DEC-11-OMGRA-A-D

BATCH-11/DOS-11

System Manager's Guide

FOR THE

BATCH OPERATING SYSTEM

AND

DISK OPERATING SYSTEM

Monitor Version VØØ8

January 1973

SOFTWARE SUPPORT CATEGORY

The software described in this document is supported by Digital Equipment Corporation under Category I as defined on page vi of this document.

For additional copies, order No. DEC-11-OMGRA-A-D from Digital Equipment Corporation, Software Distribution Center, Maynard, Massachusetts 01754.

January 1973

Your attention is invited to the last two pages of this document. The "How to Obtain Software Information" page tells you how to keep up-to-date with DEC's software. The "Reader's Comments" page, when filled in and mailed, is beneficial to both you and DEC; all comments received are acknowledged and are considered when documenting subsequent manuals.

NOTE

The material in this manual is for information purposes and is subject to change without notice. Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment which is not supplied by Digital Equipment Corporation.

Associated Documents:

DOS Monitor Programmer's Manual,
DEC-11-OMONA-A-D

Batch User's Guide
DEC-11-OBUDA-A-D

PREFACE

This manual provides information needed to install and manage a BATCH-11 or DOS-11 system. It is organized as follows:

Chapter	Purpose
1	Describe the contents of each distribution kit; define terminology.
2	Define procedures for installing the Monitor and system programs; describe backup and recovery procedures.
3	Define procedures for tailoring the Monitor; describe the configuration module.
4	Describe standard system modules (device drivers and EMT's, and how to add, delete or replace them with user-written, non-standard modules.

NOTE

Chapters 3 and 4 are intended for sophisticated users. Procedures defined are not recommended for inexperienced users.

5	Information pertinent to managing a BATCH-11 installation - default UIC, standard job termination procedure files.
6	PIP facilities of interest to system manager: file management, control of system usage, etc.

Appendices A through H contain information of general interest.

GUIDE TO THE MANUAL

This section will aid in gaining quick access to specific information.

NOTATION CONVENTIONS

All command strings shown in this manual must be terminated by typing the RETURN key, unless otherwise indicated. Although the RETURN key is typed to enter a line to the system, it does not echo on the terminal paper except to perform a carriage return/line feed operations.

Characters printed by the system are underlined to differentiate them from characters typed by the user.

Lower case letters appearing in a command string indicate notation variables whose values must be specified based on the installation configuration. For example, sysdev: indicates the system disk specifier and disdev: indicates the specifier for the distribution medium. The legal values for these notation variables are defined and explained when they are initially required in the system generation procedure.

SOFTWARE SUPPORT CATEGORIES

Digital Equipment Corporation (DEC) makes available four categories of software. These categories reflect the types of support a customer may expect from DEC for a specified software product. DEC reserves the right to change the category of a software product at any time. The four categories are as follows:

CATEGORY I
Software Products Supported at no Charge

This classification includes current versions of monitors, programming languages, and support programs provided by DEC. DEC will provide installation (when applicable), advisory, and remedial support at no charge. These services are limited to original purchasers of DEC computer systems who have the requisite DEC equipment and software products.

At the option of DEC, a software product may be recategorized from Category I to Category II for a particular customer if the software product has been modified by the customer or a third party.

CATEGORY II
Software Products that Receive Support for a Fee

This category includes prior versions of Category I programs and all other programs available from DEC for which support is given. Programming assistance (additional support), as available, will be provided on these DEC programs and non-DEC programs when used in conjunction with these DEC programs and equipment supplied by DEC .

CATEGORY III
Pre-Release Software

DEC may elect to release certain software products to customers in order to facilitate final testing and/or customer familiarization. In this event, DEC will limit the use of such pre-release software to internal, non-competitive applications. Category III software is only supported by DEC where this support is consistent with evaluation of the software product. While DEC will be grateful for the reporting of any criticism and suggestions pertaining to a pre-release, there exists no commitment to respond to these reports.

CATEGORY IV
Non-Supported Software

This category includes all programs for which no support is given

CONTENTS

Page

CHAPTER 1

INTRODUCTION TO BATCH/DOS-11 SYSTEM DISTRIBUTION AND GENERATION

1.0 OVERVIEW

The current distribution of the BATCH-11 and DOS-11 systems represents a significant departure from past DOS-11 distributions. In the past, the practice has been to tailor a system to the particular disk used as the system device (RC11, RF11, or RK11). This had the following effects:

1. Three different versions of the system had to be kept up-to-date;
2. Different system generation procedures had to be used for different disks.

Under the current scheme, one DOS-11 system, or one BATCH-11 system, is distributed, depending on which system the user has specified. System generation varies only slightly because of the disk used as the system device. Also, updates to a system are sent to all users, in a timely fashion, since only one version of DOS-11 or BATCH-11 has been distributed.

The contents of each distribution kit are:

1. A DOS-11 or BATCH-11 Monitor with Getting on the Air (GOA) batchstreams (note that BATCH-11 includes all DOS-11 facilities);
2. A System Loader program (SYSLOD);
3. System programs (MACRO-11 assembler, linker, editor, etc.); and,
4. A FORTRAN Compiler and Object Time System components (supported only on configurations of 12K or more core sizes). Pertinent information is contained in Getting FORTRAN On the Air, DEC-11-LFOAA-B-D.

Though each BATCH-11 or DOS-11 distribution kit contains the same components as every other BATCH-11 or DOS-11 kit, certain differences exist in the usability of these components. These differences are based on the memory size of the configuration. For instance, FORTRAN is not supported in less than 12K; 8K and 12K configurations must use somewhat limited versions of the MACRO-11 Assembler. In the case of the 8K user, a subset of the full Permanent Symbol Table and a PAL-11R subset of MACRO-11 (in overlaid form) must be used. The 12K user can use the full MACRO-11, but must use it in overlaid form.

The media used to distribute the BATCH-11 or DOS-11 system are described below.

1.1 DISTRIBUTION KITS

The BATCH/DOS-11 system is distributed as one of four kits, designed for the following media:

1. High-speed paper tape;
2. DECtape;
3. 7-track magnetic tape; and,
4. 9-track magnetic tape.

Each package supports the following disks as the system device: RC11, RF11, and RK11 (high density).

The system as distributed consists of one of the following kits:

1. DOS-11 Monitor, including system programs, and FORTRAN; or,
2. BATCH-11 Monitor, including system programs, and FORTRAN. Also included are the Getting on the Air (GOA) batch streams, which are listed in Appendix A of this manual.

1.1.1 Monitor Components

The Monitor components common to both kits are:

NAME	FUNCTION	MODULE FORMAT
SYSLOD.CIL	System Loader. Transfers MONLIB.LCL to disk	CIL (1.2)[1]
MONLIB.LCL	Monitor Library.	LICIL (1.2)[1]
RMON.LDA	Needed for Configuration Module modification. (Ch. 3)[1]	Load

[1] Refer to section or Chapter indicated, for details.

		MODULE
NAME	FUNCTION	FORMAT
RMON.STB	Input file for linking Configuration Module (3.3.8).	Symbol Table
TAIL.LDA	Concatenated load modules used to build Monitor. (Includes TMON).	Load
SYSTEM.MAC	Input for assembly of Configuration Module.	Source
CONFIG.MAC	Configuration Module.	Source
CFMAC.MAC	Configuration Module macros.	Source

The BATCH-11 Monitor includes (in addition to the above):

		MODULE
NAME	FUNCTION	FORMAT
CLØ.SYS to CL7.SYS	BATCH job termination processing modules - delete scratch files, output log, dump-on-error. (Ch. 5).	Source
LDUMP.LDA	Formatting program for error dumps.	Load
CMDS.MAC	BATCH command file.	Source
TMON.OBJ	Required for linking if CMDS file is modified by the installation.	Object
CCL.OBJ		Object
BATMAN.OBJ		Object
LKP.OBJ		Object

1.1.2 System Programs

The system programs included in both distribution kits are:

		MODULE
NAME	FUNCTION	FORMAT
MACRO.OBJ	The MACRO-11 Assembler object module. Usable in configurations of any size greater than 12K. Usable in 12K if linked with MACOVR.OBJ.	Object
MACROP.OBJ	PAL-11R subset of MACRO Assembler; for 8K systems.	Object
MACOVR.OBJ	Linked with MACRO.OBJ to produce overlay version for a 12K system.	Object
PST.OBJ	Full MACRO-11 symbol table.	Object
PSTP.OBJ	PAL-11R subset symbol table.	Object
PST.MAC	Source of MACRO-11 symbol table.	Source
SYSMAC.SML	System Macro library.	Source

NAME	FUNCTION	MODULE FORMAT
MACROS.LDA	Used only on 8K systems to assemble certain system modules, i.e., PST.MAC, CONFIG.MAC.	Load
EDIT.OBJ	Editor program	Object
ODT11R.OBJ	On-line Debugging program	Object
LIBR.OBJ	Librarian.	Object
FILDMP.OBJ	File Dump program.	Object
CREF.OBJ	Assembler Cross-Reference program. (Not available for 8K systems.)	Object
SPIP.8K PIP.12K PIP.16K	Special versions of PIP, used to transfer system programs to the system device. SPIP.8K transfers system programs on 8K configurations.	Load Load Load
PIP.OBJ		Object
PIPOVØ.OBJ PIPOV1.OBJ PIPOV2.OBJ PIPOV3.OBJ	PIP File Utility program modules. Each module represents an overlay or resident file.	Object Object Object Object
LINK.8K LINKOB.8K	Linker program, and Linker Overlay Builder, in 8K load module form (used to link system programs).	Load Load
LINK.12K LINKOB.12K	Linker program, and Linker Overlay Builder, in 12K load module form.	Load Load
LINK.OBJ LINKOB.OBJ	Object module of the Linker. Object module of Linker Overlay Builder	Object Object
CILUS.OBJ CILUSØ.OBJ CILUS1.OBJ CILUS2.OBJ CILUS3.OBJ	Core Image Library Update and Save program.	Object

1.1.3 FORTRAN Components

The FORTRAN components distributed consist of the compiler and its associated files, and the Object Time System (OTS) and its related modules. Refer to the manual, Getting FORTRAN on the Air, DEC-11-LFOAA-B-D.

NOTE

Though FORTRAN is distributed with all DOS-11 and BATCH-11 kits (including 8K configuration kits), it is supported only for 12K configurations and above.

1.1.4 Device Driver Sources

The following device driver source modules are included in each DOS-11 and BATCH-11 kit:

MODULE	FUNCTION
DC.MAC	RC11 driver.
DF.MAC	RF11 driver.
DK.MAC	RK11 driver.
CR.MAC	Card reader driver.
MT.MAC	Magnetic tape driver.
DT.MAC	DECtape driver.
KB.MAC	Console keyboard driver.
PR.MAC	Papertape reader driver.
PP.MAC	Papertape punch driver.
LP.MAC	Line printer driver.

1.2 TERMINOLOGY

The brief glossary that comprises this section defines the meanings of several terms used in subsequent sections of this manual.

block, logical

A unit of disk space, consisting of a number of contiguous physical blocks. Logical block size is a multiple of the physical block size, where the multiplier is a power of two. For example, the logical block size of a file on an RC11 disk might be 64, 128, 256, etc.

block, physical

The smallest addressable unit of disk storage, expressed as a decimal number of words. This varies according to the disk type, as:

RF11, RC11 - 64 words
RK11 - 256 words

boot

To load a program or Monitor into core, and begin its execution.

bootstrap

A section of code which reads and passes control to another body of code. Typically, a short bootstrap (called the ROM boot), which is wired into the processor, is started by toggling the control panel keys. The ROM boot reads one block (usually block Ø) from a specified device into location Ø, and gives it control at word Ø. This block of code, also a bootstrap, then loads the Monitor.

CIL

Core Image Library (acronym). A CIL consists of one or more Images, and an Index which contains a communications directory (COMD) for each core image in the CIL, and a list of information describing the CIL itself (the CIL Line).

CIL Line

The formatted binary line that contains information describing the CIL to which it belongs.

COMD

COMmunications Directory (acronym). The entry in the CIL Index that describes the core image, so that the core image can be loaded by the DOS Monitor.

Core image

The representation of a program as it appears when loaded in core, ready to execute. Conceptually, the result of simply writing an area of core to a device with no special formatting.

