  
 common areas, 3-9  
 defaults, 3-16  
 definition, 3-1 to 3-22  
 distributed, 1-1, 5-1  
 to 5-7  
 rebooting saved, 2-7  
 saving, 2-6  
 startup, 4-1  
 transferring distributed, 5-1  
 to 5-7

target device, 1-3, 3-4, 6-2

TARGET directive, 1-3, 3-4

Terminal handler parameters,  
 1-4, 3-1

Terminals,  
 on DH11 or DJ11 lines, C-1 to  
 C-6  
 timesharing, 4-2

TIME command, D-32

Timesharing Executive, 1-4, 4-1

Transferring distributed system  
 5-1 to 5-7

UNFIX command, D-33

UNLOAD command, D-34

USER FILE DIRECTORY command,  
 D-35

READER'S COMMENTS

NOTE: This form is for document comments only. Problems with software should be reported on a Software Problem Report (SPR) form.

Did you find errors in this manual? If so, specify by page.

---

---

---

---

---

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

---

---

---

---

---

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

---

---

---

---

---

Please indicate the type of user/reader that you most nearly represent.

- Assembly language programmer
- Higher-level language programmer
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Non-programmer interested in computer concepts and capabilities

Name \_\_\_\_\_ Date \_\_\_\_\_

Organization \_\_\_\_\_

Street \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip Code \_\_\_\_\_

or  
Country

If you require a written reply, please check here.

Please cut along this line.

-----  
**Fold Here**  
-----

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

FIRST CLASS  
PERMIT NO. 33  
MAYNARD, MASS.

BUSINESS REPLY MAIL  
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

**digital**

Software Communications  
P. O. Box F  
Maynard, Massachusetts 01754