Hook

To enter key information about a CIL into the bootstrap (starting block, size, etc.). There may be many CIL's resident on a disk or DECtape, but only one may be "hooked" to the bootstrap. The hooked CIL (usually a Monitor) is the one that is loaded and executed when the bootstrap is run.

LICIL

LInked Core Image Library. A backup version of a CIL, in linked, rather than contiguous, format. LICIL's can be stored on disk, DECtape, paper tape, or magnetic tape. They are stored as linked files, because DOS-11 does not support contiguous files on all media, and because some information in the CIL depends on the block size of the device.

1.3 SYSTEM FACILITIES

One of the principal goals of the BATCH/DOS-11 system is to provide the user with capabilities to build, update, and maintain his system more easily than in the past. An enhanced System Loader (SYSLOD) greatly simplifies the task of loading the Monitor. The CILUS program allows the user to create and modify the Monitor and other programs that are stored in the form of CIL's (core image libraries; refer to section 1.2 for a definition of CIL's).

1.3.1 System Loader

The System Loader (SYSLOD) loads the BATCH or DOS (or other) Monitor onto the system disk, boots it into core, and executes it. SYSLOD is a stand-alone program, in that it does not require DOS-11 or the DOS file structure on a disk. The functions of SYSLOD are described in Chapter 2 ("Getting On the Air").

1.3.2 CILUS Program

CILUS is an acronym derived from Core Image Library Update and Save. It is a program that allows the user to build and maintain programs in the form of core image libraries (both contiguous and linked). Its applicability to BATCH or DOS system generation derives from the fact that the BATCH or DOS Monitor is built and distributed in the form of a LICIL (linked core image library). CILUS is used when the user wishes to modify,test, or store a version of the Monitor. Appropriate information needed to use CILUS for these functions is provided in Appendix G.

CHAPTER 2

GETTING ON THE AIR

2.0 OVERVIEW OF INSTALLATION PROCEDURES

This Chapter describes the process by which a user takes a distribution kit, and builds a running BATCH-11 or DOS-11 system. The exact details of how this is done depend on the distribution medium, and the memory size of the configuration, but in general it involves:

1. Loading and executing the Monitor, via SYSLOD;
2. Loading and linking system programs on the system device;
3. Loading FORTRAN components (see Getting FORTRAN on the Air, DEC-11-LFOAA-B-D); and,
4. Creating a backup copy of the system.

2.1 SYSTEM LOADER (SYSLOD)

The System Loader program (SYSLOD) loads a Monitor core image library (CIL) onto the system disk from the backup medium. Sections 2.1.1 through 2.1.4, below, consist of the procedures necessary to get the BATCH/DOS-11 system on the air.

2.1.1 Paper Tape Procedures

A. No BM792-YA ROM

When the BM792-YA ROM option is not available, use the following procedure:

1. Toggle the Bootstrap Loader into core via the Console, loading the sequence of instructions shown below into the locations specified (i.e., nn7744 to nn7776, where nn is determined by available core; refer to Table 2-1).

The actual procedure is to set the Console Switch Register to the address nn7744, and depress LOAD ADDR. This puts nn7744 into the Address Register. Now set the Console Switch Register to 167Ø1, and Raise the DEP button. The value 167Ø1 is deposited into address nn7744, and the Address Register is automatically incremented by 2, as it is each time DEP is toggled. Continue by specifying the values shown in CONT below, toggling DEP after each value is set in the Console Switch Register.

LOC	CONT
nn7744	167Ø1
nn7746	26
nn775Ø	127Ø2
nn7752	352
nn7754	5211
nn7756	1Ø5711
nn766Ø	1ØØ376
nn7762	116162
nn7764	2
nn7766	nn74ØØ
nn777Ø	5267
nn7772	177756
nn7774	765
nn7776	17755Ø

2. Set Console ENABLE/HALT to HALT.

3. Place the Absolute Loader paper tape into the high-speed paper tape reader. The special leader code (351) must be positioned over the read sensors.

4. Set the Switch Register to nn7744, where nn is as specified in Step 1.

5. Depress LOAD ADDR.

6. Set ENABLE/HALT to ENABLE.

7. Depress START. (The Absolute Loader is read into core, and the machine halts.)

8. Place the SYSLOD tape, labelled DEC-11-OSLDA-A-PL, in the high-speed paper tape reader with blank leader tape over the read sensors.

9. Set ENABLE/HALT to HALT.

10. Set the Switch Register to nn75ØØ, where nn is as specified in Step 1.

11. Depress LOAD ADDR.

12. Set ENABLE/HALT to ENABLE.

13. Depress START.

SYSLOD is read into core. Skip to Section 2.1.4.

Table 2-1
VALUES SPECIFIED FOR NN

MEMORY SIZE	VALUE OF nn
8K	∅3
12K	∅5
16K	∅7
2∅K	11
24K	13
28K	15

B. BM792-YA ROM Available

When the BM792-YA ROM option is available, use the following procedure:

1. Set Console ENABLE/HALT to HALT.
2. Set the Switch Register to 173∅∅∅.
3. Replace the Absolute Loader paper into the high-speed paper tape reader, with the special leader code (351) over the read sensors.
4. Depress LOAD ADDR.
5. Depress START. (The Absolute Loader is read into core.)
6. Place the SYSLOD tape, labelled DEC-11-OSLDA-A-PL, in the high-speed paper tape with blank leader tape over the read sensors.
7. Set ENABLE/HALT to HALT.
8. Set the Switch Register to nn75∅∅, where nn is determined according to memory size (see Table 2-1).
9. Depress LOAD ADDR.
10. Set ENABLE/HALT to ENABLE.
11. Depress START. Skip to Section 2.1.4.

2.1.2 DECtape Procedures

The SYSLOD program is automatically loaded from DECtape using either the BM792-YB Bootstrap Loader, or the MR11-DB Bulk Storage Bootstrap Loader, depending on which is included in the hardware configuration.

✓ BM792-YB ROM Procedure

1. Mount DECtape containing SYSLOD, labelled
DOS → DEC-11-ODMOA-A-UC, or DEC-11-OBMOA-A-UC, on DECtape unit Ø. (ODMOA is the DOS-11 version; OBMOA is the BATCH-11 version.)

2. Set the REMOTE and WRITE LOCK switches on unit Ø.

3. Set the Console ENABLE/HALT switch to HALT.

4. Set the Console Register to 1731ØØ.

5. Depress LOAD ADDR.

6. Set ENABLE/HALT to ENABLE.

7. Set the Switch Register to 177344.

8. Depress START.

SYSLOD is read into core. Skip to Section 2.1.4.

MR11-DB Procedure

1. Mount DECtape DEC-11-ODMOA-A-UC or DEC-11-OBMOA-A-UC on unit Ø.

2. Set the REMOTE and WRITE LOCK switches on unit Ø.

3. Set the Console ENABLE/HALT switch to HALT, then to ENABLE.

4. Set the Console Switch Register to 17312Ø.

5. Press LOAD ADDR.

6. Press START.

SYSLOD is loaded into core. Skip to Section 2.1.4.

2.1.3 Magtape Procedures

Two procedures may be used to load SYSLOD from magnetic tape (either 7- or 9-channel tape). The first is to use the MR11-DB Bootstrap Loader as follows:

1. Mount magnetic tape DEC-11-ODFSA-A-MC, or DEC-11-OBFSA-A-MC on unit Ø. The tape must be at the load point. Turn the unit ON LINE. (ODFSA is DOS-11; OBFSA is BATCH-11.)
2. Set Console ENABLE/HALT to HALT, then to ENABLE,
3. Set the Console Switch Register to 173136.
4. Depress LOAD ADDR.
5. Depress START.

SYSLOD is loaded into core. Skip Section 2.1.4.

NOTE

If SYSLOD does not type out its identification and the processor halts, it indicates that a parity error was detected in reading the tape. Retry the procedure. If the error persists, it will be necessary to obtain a replacement tape.

When the MR11-DB Bootstrap Loader is not available, use the following procedure:

1. Mount magnetic tape on unit Ø, as above.
2. Deposit the following routine into core via the Console Switch Register and the DEP switch.

Address	Contents	Symbolic
1ØØØØ	127ØØ	MOV #172524,RØ
1ØØØ2	172524	
1ØØØ4	531Ø	DEC (RØ)
1ØØØ6	1274Ø	MOV #6ØØ11,-(RØ)
1ØØ1Ø	6ØØ11	
1ØØ12	1Ø571Ø	TSTB (RØ)
1ØØ14	1ØØ376	BPL .-2
1ØØ16	571Ø	TST (RØ)
1ØØ2Ø	1ØØ767	BMI .-2Ø
1ØØ22	1271Ø	MOV #6ØØØ3,(RØ)
1ØØ24	6ØØØ3	
1ØØ26	1Ø571Ø	TSTB (RØ)
1ØØ3Ø	1ØØ376	BPL .-2
1ØØ32	571Ø	TST (RØ)
1ØØ34	1ØØ777	BMI .
1ØØ36	5ØØ7	CLR PC

NOTE

When an earlier version of DOS is available, the MODIFY command can be used to generate this routine.

3. Put 1ØØØØ in the Switch Register.
4. Depress LOAD ADDR.
5. Put ENABLE/HALT to ENABLE.
6. Depress START.

SYSLOD is read into core. Skip to Section 2.1.4.

2.1.4 Loading the Monitor

When SYSLOD is in core, it identifies itself by printing the following message at the keyboard:

```
SYSLOD Vxx
#
```

(Note: xx is replaced by the SYSLOD version number.)

The Monitor is loaded by typing in the following command string:

```
#sysdev:MONLIB.CIL/ZERO/HOOK/BOOT<disdev:MONLIB.LCL
```

CAUTION

The above command zeros the disk specified as "sysdev." Be sure to read section 2.4 before loading a VØ8 Monitor onto a current system disk.

where:

sysdev: is the system disk specifier, i.e.,

DC: = RC11
DF: = RF11
DKØ:= RK11 disk unit Ø.

disdev: is the specifier for the distribution medium, i.e.,

DTØ:= DECtape unit n;
MTØ:= Magtape unit n; and,
PR: = Paper tape reader. Paper tapes containing MONLIB.LCL must be mounted and read in the order DEC-11-ODMLA (or OBMLA)-A-P01 to -P05, sequentially. The first paper tape must be mounted before the command string is input. After each of the tapes P01 to P04 is read, the message

```
AØØ2  Ø6332Ø
PLEASE PRESS CONTINUE WHEN READY
```

is printed. After mounting the next tape, depress the CONTINUE switch on the console panel to read in the next segment of MONLIB.LCL. After the last paper tape is read, the Monitor comes on the air as described below.

Switches

The switches that may be specified in the SYSLOD command string are /ZERO, /HOOK, and /BOOT. Note that in the example these switches are completely spelled out. SYSLOD recognizes only the first two letters of each switch, however, so they can be abbreviated to /ZE, /HO, /BO, and /FO, at the user's option.

/ZERO is the switch used to initialize the disk to the BATCH/DOS-11 file structure. This switch also clears the disk.

NOTE

The /FORMAT switch may be specified when the output device is the RK11 disk. This switch causes the disk to be formatted with hardware information (read marks, etc.). (See Appendix F.)

/HOOK   is the switch that causes the Monitor to be hooked to the bootstrap that loads the system.

/BOOT   is the switch that causes the system to be bootstrapped.

/HOOK and /BOOT are SYSLOD defaults, and are not required in the command. Any error detected during SYSLOD operation is reported via a message at the keyboard. These messages are summarized in Appendix F.

The first time the Monitor comes on the air, it prints the following message, related to line frequency preceding its identification:

```
SYSLD COMPLETE!
ANSWER WITH CARRET OR 'Y' CARRET
IS YOUR LINE FREQUENCY 5Ø HERTZ?
```

Return for 60~

The Monitor waits for either a carriage return, or a Y followed by a carriage return. Y↵ indicates that the system is using a line frequency of 5Ø Hertz. If the response is simply ↵, 6Ø Hertz is assumed. (↵ indicates a carriage return, or CARRET.)

The Monitor identifies itself by printing:

```
DOS Vxx    or    BATCH Vxx
$                $
```

Any error detected during the Monitor initialization procedures is reported via a brief message on the teleprinter. Refer to Table 3-1, Chapter 3. Should problems arise at this point, contact a DEC Software Specialist.

The Monitor can now be used to load and link system programs.

Use the DATE and TIME commands to enter the correct date and time, and log-in, e.g.;

```
$DATE 9-OCT-72
$TIME 1Ø:Ø5
$LO 1,1
```

You are now ready to load and link system programs. Refer to Section 2.2. RC11 systems users should refer specifically to 2.2.3.

2.1.5 Rebooting the Monitor

Rebooting the Monitor is accomplished in one of two ways:

A. BM792-YB Procedure

1. If the system device is an RK11 disk, turn WRITE ENABLE off. Otherwise go to Step 2.
2. Depress HALT.
3. Set the Switch Register to 1731ØØ.
4. Depress LOAD ADDRESS.
5. Set the Switch Register to one of the following values:

   177462 - RF11 disk
   1774Ø6 - RK11 disk
   17745Ø - RC11 disk

6. Turn HALT/ENABLE to ENABLE.
7. Press START.

B. MR11-DB Procedure

1. Set the Console Switch Register to one of the following values according to which device is involved:

   RF11 = 1731ØØ
   RK11 = 17311Ø
   RC11 = 17322Ø

2. Depress HALT.
3. Set Console HALT/ENABLE to ENABLE.
4. Depress LOAD ADDR.
5. Depress START.

A successful Monitor reboot is indicated by a response of

```
DOS Vxx   or   BATCH Vxx
$              $
```

at the keyboard. If this response is not forthcoming, retry the above procedure, making certain the correct Switch Register settings are used. Should a second attempt elicit no response, it is probable that the disk has been corrupted, and the Monitor is unusable. The cause of the problem should be determined, but the system disk will probably have to be reloaded.

2.2 GETTING SYSTEM PROGRAMS ON THE AIR

Once the Monitor is running, and the log-in procedure is completed, it is necessary to load and link system programs. For BATCH-11 users, this is done by running a DEC-supplied batch stream, which resides on the distribution medium; refer to section 2.2.1.

Users of the DOS-11 Monitor must follow the procedures described in section 2.2.2. RC11 users should refer to section 2.2.3.

2.2.1 BATCH-11 Procedure

1. Run PIP, by typing in the following command:

```
$RU dev:PIP.16K
```

where "dev" is replaced by the mnemonic name of the distribution medium, viz.;

```
PR:  - Paper tape reader
DTØ: - DECtape unit Ø
MTØ: - Magnetic tape unit Ø
```

NOTE 1

When the input medium is paper tape, it is not necessary to include a file name. The appropriate paper tape must be mounted in the reader prior to issuing the command, however. For DECtape and magnetic tape, the file name is obligatory.

NOTE 2

DECtape users, mount DECtapes as follows:

Label	Unit
DEC-11-OCSPA-A-UC1	Ø
DEC-11-OCSPA-A-UC2	1

2. Type the following sequence of command strings to PIP

Paper Tape

```
#SY:GOAP16.DOS<PR:/FA/SU     Mount GOAP16.DOS in reader
#SY:GOAP.16K<PR:/FA/SU       Mount GOAP.16K in reader
#SY:GOAP.ALL<PR:/FA/SU       Mount GOAP.ALL in reader
#SY:GOTA.16L<PR:/FA/SU       Mount GOTA.16L in reader
```

Go to step 3.

(Note: /FA indicates Formatted ASCII; /SU is the abbreviated form of /SUPERSEDE. See the PIP-11 Programmer's Manual for a discussion of PIP switches.)

DECtape

```
#SY:<DT1:GOTA16.DOS/FA/SU
#SY:<DT1:GOTA.16K/FA/SU
#SY:<DT1:GOTA.ALL/FA/SU
#SY:<DT1:GOTA.16L/FA/SU
```

Go to step 3.

Magnetic Tape

```
#SY:<MTØ:GOAM16.DOS/FA/SU
#SY:<MTØ:GOAM.16K/FA/SU
#SY:<MTØ:GOAM.ALL/FA/SU
#SY:<MTØ:GOTA.16L/FA/SU
```

3. Kill PIP;

```
#↑C
.KI
```

4. Invoke BATCH, specifying the appropriate batch stream:

```
$BA  GOAP16 .DOS      (papertape)
$BA  GOTA16 .DOS      (DECtape)
$BA  GOAM16 .DOS      (Magnetic tape)
```

At this point, the BATCH-11 Monitor begins reading a batch stream from the system device. Refer to Appendix A for a listing of the command strings in the batch stream which load and link system programs. Messages are printed during the batch stream execution, indicating which tapes to mount. After mounting the indicated tape, type the CO command to continue. When loading and linking of system programs is completed, $FINISH and BATCH V xx messages are printed. At this point, refer to the FORTRAN Getting on the Air document. If FORTRAN is not being installed, refer to section 2.3 for back-up procedures.

2.2.2 DOS-11 Monitor Procedures

There is some variation in the procedures used for one distribution medium or another, or for different system devices. RC11 users should refer to section 2.2.3. RF11 and RK11 users proceed as follows:

1. Load the non-overlaid 12K version of PIP, using one of the following Monitor commands:

```
$RUN PR:                Paper tape - put tape labelled
                        DEC-11-UP12A-A-PL in the reader
                        prior to issuing command.

$RUN DTØ:PIP.12K        DECtape DEC-11-OCSPA-A-UC1
                        should now be on DTØ

$RUN MTØ:PIP.12K        Magnetic tape distribution
                        (7- or 9-track)
```

   Note that DECtape and magnetic tape users, with 8K configurations, must specify the special version of PIP in the above commands, as $RUN DTØ:SPIP.8K or $RUN MTØ:SPIP.8K respectively.

   PIP is loaded, and responds

```
PIP Vxx
#
```

2. Transfer the Linker load modules from the distribution medium to the disk, using PIP. The appropriate Linker load modules to transfer are determined by system core size, as shown in Table 2-2.

TABLE 2-2

CORE SIZE - LINKER RELATIONSHIP

Core Size	Modules to Transfer
8K	LINK.8K LINKOB.8K
12K up	LINK.12K LINKOB.12K

The command strings specified must be one of the following pairs:

For 8K

```
#LINK.LDA<PR:/FB   }      Paper tape - tape DEC-11-UL8KA-A-PL1
#LINKOB.LDA<PR:/FB }      Tape DEC-11-UL8KA-A-PL2

#LINK.LDA<DTØ:LINK.8K/FB     } DECtape DEC-11-OCSPA-A-UC1 on
#LINKOB.LDA<DTØ:LINKOB.8K/FB } unit Ø

#LINK.LDA<MTØ:LINK.8K/FB     }  Magnetic tape DEC-11-ODFSA-A-MC
#LINKOB.LDA<MTØ:LINKOB.8K/FB }  or DEC-11-OBFSA-A-MC
```

For 12K or more:

#LINK.LDA<PR:/FB	Place paper tape DEC-11-UL12A-A-PL1 in reader.
#LINKOB.LDA<PR:/FB	Place paper tape DEC-11-UL12A-A-PL2 in reader.
#LINK.LDA<DTØ:LINK.12K/FB #LINKOB.LDA<DTØ:LINKOB.12K/FB	DECtape DEC-11-OCSPA-A-UC1 on unit Ø
#LINK.LDA<MTØ:LINK.12K/FB #LINKOB.LDA<MTØ:LINKOB.12K/FB	Magnetic tape DEC-11-ODFSA-A-MC or OBFSA-A-MC on unit Ø

3. Transfer the appropriate MACRO object modules - refer to Table 2-3. Note that 8K configurations use the PAL-11 subset of MACRO (MACROP), a smaller Permanent Symbol Table (PSTP), and an Overlay version. The sole difference between 12K and over 12K MACRO is that 12K uses the overlay version.

TABLE 2-3

CORE SIZE - MACRO RELATIONSHIP

Core Size	Module to Transfer
8K	MACROP.OBJ PSTP.OBJ MACOVR.OBJ
12K	MACRO.OBJ MACOVR.OBJ PST.OBJ
Over 12K	MACRO.OBJ PST.OBJ

The command strings are:

8K Paper Tape	Tape to Mount (DEC-11-prefix)
#MACROP.OBJ<PR:/FB/SU	LMOPA-A-PR
#PSTP.OBJ<PR:/FB/SU	LMTPA-A-PR
#MACOVR.OBJ<PR:/FB/SU	LMOVA-A-PR

NOTE

In the following examples, SY: is used when transferring files from DECtape or magnetic tape. SY: indicates the system device. Files are transferred to the system device, with the correct file name and extension.

```
8K DECtape                              Tape to Mount  (DEC-11-prefix)

#SY:<DTØ:MACROP.OBJ/FB/SU
#SY:<DTØ:PSTP.OBJ/FB/SU                     OCSPA-A-UC1
#SY:<DTØ:MACOVR.OBJ/FB/SU

8K Magnetic Tape

#SY:<MTØ:MACROP.OBJ/FB/SU                  ODFSA-A-MC   (DOS-11)
#SY:<MTØ:PSTP.OBJ/FB/SU                    OBFSA-A-MC   (BATCH-11)
#SY:<MTØ:MACOVR.OBJ/FB/SU

12K Paper Tape

#MACRO.OBJ<PR:/FB/SU                        LMCOA-A-PR
#PST.OBJ<PR:/FB/SU                          LMSTA-A-PR
#MACOVR.OBJ<PR:/FB/SU                       LMOVA-A-PR

12K DECtape

#SY:<DTØ:MACRO.OBJ/FB/SU
#SY:<DTØ:PST.OBJ/FB/SU                     OCSPA-A-UC1
#SY:<DTØ:MACOVR.OBJ/FB/SU

12K Magnetic Tape

#SY:<MTØ:MACRO.OBJ/FB/SU
#SY:<MTØ:PST.OBJ/FB/SU                     ODFSA-A-MC  or
#SY:<MTØ:MACOVR.OBJ/FB/SU                  OBFSA-A-MC

Over 12K - Paper Tape

#MACRO.OBJ<PR:/FB/SU                       LMCOA-A-PR
#PST.OBJ<PR:/FB/SU                         LMSTA-A-PR

Over 12K - DECtape

#SY:<DTØ:MACRO.OBJ/FB/SU
#SY:<DTØ:PST.OBJ/FB/SU                     OCSPA-A-UC1

Over 12K - Magnetic Tape

#SY:<MTØ:MACRO.OBJ/FB/SU                   ODFSA-A-MC  or
#SY:<MTØ:PST.OBJ/FB/SU                     OBFSA-A-MC
```

4. Continue using PIP until all system programs have been transferred. (Paper tape users will note that tapes are labelled as shown in Appendix D; tapes must be placed in the reader prior to issuing the command strings.)

Paper Tape Procedure

The general form of the command used to transfer files from paper tape is:

```
#progname.ext<PR:/FB/SU
```

where "progname.ext" is replaced by the program name and extension of the file to be transferred.

Using this form, transfer the following programs:

Program name & ext

```
EDIT.OBJ
LIBR.OBJ
FILDMP.OBJ
CREF.OBJ
LINK.OBJ
LINKOB.OBJ
PIP.OBJ
PIPOVØ.OBJ
PIPOV1.OBJ
PIPOV2.OBJ
PIPOV3.OBJ
CILUS.OBJ
CILUSØ.OBJ
CILUS1.OBJ
CILUS2.OBJ
CILUS3.OBJ
```

Specify also:

```
#SYSMAC.SML<PR:/FA/SU
```

DECtape Procedure

Using PIP, specify the following commands:

```
           wild card
#SY:<DTØ:*.OBJ/FB/SU          Mount DEC-11-OCSPA-A-UC1
#SY:<DT1:*.OBJ/FB/SU          on DTØ
#SY:<DT1:SYSMAC.SML/FA/SU     Mount DEC-11-OCSPA-A-UC2
                              on DT1
```

Magnetic Tape Procedure

```
#SY:<MTØ:*.OBJ/FB/SU
#SY:<MTØ:SYSMAC.SML/FA/SU
```

5. Link system programs as follows:

   a. Kill PIP:

   ```
   ↑C
   .KILL
   ```

   b. Run Link Overlay Builder

   ```
   $RUN LINKOB
   ```

   (no command string is used)

   c. Run Link:

   ```
   $RUN LINK
   ```

Users with 16K or more of core memory can now utilize the following command strings to link the maximum size version of the Linker:

```
#LINK.LDA<LINK.OBJ/E
#LINKOB.LDA<LINKOB.OBJ/E
↑C
.KI
$RUN LINKOB
$RUN LINK
```

d. Link correct version of MACRO by specifying one of the following command strings:

```
#MACRO<MACROP.OBJ,MACOVR.OBJ,PSTP.OBJ/E      (8K)
#MACRO<MACRO.OBJ,MACOVR.OBJ,PST.OBJ/E        (12K)
#MACRO<MACRO.OBJ,PST.OBJ/E               (over 12K)
```

e. Link PIP to obtain the overlaid version, saving core:

```
#PIP<PIP.OBJ/CC/OV:4/E
#PIPOVØ<PIPOVØ.OBJ/CC/E
#PIPOV1<PIPOV1.OBJ/E
#PIPOV2<PIPOV2.OBJ/CC/E
#PIPOV3<PIPOV3.OBJ/E
```

f. Link remaining system programs:

```
#EDIT<EDIT.OBJ/E
#LIBR<LIBR.OBJ/E
#FILDMP<FILDMP.OBJ/T:3746Ø/E
#CREF<CREF.OBJ/E (omitted for 8K configurations)
#CILUS<CILUS.OBJ/CC/OV:4/E
#CILUSØ.OVR<CILUSØ.OBJ/E
#CILUS1.OVR<CILUS1.OBJ/E
#CILUS2.OVR<CILUS2.OBJ/E
#CILUS3.OVR<CILUS3.OBJ/CC/E
```

6. Delete object modules from system disk:

a. Kill the Linker:

```
↑C
.KILL
```

b. Run PIP:

```
$RUN PIP
PIP Vxx
#ODT/RE<ODT11R.OBJ
#*.OBJ/DELETE
```

c. `#ODT.OBJ/RE<ODT`

x d. Install FORTRAN (12K or larger configurations only). See Getting FORTRAN on the Air, DEC-11-LFOAA-B-D.

Rollin

2.2.3 RC11 Procedures

When the system disk is an RC11, the procedures described in 2.2.3.1 (BATCH) or 2.2.3.2 (DOS) must be used.

2.2.3.1 BATCH Procedures

1. Mount DEC-11-OCSPA-A-UC1 on DTØ;
   mount DEC-11-OCSPA-A-UC2 on DT1;

2. Invoke PIP, as

```
$RUN DTØ:PIP.16K
PIP-11Vxx
```

3. Specify the following command string to PIP:

```
#DC:<DT1:GOTA16.DC
```

Dismount DEC-11-OCSPA-A-UC2, and mount and WRITE ENABLE a back-up DECtape on DT1. The UIC [1,1] must be entered on the back-up DECtape, which is done by the PIP command:

```
#DT1:/EN
```

4. Kill PIP:

```
#↑C
.KI
```

5. Enter the BATCH command, as:

```
$BA GOTA16.DC
```

6. In the course of processing, two messages will be printed at the keyboard. The first is:

```
LINKSY:MOUNT DEC-11-OCSPA-A-UC1 ON DTØ:  WRITE LOCK
LINKSY:MOUNT BACK-UP DECTAPE ON DT1:  WRITE ENABLE
AØ5Ø   ØØØØØØ
```

Mount the indicated DECtapes, and type

```
$CO
```

The second message is:

```
LINKSY:MOUNT DEC-11-OCSPA-A-UC2 ON DT1:  WRITE LOCK
AØ5Ø   ØØØØØØ
```

After the back-up DECtape has been dismounted, and this DECtape mounted, type

```
$CO
```

The completion of processing is indicated by the print-out of a directory listing followed by:

```
$FINISH
TIME:-xx:xx:xx    (approximate duration = 20 minutes)
BATCH Vxx
```

2.2.3.2 DOS Procedures

```
 1.  Mount DEC-11-OCSPA-A-UC1 on DTØ:  WRITE LOCK  \ Read NOTE before
     Mount Back-up DECTAPE on  DT1:  WRITE ENABLE  / starting
 2.  $RUN DT:LINKOB.12K
 3.  $RUN DT:PIP.16K
     #DC:<DT:MACRO.OBJ,PST.OBJ
     #↑C
     .KI
 4.  $RUN DT:LINK.12K
     #DC:<MACRO,PST/E
     #↑C
     .KI
 5.  $RUN DT:PIP.16K
     #DT1:<MACRO.LDA/SU
     #MACRO.OBJ,PST.OBJ,MACRO.LDA/DE
     #DC:<DT:PIP.OBJ,PIPOVØ.OBJ,PIPOV1.OBJ
     #DC:<DT:PIPOV2.OBJ,PIPOV3.OBJ
     #↑C
     .KI
 6.  $RU DT:LINK.12K
     #DC:<PIP/CC/OV:4/E
     #DC:<PIPOVØ/CC/E
     #DC:<PIPOV1/E
     #DC:<PIPOV2/CC/E
     #DC:<PIPOV3/E
     #↑C
     .KI
 7.  $RU PIP
     #*.OBJ/DE
     #DC:<DT:LINK.OBJ,LINKOB.OBJ
 8.  $RU DT:LINK.12K
     #DC:<LINK/E
     #DC:<LINKOB/E
     #↑C
     .KI
 9.  $RU LINKOB
10.  $RU PIP
     #LINKOB.*/DE,LINK.OBJ
     Mount DEC-11-OCSPA-A-UC2 on DT1: WRITE LOCK
     #DC:<DT1:*.OBJ
     #↑C
     .KI
11.  $RU LINK
     #DC:<CILUS/CC/OV:4/E
     #CILUSØ.OVR<CILUSØ/E
     #CILUS1.OVR<CILUS1/E
     #CILUS2.OVR<CILUS2/E
     #CILUS3.OVR<CILUS3/CC/E
     #↑C
     .KI
12.  $RU PIP
     #*.OBJ/DE
     #DC:<DT:EDIT.OBJ,CREF.OBJ
     #↑C
     .KI
```

NOTE

The procedures shown here are for 16K configurations. For 8K configurations, use LINK.8K, LINKOB.8K, and SPIP.8K. Copy LINKOB.8K to disk, run it from disk, and delete it after use. At step 3, for the transfer command, use:

```
#DC:<DT:MACROP.OBJ,PSTP,OBJ,MACOVR.OBJ
```

At step 4, for the transfer command, use:

```
#DC:<DT:MACROP.OBJ,PSTP.OBJ,MACOVR.OBJ/E
```

For 12K systems, use LINK.12K, LINKOB.12K (as shown), and PIP.12K (instead of PIP.16K). At step 3, for the transfer command, use:

```
#DC:<DT:MACRO.OBJ,PST.OBJ,MACOVR.OBJ
```

At step 4, for the transfer command, use:

```
#DC:<DT:MACRO.OBJ,PST.OBJ,MACOVR.OBJ/E
```

```
13.  $RU LINK
     #DC:<EDIT/E
     #DC:<CREF/E
     #↑C
     .KI
14.  $RU PIP
     #*.OBJ/DE
     #DC:<DT:LIBR.OBJ,FILDMP.OBJ
     #↑C
     .KI
15.  $RU LINK
     #DC:<FILDMP/T:3746Ø/E
     #DC:<LIBR/E
     #↑C
     .KI
16.  $RU PIP
     #*.OBJ/DE
     #DC:ODT.OBJ<DT:ODT11R.OBJ
     #DC:/DI
```

2.3 BACKUP PROCEDURES

2.3.1 Use of ROLLIN

The ROLLIN utility program, if available, can be used to dump the system disk to a backup storage device (magnetic tape, DECtape or on an alternate RK11 disk). The procedure is:

1. Load and start ROLLIN (see Chapter 4 of the ROLLIN manual for these procedures).
2. For DECtape backup, specify:

   ```
   #DTØ:<disk:
   ```

   where disk: is DKØ:, DF:, or DC:, for RK11, RF11, or RC11, respectively. The number of DECtapes required to back-up the disk varies: 1Ø DECtapes are needed for a 2200 bpi RK11 cartridge.

3. For magnetic tape backup, specify:

   ```
   #MTØ:filename<disk:
   ```

   where a file name (e.g., SYSFIL) is assigned to the backup copy, and disk: is specified as DKØ:, DF:, or DC:.

4. For an alternate RK11 cartridge, specify:

   ```
   #DK1:<DKØ:
   ```

   The entire contents of DKØ: are copied to DK1:.

Restoring the system is done by reversing the order of the command string, e.g.;

```
#DKØ:<DTØ:
    or
#DKØ:<MTØ:SYSFIL/FIND
```

The /FIND switch and operating procedures are described in the document PDP-11 ROLLIN UTILITY PROGRAM, DEC-11-OROAA-A-D. To obtain ROLLIN, order either DEC-11-OROLA-A-PB (paper tape), or DEC-11-OROLA-A-UB (DECtape) from the Software Distribution Center, Main St., Maynard Mass.

2.3.2 Use of PIP

The PIP file utility program can be used to store backup copies of specific programs; for example, the load module form of MACRO can be backed up on DECtape by running PIP, and specifying the command string:

```
#DTØ:<MACRO.LDA/FB
```

To copy most system files, specify the command

```
#DTØ:<*.LDA[1,1]
```

Files not copied as a result of this command may be copied individually.

NOTE

MONLIB.CIL should not be copied. It is backed up on the distribution medium as MONLIB.LCL, and must be reloaded by SYSLOD.

The copies can be restored to the disk by the PIP command string:

```
#SY:<DTØ:*.LDA/FB
```

2.3.3 Reloading the System Disk

Should it be necessary to reload the system disk, the method used depends on whether ROLLIN was used to back up the disk. If it was, ROLLIN can be used to restore the disk. Follow the procedures described in the ROLLIN document, pertinent to the medium used to back up the system disk.

If ROLLIN was not used, the Monitor must be reloaded according to the procedures in section 2.1. Then, PIP can be used to copy backup versions of system files to the disk. If no backup versions were created, follow the procedures in section 2.2. If the backup medium (e.g., DECtape Ø) contains only system files, the following command to PIP can be used:

```
#dev:<DTØ:*.*[1,1]
```

2.4 USING SYSLOD WITH AN EXISTING SYSTEM DISK

A VØ8 Monitor (BATCH or DOS) can be installed on an existing system disk by following the general procedures outlined below:

1. Put all files on a backup medium (e.g., DECtape or magnetic tape), via PIP;

2. Load the system, as described in sections 2.1 and 2.2;

3. Use PIP to copy files from the backup medium to disk.

2.5 ENTERING UIC'S

After the system is installed, and system programs are loaded, the System Manager should enter his user identification codes. This procedure is described in Chapter 6.

CHAPTER 3

MONITOR MODIFICATION

3.1 INTRODUCTION

When a BATCH-11 or DOS-11 Monitor is loaded, it is initialized. That is, when the Monitor is bootstrapped from the system device, an initialization routine is run, which establishes the conditions in which the Monitor will operate. It does this by examining a module in the Monitor, called the Configuration Module. Each Monitor contains a standard version of the Configuration Module, which appears in Appendix E. Other standard components of each Monitor include:

- Resident Monitor (RMON)
- Keyboard command routine
- Device drivers
- EMT routines
- Clock routines
- Transient monitor

The Configuration Module is the means by which the user modifies the system to his special needs. He may add, replace, or delete device drivers, or EMT routines. The Configuration Module is the medium through which he communicates such changes to the Monitor.

The remainder of this Chapter describes the method of modifying the Configuration Module. Adding, replacing and deleting system facilities (drivers, etc.) are described in Chapter 4.

3.2 BUILDING THE CONFIGURATION MODULE

The Configuration Module (CFTBL) is assembled using three source files:

1. CFMAC.MAC A set of macro definitions.

2. SYSTEM.MAC A file that defines the system. SYSTEM.MAC is edited to reflect configuration changes. (Refer to section 3.3.)

3. CONFIG.MAC A file of macro calls and macro directives that create the Configuration Module in object form.

The procedure is:

```
        $RUN MACRO
        MACRO Vxx
        #CFTBL.OBJ,LP:/LI:ME/CRF<CFMAC.MAC,SYSTEM.MAC,CONFIG.MAC
```

NOTE

> For users with 8K configurations, DEC has supplied the MACROS.LDA load module, which is used to assemble these source files. The appropriate command is $RUN MACROS.LDA.

3.3 EDITING SYSTEM.MAC

When a configuration is modified, the source file SYSTEM.MAC must be edited, and the Configuration Module re-assembled to reflect the changes.

Editing can be performed on the following elements in SYSTEM.MAC:

		Symbol	
1.	Default UIC	DFU	
2.	Clocks	FTCLL FTCLP	
3.	Standard devices	FTxx	
4.	User-defined EMT's	UDEMT	Highest EMT number -77$_8$;
		UDEMSW	No. of swappable EMT's;
		UDEMRS	No. of resident EMT's;
		GEMUØ -GEMU5	Macros used to specify EMT's;
		GLOTAB	Definition of globals that cause EMT's to be loaded by the Linker.
5.	User-defined (non-standard) devices	UDDEV	Total of user-written device handlers
		GDVUØ -GDVU2	Macros used to define devices.

The following sections are pertinent to the comments included in the listing of SYSTEM.MAC. Refer to Appendix E.

3.3.1 Default UIC Value

The system default UIC (User Identification Code) is used when no UIC is specified on a Batch job. All files created during processing of the job are put into the default UIC area. The standard default UIC is [1,2]; i.e., project no. 1, user no. 2. To change this (e.g., to [Ø,Ø]), edit the line in SYSTEM.MAC that reads:

```
        DFU     1,2
```

Change the values of 1,2 to those desired, e.g.,

```
        DFU     Ø,Ø
```

NOTE

By specifying Ø,Ø as the default UIC, you prevent Batch jobs from being run if they omit an authorized UIC on the JOB card. The standard default UIC [1,2] is entered by SYSLOD when the BATCH-11 Monitor is loaded.

3.3.2 Clock Routine Definition

The lines

```
FTCLL=1
FTCLP=1
```

define the line frequency clock (KW11-L) and programmable clock (KW11-P), respectively. In its unmodified state, SYSTEM.MAC uses the following rules to determine which clock routine will be loaded:

1. If no clock is present, no clock routine will be loaded.
2. If two clocks are present, the KW11-L clock routine will be loaded.
3. When one clock is present, the corresponding clock routine will be loaded.

To prevent a clock routine from being loaded, even though the corresponding clock is present, delete the appropriate line from SYSTEM.MAC. For example, to prevent the line clock from being loaded, delete the line:

```
FTCLL=1
```

3.3.3 Standard Devices

Standard devices are defined by symbols in the form FTxx where xx corresponds to the device mnemonic. To ignore an existing device, simply delete the corresponding FTxx=1 symbol.

E.g., to omit magnetic tape, delete the line:

```
FTMT=1
```

This might be done to omit a device that is malfunctioning.

3.3.4 User-Written EMT's

User-written EMT's are defined starting with the variable UDEMT=n, where n equals the maximum user-written EMT value, minus 77_8. That is, if six user-written EMT's have been added, the highest of which is EM.154, the value of n is 55_8 (154_8-77_8). When no EMT's have been added, n equals Ø. Note that n can range from 41_8 to 70_8. User EMT's must be in the range 140_8 to 177_8.

The next variable, UDEMSW=n, defines the number of swappable user-written EMT's, i.e., the number of EMT's that are loaded when needed.

The variable UDEMRS=n defines the number of user-written EMT's to be made resident, i.e., EMT's that are loaded at system initialization time and remain in core whether in use or not.

After the three UDEMxx=n variables are defined, specify all user-written EMT's in macros GEMUØ to GEMU5, as follows:

1. Specify swappable EMT's first;
2. After all swappable EMT's are specified, start specifying resident EMT's in a new macro;
3. Specify 13_8 EMT's per macro (or the remainder if less than 13 remain to be specified);
4. EMT specifications must begin with macro GEMUØ, and the sequence GEMUØ, GEMU1...GEMU5 must be followed. No macros may be skipped;
5. Specify EMT's by replacing the line .IRP < > with a line including the user-written EMT numbers; e.g.,

   ```
   .IRP P,<141,142,143,144,146,151,152,153,154,16Ø,161>
   ```

6. User-written EMT's must be added to the CIL. (Refer to Chapter 4, Section 4.2.1).

3.3.5 Non-Standard Devices

The system configuration module can be modified to include non-standard devices as follows:

1. Define variable UDDEV=n, where n equals the number of non-standard devices. User-defined drivers for these devices should be in the Monitor CIL (refer to Chapter 4).

2. Non-standard devices are defined in the macros GDVUØ to GDVU2.

3. Three devices may be defined per macro. To add more than nine, add macros named GDVU3, GDVU4,... etc., in the format of GDVUØ.

4. Macros should be used in sequence, without skipping. When three or more devices remain to be defined, three should be defined in the current macro.

5. A non-standard device definition has the general format:

   ```
   <X,IV>
   ```

   where X = the device mnemonic, and,
   IV = interrupt vector address.

   A full device macro definition has the general format:

   ```
   .IRP P,<<X1,IV1>,<X2,IV2>,<X3,IV3>>
   ```

3.3.6 Configuration Title

To help the user keep track of the configuration his system is using, a title can be assigned in the Configuration Module. This is done by inserting text between the angle brackets (< >) on the line reading:

```
IDENT < >
```

For example, the line could be edited to read

```
IDENT <RK,NO RF>
```

When the Monitor identifies itself it also prints the Configuration Title; e.g.:

```
RK,NO RF
DOS VØ8
$
```

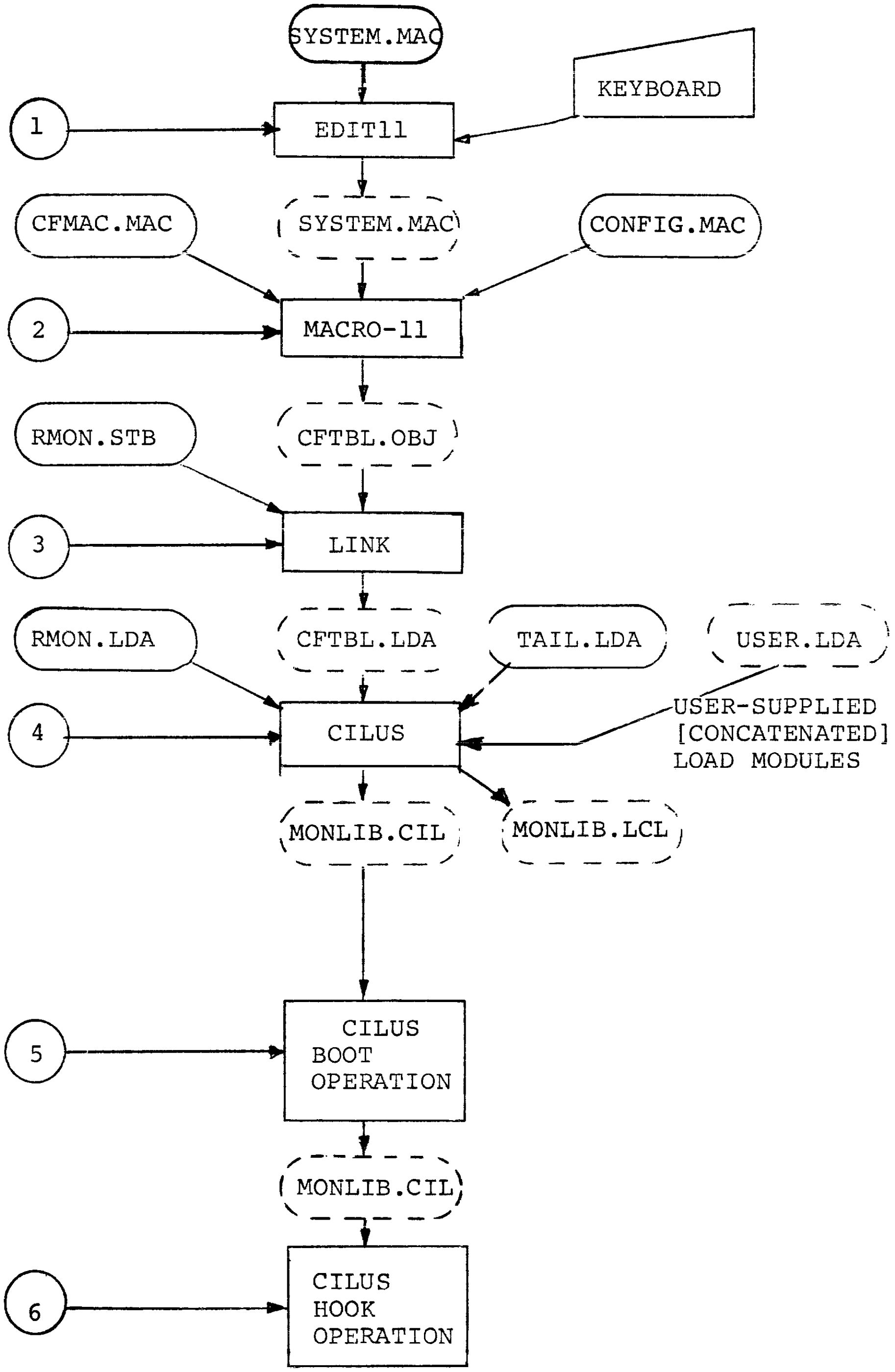
3.3.7 Steps in Modifying the Configuration Module

Figure 3-1 illustrates the flow of procedures used to update the Configuration Module. Each step is numbered. The command strings for each step are shown on the page following Figure 3-1 (section 3.3.8), headed by the corresponding number. If the files needed by step N are available, processing may begin at step N. That is, if the files RMON.LDA, CFTBL.LDA, TAIL.LDA, and USER.LDA are available (steps 1 to 3), processing can start at step 4.

The symbols in Figure 3-1 are interpreted as follows:

Ⓝ	-	An entry point at which processing can begin, if processing for all prior entry points is complete.
AAA	-	A DEC-supplied module.
AAA	-	A user-supplied, or modified module
NAME	-	A BATCH/DOS-11 System Program

Figure 3-1 is formatted to assist in choosing a starting point and to illustrate the order in which input files must be specified.



NOTE: The left-to-right order of input files corresponds to their correct sequence in the command string to the system program.

Figure 3-1 Overview of Monitor Initialization

3.3.8 Monitor Initialization Command Strings

1. Editing SYSTEM.MAC

```
$RUN EDIT
#SYSTEM.MAC<dev:SYSTEM.MAC
*R
(perform edits)
*EX
```

2. Assembly of Configuration Module (CFTBL)

```
$RUN MACRO      ;MACROS.LDA FOR 8K SYSTEMS
#CFTBL,LP:/LI:ME/CRF<CFMAC,SYSTEM,CONFIG
```

3. Linking CFTBL

```
$RUN LINK
#CFTBL,LP:<RMON.STB,CFTBL/B:Ø/E
```

4. Building Monitor CIL

   a. With no user-supplied modules

```
$RUN CILUS
#NEWMON.CIL,LP:,SY:NEWMON.LCL<RMON/BU,CFTBL,TAIL/E
```

   b. With user-supplied modules

```
$RUN CILUS
#NEWMON.CIL,LP:,SY:NEWMON.LCL<RMON/BU,CFTBL,TAIL,USER/E
```

All user-supplied modules should be concatenated in a file named USER.LDA.

5. Booting the Monitor CIL

```
$RUN CILUS
#NEWMON.CIL/BO
```

The HOOK operation should not be performed until the new Monitor is checked out.

NOTE

A FINISH command reinitializes the Monitor, which has the following consequences:

1. If the bootstrap on the system device is not a VØ8 bootstrap, the system crashes. The bootstrap must be a VØ8 bootstrap. Old bootstraps are not compatible.

2. A FINISH command reloads the system that is hooked to the system device.

First initialization procedure - Notes

a. Respond to line frequency question

b. Do not HALT manually until

```
DOS VØ8          BATCH VØ8
$           or   $
```

is printed.

6. Hooking the CIL

The CIL is hooked as follows:

```
$RUN CILUS
#NEWMON.CIL/HOOK/TO:8
```

NOTE

The /TO:8 switch allows the Monitor to be used on machines of 8K core memory and above. This switch should be used when hooking the CIL.

3.4 CAUTIONS

The following points should be noted:

1. The first bootstrapping of a newly-created system should not be interrupted, because the CIL is in the process of being modified. This process occurs between the start of the read, and the Monitor's printout of its identification. This caution does not apply to subsequent bootstrapping of the system.

2. Do not use PIP to transfer CIL's. The recommended procedure is to create a LICIL version of the file, which can then be transferred by PIP.

3. The hardware configuration need not be exactly configured because the initialization routine selects those devices that are defined in the Configuration Module which are also present in the configuration. Thus, DEC cartridges on RK11 controllers can be exchanged between machines, when the configuration has been defined as having all possible peripherals, and the initialization routine will select only those devices actually present.

4. If two clocks are defined and both are installed (KW11-P and KW11-L), the KW11-L will be selected. Otherwise, the existing clock will be selected.

Table 3-1 lists errors that may occur when the system is first loaded. Error messages are printed at the keyboard, with the following additional text:

```
CIL IS WRONG.  CHECK ERROR LIST.  REBUILD CIL
AFTER CORRECTING ERROR, AND TRY AGAIN.
SYSTEM WILL HALT NOW; LOAD A BACKUP SYSTEM.
```

Because the system is not yet loaded, normal error processing cannot be performed.

TABLE 3-1

Initial System Load Errors

ERROR MESSAGE	CAUSE
CILERR	The format of the CIL index is not right. A high-order byte of a binary file was not equal to Ø. Check CIL index and recreate CIL.
SPCERR	Core size is insufficient to initialize a system. Notify a DEC Software Specialist.
CL3ERR	No CIL line with code 3 has been found in the General Information CIL line. Rebuild the CIL.

CLCERR	The CIL lacks a module required by the defined configuration. The module type can be determined by examining R3:
	R3 = 4; EMT module
	R3 = 3; Keyboard module
	R3 = 2; Driver module
	R3 = 1; Clock module
	To determine the total number of missing modules, check the memorv address referenced by the contents of R4 minus 2.

DSZERR	The disk block size does not equal either 100_8 or 400_8, according to the Bootstrap.
CORERR	The nonexistent memory loop did not detect nonexistent memory. Call Field Service.
MRTERR	The format of MRT in the CFTBL file is wrong.
DX1ERR	The format of KB.DX1 (keyboard module index) is wrong.
SPEERR	EMT's defined as special in CFTBL are not known as special by RMON5. May create compatibility problems.
SQZERR	A keyboard module had more than nine overlays.
CSQERR	Sequence verification error. Successive CIL members should be on successive CIL blocks. (The CIL block number should be a monotonic increasing function of the CIL sequence number.)
SEQERR	A keyboard module has its overlays in disarray. Recreate the CIL.

Table 3-1

Initial System Load Errors (Cont.)

ERROR MESSAGE	CAUSE
CFTERR	The module named CF.TBL should be the second module in the CIL. This error results from using an incorrect CILUS command when replacing CF.TBL.
KBCERR	Keyboard module not found in Table DX1.
MODERR	One of the following essential modules is not in the CIL: KBI XIT EM77 KBDX1 KILL TMON
SYSERR	The bootstrap device is not recognized as a valid system device by the initialization code.

The recommended procedure in the event an initialization error occurs, while installing the standard DEC Monitor, is to contact a DEC Software Specialist.

3.5 MONITOR MODULE RESIDENCY NOTES

The following points should be noted when deciding to make Monitor modules resident:

- Any transient module can be made resident. A transient module is any module which uses the main swap buffer, including all modules that service programmed monitor requests. (The exception is RWN., which serves .READ and .WRITE requests, and is always resident.)

- Device drivers should not be made resident, because they are resident from .INIT to .RLSE.

- Keyboard command modules may not be made resident.

- GNM and EDP may not be made resident. GNM is the global name of a subsidiary routine used by the RWN module. The RWN module processes .READ and .WRITE requests, and is permanently resident. EDP is the Error Diagnostic Print routine.

- When a transient module is made resident, any subsidiary routines should also be made resident. Otherwise, swap buffer conflicts may occur at interrupt time. Subsidiary routines are defined in Appendix C of the DOS Monitor Programmer's Reference Manual, DEC-11-OMONA-A-D.

- If a non-reentrant module is made resident, the user must ensure that it is not reentered prematurely, by issuing a .WAIT prior to its subsequent usage. Note that modules that are non-reentrant simply because they have overlays are exceptions, since making them resident (along with their overlays) removes the problem. (The RLS. module should never be made resident, in any case.)

- A reentrant module is one that can be recalled to process a second request before it has finished processing the first request. For example, the BLO module might be called to initiate data transfer, and return immediately to the user. It could then be called again before the first data transfer was finished. A transient routine is not reentrant if either of the following conditions applies:

  1. It calls the routines that get and release buffer space (S.GTB, S.RLB);

  2. It has overlays, or calls subsidiary routines.

  Routines that call S.GTB and S.RLB are summarized in Table 3-2. Subsidiary routines are described in Appendix C of the DOS Monitor Manual. The .INIT processor (INR) has an overlay, but it is automatically loaded if INR is made resident.

Table 3-2

Modules That Call S.GTB and S.RLB

Module	Function
ALO	Allocate
APP	Append
CBA	Allocate contiguous blocks
CKX	Check access privileges
CLS	Common CLOSE Processor
DCN	Delete contiguous files
DEL	Set-up procedure for Delete
DIR	Directory search
DLN	Delete linked files
FCL	Close a file
FCR	Create a file
FOP	Open a file
GMA	Transfer bit-map to/from core
INR	INIT processor
LBA	Allocate blocks to linked files
OPN	Common OPEN Processor
PRO	Set file protection
RNM	Rename a file
RUN	Process RUN EMT

CHAPTER 4

SYSTEM FACILITIES

4.1 DEVICE DRIVERS

The BATCH/DOS-11 system includes standard versions of the following device drivers:

Driver	Device
DV.CR	CR11/CM11 Card Reader/Mark Sense Reader (assembled for ASCII only).
DV.LP	LP11 Line Printer (assembled for 80 columns).
DV.MT	TU1Ø Magnetic Tape (with automatic rewinds).
DV.QT	TU1Ø Magnetic Tape (without automatic rewinds).
DV.DC	RC11 Disk.
DV.DF	RF11 Disk.
DV.DK	High-density RK11 Disk (assumes all units high density) or low-density RK11 Disk (assumes unit Ø low density).
DV.DT	DECtape.
DV.KB	Console Teleprinter.
DV.PR	Paper Tape reader.
DV.PP	Paper Tape punch.

4.1.1 Naming Convention for Device Drivers

The convention established for device driver modules is to use a TITLE statement of the following form:

```
.TITLE  DV.xx
```

where xx is the name of the driver. E.g.,

```
.TITLE  DV.LP
```

is the name of the LP11 Line Printer driver. If the user writes a driver for a non-standard device (e.g., QC), he must observe this convention. That is, the driver (in this case) must have the following title:

```
.TITLE  DV.QC
```

The drivers have been written by DEC to function in certain ways, as defined in the Device Driver write-ups (DEC-11-NIZA-D and supplements). Through the use of optional assembly parameters, the user can alter the manner in which some of these drivers function. Table 4-1 summarizes the drivers that can be modified, and the assembly parameters that apply to each.

Optional assembly parameters are specified for one of two reasons:

1. To reduce the size of the device driver; or,
2. To increase or alter the driver's capabilities.

4.1.2 Assembling Device Drivers

When the assembly parameters have been decided upon, it is necessary to assemble the driver, including the parameters. One method of doing this is to edit the source program, specifying the parameters there. A simpler method is to specify the parameters at the Keyboard, following input of the MACRO command string as shown in the following example:

```
$RUN MACRO
MACRO Vn XX
#CR,LP:/CRF<KB:/PA:1,SY:CR
ONLY29=Ø
↑C
.END (two carriage returns, or carriage return, line feed)
```

ın the preceding example, the user has specified that the Card Reader driver be assembled, with input coming from the keyboard and the file CR.MAC. The keyboard input, which is read only during Pass 1 (/PA:1), follows on the succeeding line:

ONLY29=Ø   indicates that cards are to be read in Ø29 mode only; Ø26 control cards will be ignored.

The ↑C .END sequence signals the end of keyboard input. Input is now read from the system disk, as file name CR.MAC or CR.PAL.

This method of specifying assembly parameters for device drivers is recommended, because it precludes the possibility of errors that may occur in editing the driver source.

TABLE 4-1

DEVICE DRIVER PARAMETERS

Driver	Parameter	Function
DV.LP	LP11=132.	This parameter defines the number of print columns, in decimal. LP11=8Ø. is the default.
DV.CR	BLANKS=Ø	This parameter enables the blank suppress option. When enabled, a blank suppress card causes columns 73-80, and any trailing blanks to be ignored on following data cards, and to be replaced by CR/LF. A second blank suppress card in the deck disables the option. This parameter is included in the standard Card Reader driver distributed by DEC. To eliminate this, the driver must be re-assembled with BLANKS=Ø deleted.
	BINARY=Ø	This parameter enables the reading of binary data, as well as ASCII.

	DEFALT=Ø
	ONLY26=Ø
	ONLY29=Ø
		This group of parameters defines the card codes accepted. Normally, either Ø26 or Ø29 cards may be read, where Ø29 is assumed by default if no Ø26 control card is present. DEFALT reverses this, to make Ø26 the default value. ONLY26 and ONLY29 restrict the driver to one or the other.

	MARKS=Ø	The Card Reader driver, in standard form, will read either 80-column punched cards, or 40-column CM11 Mark Sense cards. Specifying MARKS=Ø restricts the driver to 40-column cards.
DV.DK	LOWDEN=Ø	This parameter specifies that all RK11 drives are low density and causes creation of DV.DKL. (If no parameter is specified, all RK11 drives are assumed to be high density.)

	MIXED=Ø
	CONFIG=N
		These two parameters define a system having RK11 drives of mixed density, some high, some low and causes creation of DV.DKH. The value of N defines those having low density, by matching bits against unit numbers. N=112$_8$ is Ø1ØØ1Ø1Ø$_2$; thus units 1, 3, and 6 are low density (bit Ø being the rightmost bit). These parameters must be used together.

DV.MT	DVRRWD=Ø	This parameter eliminates the rewind function. When defined, no rewinds occur, and the driver name is changed to DV.QT. The QT version is used when DV.QT is specified as the driver name rather than DV.MT. Under normal circumstances, magnetic tape is rewound with each .OPEN and each .CLOSE, which can be time-consuming. It may be useful to have both versions of the driver available, affording a choice based on the most appropriate for the circumstances. Both versions are included in the distributed system.

NOTE

The conditional assembly parameter SYSDV, formerly included for disk drivers, is no longer used.

To change the Line Printer driver (DV.LP) to accommodate 132-column lines, assemble the driver, specifying the following commands:

```
$RUN MACRO
MACRO Vn XX
#LP,LP:/CRF<KB:/PA:1,SY:LP
LP11 = 132.
↑C
.END  (terminate the END command with either two carriage
      returns, or carriage return, line feed).
```

4.1.3 Adding Device Drivers

To add a device driver, the procedure outlined in steps 1 through 5 below must be observed. It is also necessary to observe the device driver naming convention, as defined in section 4.1.1

Steps in adding a driver:

1. Create the driver source(s), with reference to the document PDP-11 Device Driver Package (DEC-11-NIZB-D).

2. Assemble the edited source module(s), e.g.,

```
$RUN MACRO
MACRO Vxx
#DVR1,LP:<DTØ:DVR1     (the source program must
#DVR2,LP:<DTØ:DVR2     be correctly titled.  I.e.,
                       DV.DT, for a DECtape driver.)
```

3. Link the assembled object module(s), as:

```
$RUN LINK
LINK Vxx
#DVR1,LP:<DVR1/B:Ø/E (a bottom value of Ø is required)
#DVR2,LP:<DVR2/B:Ø/E
```

4. Use PIP to concatenate the driver modules (optional):

```
$RUN PIP
PIP Vxx
#USER.LDA<DVR1.LDA,DVR2.LDA
```

5. Update the Monitor CIL, as:

```
$RUN CILUS
CILUS Vxx
#NEWMON/BOOT,,NEWMON.LCL<MONLIB.LCL,USER.LDA/E
*I  DT:DVR1,DVR2  (I is the Insert command)
*E
```

The two USER.LDA modules (DVR1,DVR2) are inserted into the Monitor CIL ahead of the DECtape driver module (DT).

NOTE

> To determine the order of modules in the Monitor, run CILUS, specifying the following command string:
>
> ```
> #LP:<NEWMON.CIL/LIST
> ```

Note that the output CIL has been named NEWMON. This name is assigned to differentiate this version of the Monitor CIL from the currently checked out version (MONLIB.CIL). The new version should be thoroughly tested before it is named MONLIB.CIL and hooked to the bootstrap. (Any user-created name may be assigned; NEWMON is merely an example.)

4.1.4 Deleting Drivers

To delete a device driver, run CILUS, specifying the Delete command. For example, to delete the RF11 disk driver:

```
$RUN CILUS
CILUS Vxx
#NEWMON.CIL/BO,LP:,SY:NEWMON.LCL<DT1:MONLIB.LCL/E
*D DV.DF    (D is the Delete command)
*E
```

The Configuration Module must be edited and re-assembled to reflect the modified configuration. Refer to section 3.3.3, and Figure 3-1.

4.1.5 Replacing Drivers

To replace a device driver, observe the following procedures:

1. Assemble and link the replacement driver.
2. Create a concatenated file of load modules (e.g., USER) containing the new driver.
3. Run CILUS, specifying the R edit command:

   ```
   $RUN CILUS
   CILUS Vxx
   #NEWMON.CIL,,NEWMON.LCL<MONLIB.LCL,USER.LDA/E
   *R DV.XX:DV.XX   (R is the Replace command)
   *E
   ```

   The version of DV.XX in the Monitor CIL is replaced by the version of DV.XX from USER.LDA.

4.2 EMT MODULES

4.2.1 Adding EMT's

User-written EMT's can be added as follows:

1. Refer to the DOS Monitor System Programmer's Manual, (DEC-11-MNDA-D) for details on writing EMT routines.

2. Create the source module, observing the EMT naming convention in the .TITLE directive; viz:

```
.TITLE   EM.NNN
```

   where NNN is a value from 14Ø$_8$ to 167$_8$.

3. Use PIP to concatenate user-written EMT's into a file named USER.LDA.

4. Update the Configuration Module (refer to section 3.3.4, and Figure 3-1).

5. Insert the EMT into the Monitor CIL:

```
$RUN CILUS
CILUS VØØ4A
#NEWMON,,NEWMON.LCL<RMON,USER.LDA/E
#I  EM.14Ø
*E
```

In this example, the module EM.14Ø is appended to the Resident Monitor (RMON).

4.2.2 Deleting EMT's

EMT's must not be deleted, since this causes initialization errors.

4.2.3 Replacing EMT's

To replace an EMT:

1. Create the source module, using the same name as that of the version to be replaced;

2. Assemble and link the modified module;

3. Run CILUS, using the Replace command to edit the Monitor CIL:

```
$RUN CILUS
CILUS Vxx
#NEWMON/BO,LP:,SY:NEWMON.LCL<DT1:MONLIB.LCL,USER.LDA/E

*R EM.14Ø:EM.14Ø   (The current EM.14Ø is replaced by the
                   backup version, also named EM.14Ø, in
*E                 secondary input dataset USER.LDA.)
```

NOTE

Some EMT's are built from overlays. This must be considered when EMT's are replaced.

CHAPTER 5

BATCH-11 CONSIDERATIONS

5.1 INTRODUCTION

The following sections provide information pertinent to managing a system that includes the Batch-11 Monitor. It may be necessary to modify standard Batch-11 parameters to fit a particular installation. For example, it may be desirable to output listings to a default device other than the line printer when using Batch concise commands. Or, it may be useful to modify the actions taken when a Batch job terminates, or to change the default User Identification Code to limit accessibility to the system. Methods for implementing these changes are described in this Chapter.

5.2 CONCISE COMMAND PROCEDURE FILES

The Batch Command Language contains a set of commands called concise commands (refer to section 1.2 of the PDP-11 Batch User's Guide). Concise commands function by invoking associated procedures, which expand the commands to perform the appropriate processing (e.g., compile, link, and execute a FORTRAN program). The expansion of the command depends on how much information has been supplied, in the form of input and output datasets. Certain of these commands provide defaults, allowing input and/or output datasets to be omitted. Default datasets are supplied by the BATCH system.

The default output datasets for certain commands are summarized in Table 5-1.

TABLE 5-1

CONCISE COMMAND DEFAULT OUTPUT DEVICES

Command	Dataset	Default Output Device
$EXECUTE	Listing	Line Printer
$FORTRN	Listing	Line Printer
$LINK	Load map	Line Printer
$MACRO	Listing	Line Printer
$DIR	Output	Line Printer
$LIST	Output	Line Printer

To change the default dataset (or to eliminate it altogether), the Concise Command procedure files must be edited. The source to be edited is named CMDS.MAC.

CMDS.MAC has in it the global symbol DFLIST, which defines the default listing device. The default listing dataset is used only when the appropriate dataset specification is absent in one of the Concise Commands. A specified listing dataset overrides the default. Thus,

```
        $DIR TO KB:
```

will list the directory on the teleprinter, regardless of the default listing dataset specification. In standard form, DFLIST consists of

```
        DFLIST:
               .BYTE  3         ;BYTE COUNT - Ø=NO DEFAULT
               .ASCII 'LP:'     ;SYSTEM DEFAULT DATASET
```

As specified, the octal number associated with the .BYTE directive tells the system to examine the next three characters to determine the default listing dataset. In standard form, this is the line printer (LP:). This can be changed by editing the ASCII string to the desired value, and updating the byte count as needed. E.g.:

```
        DFLIST:
               .BYTE  3
               .ASCII 'KB:'
```

In this example, the keyboard becomes the default listing device. No adjustment to the byte count is needed.

To eliminate the default listing device, change the byte count to Ø; i.e.,

```
        .BYTE  Ø
```

The ASCII string following is now ignored.

NOTE

When the DFLIST byte count is set to Ø, the MACRO control line must be edited to remove the /CR switch.

5.2.1 Assembling and Linking Edited Procedure Files

After a CMDS.MAC is edited, it must be re-assembled and relinked as part of the BATCH-11 Transient Monitor (TMON). Assembly is accomplished by running MACRO-11, with the following command string specified:

```
#CMDS,listing<CMDS
```

The relinking process is performed by running the Link-11 Linker, with the following command string specified:

```
#TMON,load map<TMON,LKP,BATMAN,CCL,CMDS/B:Ø/E
```

The input files are:

TMON	-	Current Transient Monitor
LKP	-	File Structured Lookup (EMT 46)
BATMAN	-	BATCH MANAGER
CCL	-	Concise Command Language Processor
CMDS	-	Command Procedure Files

For these command strings, all input and output files are assumed to be on the system disk.

CILUS is now used to replace TMON in the Monitor CIL, as in the following example:

```
$RUN CILUS
CILUS Vxx    (CILUS identifies itself)
#NEWMON.CIL/BO,LP:,SY:NEWMON.LCL<MONLIB.LCL,TMON/E
*R  TM:TM
*E
```

Note that in the example, the output CIL is named NEWMON.CIL to prevent its replacing the Monitor CIL currently on the system device. It is booted, but not hooked, since it still must be tested. Only after checking its performance and being satisfied that it runs properly should the user hook it to the bootstrap.

5.3 JOB TERMINATION PROCESSING

Jobs are normally terminated when a $FINISH card (statement) is detected in the batch stream, though in some cases termination is effected by a fatal error, or by detection of the $JOB statement defining the start of the next job. When a job terminates, the Batch operating system performs certain tasks which may be classified as "clean-up" tasks. These tasks include:

1. Deletion of files in the default UIC area (if the current UIC matches the default UIC).
2. Output of the error dump, if present;
3. Printing the default log, if used; or,
4. Any combination of the above.

The Batch system invokes termination processing by means of a $CHANGE command, with a file specified according to the job's requirements; i.e., one file is specified if only default UIC processing is appropriate, a different file is specified if default UIC and dump processing are required, etc. The files summarized in Table 5-2 are invoked only when a job is terminated by a $FINISH statement. A default file (CLØ.SYS) may be invoked when termination occurs under the following conditions: the default UIC was used, and no $FINISH command was included.

TABLE 5-2

SYSTEM FILES FOR JOB TERMINATION

File Name	Function
CL1.SYS	Processes default UIC.
CL2.SYS	Processes error dumps.
CL3.SYS	Processes default UIC and error dumps.
CL4.SYS	Processes default log.
CL5.SYS	Processes default UIC and default log.
CL6.SYS	Processes default log and error dumps.
CL7.SYS	Processes error dumps, default UIC, and default log.

These files consist of commands which call the system programs (PIP or LDUMP) that perform the needed functions. All these files must reside in the [1,1] area on the system disk.

5.3.1 CL1.SYS

```
$;           CL1.SYS IS USED TO DELETE FILES FROM THE DEFAULT
$;           UIC DISK AREA.
$RUN PIP
#*.*/DE
```

5.3.2 CL2.SYS

```
$;           CL2.SYS IS USED TO FORMAT AND PRINT DUMPS.
$GET LDUMP
$AS  BI:,6
$BEGIN
LP:<DMP.SYS[1,1]
$KILL
```

NOTE: The command string to LDUMP does not begin with a # character.

5.3.3 CL3.SYS

```
$;           CL3.SYS IS USED TO DELETE FILES FROM THE DEFAULT
$;           UIC DISK AREA, AND TO FORMAT AND PRINT DUMPS.
$GET LDUMP


$AS BI:,6
$BEGIN
LP:<DMP.SYS[1,1]
$KILL
$RUN PIP
#*.*/DE
```

5.3.4 CL4.SYS

```
$;
$;                 CL4.SYS IS USED TO PRINT THE DEFAULT LOG.
$RUN PIP
#LP:<CMO.SYS
```

5.3.5 CL5.SYS

```
$;                 CL5.SYS IS USED TO DELETE FILES FROM THE DEFAULT
$;                 UIC DISK AREA, AND TO PRINT THE DEFAULT LOG.
$RUN PIP
#LP:<CMO.SYS
#*.*/DE
```

5.3.6 CL6.SYS

```
$;
$;
$GET LDUMP
$AS BI:,6
$BEGIN
LP:<DMP.SYS[1,1]
$KILL
$RUN PIP
#LP:<CMO.SYS
```

CL6.SYS IS USED TO FORMAT AND PRINT DUMPS, AND TO PRINT THE DEFAULT LOG.

5.3.7 CL7.SYS

```
$;
$;
$GET LDUMP
$AS BI:,6
$BEGIN
LP:<DMP.SYS[1,1]
$KILL
$RUN PIP
#LP:<CMO.SYS
#*.*/DE
```

CL7.SYS IS USED TO DELETE FILES FROM THE DEFAULT UIC DISK AREA, TO FORMAT AND PRINT DUMPS, AND TO PRINT THE DEFAULT LOG.

5.3.8 Default File (CLØ.SYS)

In the event that the $FINISH statement was omitted, or the appropriate file cannot be found, the only termination processing done is to delete all files in the default UIC area, if necessary. A file named CLØ.SYS normally duplicates CL1.SYS, and resides in the [1,1] area of the system disk.

CLØ.SYS consists of:

```
$RUN PIP
#*.*/DE
```

NOTE

Comment lines are not allowed in CLØ.SYS. That is, a line such as

```
;THIS IS CLØ.SYS
```

is not permitted.

5.3.9 Editing Job Termination Files

It will be noted that all job termination files that involve dumps or the default log, specify LP (line printer) as the listing device. For installations not having a line printer, these files must be edited to specify the appropriate device, such as the teleprinter. Using CL5.SYS as an example, the edited version might be:

```
$;            CL5.SYS IS USED TO DELETE FILE FROM THE DEFAULT
$;            UIC DISK AREA, AND TO PRINT THE DEFAULT LOG.
$RUN PIP
#KB:<CMO.SYS
#*.*/DE
```

The same type of editing would be done to the output-on-error dump. To put an error dump on DECtapeØ, for example, CL2.SYS would be altered to:

```
$;            CL2.SYS IS USED TO FORMAT AND PRINT DUMPS
$GET LDUMP
$AS BI:,6
$BEGIN
DTØ:DMP<DMP.SYS[1,1]
$KILL
```

5.4 MODIFYING THE DEFAULT UIC

The standard default UIC is [1,2]. This UIC is used when no UIC is specified in a $JOB statement. To prevent unauthorized users from running Batch jobs, or to ensure that jobs are run only when properly identified, the default UIC can be changed to [Ø,Ø]. This forces users to provide a UIC that is currently entered (see Chapter 6, "Entering UIC's"), since a job associated with a UIC of [Ø,Ø] will not be run.

To modify the default UIC, the Configuration Module must be edited, in its source form, re-assembled and relinked. Refer to Chapter 3, section 3.3.1.

CHAPTER 6

PIP SYSTEM MANAGEMENT FACILITIES

6.1 PIP FUNCTIONS

The System Manager is able to use the PIP File Utility Package to perform the following tasks:

1. Manage User Identification Codes (UIC's);
2. Manage files on disk and/or DECtapes;
3. Initialize devices.

The System Manager operates in restricted mode; i.e., he is permitted to use the system in ways not allowed to other users. To enter restricted mode:

1. Log in under UIC 1,1

```
        $LO 1,1
```

2. Run PIP

```
        $RUN PIP
        PIP Vxx
        #          (enter the appropriate command string)
```

Because the operating system does not have built-in hardware/software protection, certain potentially dangerous functions have been isolated from the general user and are available in PIP in a restricted mode. That is, the user must be logged under UIC 1,1. Thus, the UIC 1,1 should be restricted for the system manager and persons whom he deems trustworthy.

6.2 MANAGING UIC'S

UIC stands for User Identification Code. In order to operate under the BATCH/DOS-11 Monitor, the user must log in, and identify himself to the system. He does this by entering his UIC:

```
    $LO xxx,yyy
```

where

xxx = an octal number representing the project identification; and,

yyy = an octal number representing the programmer's identification.

WARNING

> Ø through 1Ø and 377 are not legal project or programmer identifications and should not be assigned to the general user.

NOTE

> Under the BATCH Monitor, a Batch job can run under the default UIC; i.e., a $JOB card will be accepted without a UIC. However, a UIC *must* be included when logging in at the keyboard.

Since it is the System Manager's responsibility to control the manner in which the system is used, the following capabilities are available to him:

1. Listing UIC's;
2. Entering UIC's onto the system disk; and,
3. Withdrawing UIC's.

6.2.1 Listing UIC's (/UIC)

A listing of all UIC's on a device is obtained by specifying a command string in the following general format:

```
#dev:/UIC    where dev: comprises the appropriate
             device specifier.
```

For example, to list all UIC's on RK11 unit Ø, enter:

```
#DK:/UIC
```

SAMPLE LIST

```
DKØ:

[  1,1  ]
[234,234]
[ 35,1  ]
[14Ø,252]
[  1,2  ]
[2Ø1,2Ø1]
[2ØØ,2ØØ]
```

6.2.2 Entering UIC's (/EN)

The /EN switch is used to enter UIC's into the Master File Directory (MFD) of a device. One or more UIC's can be entered, as shown in the following example:

```
#DK:[2ØØ,2Ø1],[151,36]/EN
```

As the result of the above command strings, project 2ØØ, user 2Ø1; and project 151, user 36, are included as authorized UIC's in RK11 unit Ø's MFD.

In the absence of an explicit UIC, the UIC of the current user is entered. I.e.,

```
#DK:/EN
```

causes UIC 1,1 to be entered (since the System Manager logs in under 1,1).

6.2.3 Withdrawing UIC's (/WD)

The System Manager may wish to withdraw a UIC, and its associated files, from the MFD of a device. He does this by using the /WD switch, as shown:

Example 1.

```
#DKØ:[27,311]/WD
```

Project number 27, programmer 311 is now no longer a recognized UIC on RK11 unit Ø. All files are deleted under 27,311, and that space is now free to be used for other user's files.

Example 2.

```
#DK:[2ØØ,2ØØ],[55,67]/WD
```

The UIC's [2ØØ,2ØØ], and [55,67] are removed from the MFD of RK11 unit Ø. All files associated with these UIC's are also deleted from the disk.

Example 3.

```
#DK1:[2ØØ,*]/WD
```

This example shows the use of the asterisk (*) to indicate "all". The UIC specification is interpreted as user group 2ØØ, all users. PIP withdraws all UIC's in user group 2ØØ from the MFD on RK11 unit 1.

6.3 FILE MANAGEMENT

System Manager file management functions include:

1. Listing file directories;
2. Deleting files;
3. Transferring files from one device to another; and,
4. Superseding files.

6.3.1 Listing File Directories (/DI)

The PIP /DI (Directory) switch is used to list file directories. Directories can be listed in a number of ways, as illustrated in the following examples.

Example 1.

```
#LP:<DK:[*,*]/DI
```

All directories on RK11 unit Ø are listed on the line printer.

NOTE

> Although the line printer (LP:) is specified as the listing device in these examples, the Directory can be listed on any device capable of accepting output; keyboard (KB:), DECtape (DT:), etc.

Example 2.

```
#LP:<DK:[2ØØ,*]/DI
```

A listing is produced of all directories in user group 2ØØ, on RK11 unit Ø.

Example 3.

```
#LP:<DK:*.MAC[*,*]/DI
```

A directory of all file names with extension .MAC. on RK11 unit Ø, is produced.

Example 4.

```
#LP:<FORTRN.*[27,27]/DI
```

A directory listing of all files named FORTRN (all extensions) in the 27,27 UIC area is produced.

6.3.2 Deleting the Files (/DE)

The System Manager can delete files belonging to specific users, or he can delete more than one user's files at a time, by specifying various values in the UIC field of the command string. This is shown in the following examples.

Example 1. Deleting all files, with a specified extension.

```
#DKØ:*.FTN[*,*]/DE
```

All files with extension .FTN, under all UIC's on the RK11 unit Ø are deleted.

Example 2. Deleting a specific file

```
#DT1:PGM.MAC[2ØØ,2ØØ]/DE
```

The file named PGM.MAC, belonging to user 2ØØ, project 2ØØ in the MFD of DECtape unit 1 is deleted.

6.3.3 Transferring Files

The general method of transferring files is described in the PIP manual. The System Manager has the added capability of transferring files across UIC's as illustrated in the following examples.

Example 1.

```
#DK:[*,*]< DK1:*.LDA[*,*]
```

All files with extension .LDA are transferred from RK11 unit 1 to RK11 unit Ø. Any UIC's on RK11 unit 1 that are not already in RK11 unit Ø's MFD are automatically entered.

Example 2.

```
#DK1:[*,*]<DF:*.*[277,*]
```

All files belonging to all users in project 277 are transferred from the RF11 disk to RK11 unit 1. UIC's within project 277 that are not already in RK11 unit 1's MFD are entered automatically.

Example 3. Transferring files to System Manager's area

```
#DK:< DK1:*.*[*,*]
```

By omitting a UIC specification on the output side of the command string, the System Manager forces all files to be transferred from RK11 unit 1 to RK11 unit Ø, in area [1,1].

6.3.4 Superseding Files

The method of superseding files on a device is described in the PIP manual (DEC-11-UPUPA-A-D).

6.4 DEVICE INITIALIZATION

Devices that can be initialized are DECtape, magnetic tape, and RK11 disk. Initialization is accomplished by means of the Zero switch (/ZE). Refer to the PIP manual for details.

APPENDICES

APPENDIX A

SYSTEM GENERATION BATCH STREAMS

A.1 BUILDING A DISTRIBUTION KIT

The process of creating a distribution kit which contains the BATCH/DOS-11 system programs is accomplished by the following steps:

1. Assemble the system programs, putting them on an RK11 disk in object form;

2. Use PIP to transfer the object programs from the RK11 disk to the distribution medium.

A.1.1 Assembling System Programs

The following BATCH-11 command sequence causes the system programs to be assembled. Each $CH command refers to the assembly commands for a particular program:

```
$JOB CSPASM[1,1]
$CH DK1:LINK.ASM
$CH DK1:LIBR.ASM
$CH DK1:EDIT.ASM
$CH DK1:CREF.ASM
$CH DK1:PIP.ASM
$CH DK1:MACRO.ASM
$CH DK1:ODT.ASM
$CH DK1:FILDMP.ASM
$CH DK1:CILUS,ASM
```

The individual batch streams that assemble each program are:

LINK.ASM

```
$RU MACRO
#DK1:LINK<DK1:LINK
#DK1:LINKOB<DK1:OVRBLD,LINK
```

LIBR.ASM

```
$RU MACRO
#DK1:LIBR<DK1:LIBR
```

EDIT.ASM

```
$RU MACRO
#DK1:EDIT<DK1:EDIT
```

CREF.ASM

```
$RU MACRO
#DK1:CREF<DK1:CREF
```

PIP.ASM

```
$RU MACRO
#DK1:SPIPEX.COR<DK1:SYMDEF,SPIPEX
#DK1:PIPEX.COR<DK1:SYMDEF,PIPEX
#DK1:TRNFER.COR<DK1:SYMDEF,TRNFER
#DK1:GLSUBS.COR<DK1:SYMDEF,GLSUBS
#DK1:ALLOC.COR<DK1:SYMDEF,ALLOC
#DK1:ENTER.COR<DK1:SYMDEF,ENTER
#DK1:ZERO.COR<DK1:SYMDEF,ZERO
#DK1:RENAME.COR<DK1:SYMDEF,RENAME
#DK1:UNLOC.COR<DK1:SYMDEF,UNLOC
#DK1:FASTMT.COR<DK1:SYMDEF,FASTMT
#DK1:PROTEC.COR<DK1:SYMDEF,PROTEC
#DK1:DIREC.COR<DK1:SYMDEF,DIREC
#DK1:DELETE.COR<DK1:SYMDEF,DELETE
#DK1:PIPEX.OBJ<DK1:OLAY,SYMDEF,PIPEX
#DK1:TRNFER.OBJ<DK1:OLAY,SYMDEF,TRNFER
#DK1:GLSUBS.OBJ<DK1:OLAY,SYMDEF,GLSUBS
#DK1:ALLOC.OBJ<DK1:OLAY,SYMDEF,ALLOC
#DK1:ENTER.OBJ<DK1:OLAY,SYMDEF,ENTER
#DK1:ZERO.OBJ<DK1:OLAY,SYMDEF,ZERO
#DK1:RENAME.OBJ<DK1:OLAY,SYMDEF,RENAME
#DK1:UNLOC.OBJ<DK1:OLAY,SYMDEF,UNLOC
#DK1:FASTMT.OBJ<DK1:OLAY,SYMDEF,FASTMT
#DK1:PROTEC.OBJ<DK1:OLAY,SYMDEF,PROTEC
#DK1:DIREC.OBJ<DK1:OLAY,SYMDEF,DIREC
#DK1:DELETE.OBJ<DK1:OLAY,SYMDEF,DELETE
$RU PIP
#DK1:PIP.OBJ,PIPOVØ.OBJ,PIPOV1.OBJ,PIPOV2.OBJ/DE
#DK1:PIPOV3.OBJ,TEMP1,TEMP2/DE
#DK1:PIP.OBJ<DK1:PIPEX.OBJ,GLSUBS.OBJ
#DK1:PIPOVØ.OBJ<DK1:TRNFER.OBJ,ENTER.OBJ
#DK1:PIPOV1.OBJ<DK1:DIREC.OBJ
#DK1:TEMP1<DK1:ALLOC.OBJ,ZERO.OBJ,DELETE.OBJ
#DK1:TEMP2<DK1:UNLOC.OBJ,RENAME.OBJ,PROTEC.OBJ
#DK1:PIPOV2.OBJ<DK1:TEMP1,TEMP2
#DK1:PIPOV3.OBJ<DK1:FASTMT.OBJ
#DK1:TEMP1,TEMP2/DE
```

MACRO.ASM

```
$RU MACRO
#DK1:PST<DK1:PST
#DK1:PSTP<DK1:MINI,PST
#DK1:MACRO<DK1:MACRO
#DK1:MACROP<DK1:MINI,MACRO
#DK1:MACOVR<DK1:MACOVR
```

ODT.ASM

```
$RU MACRO
#DK1:ODT11R<DK1:ODT
```

FILDMP.ASM

```
$RU MACRO
#DK1:FILDMP<DK1:FILDMP
```

CILUS.ASM

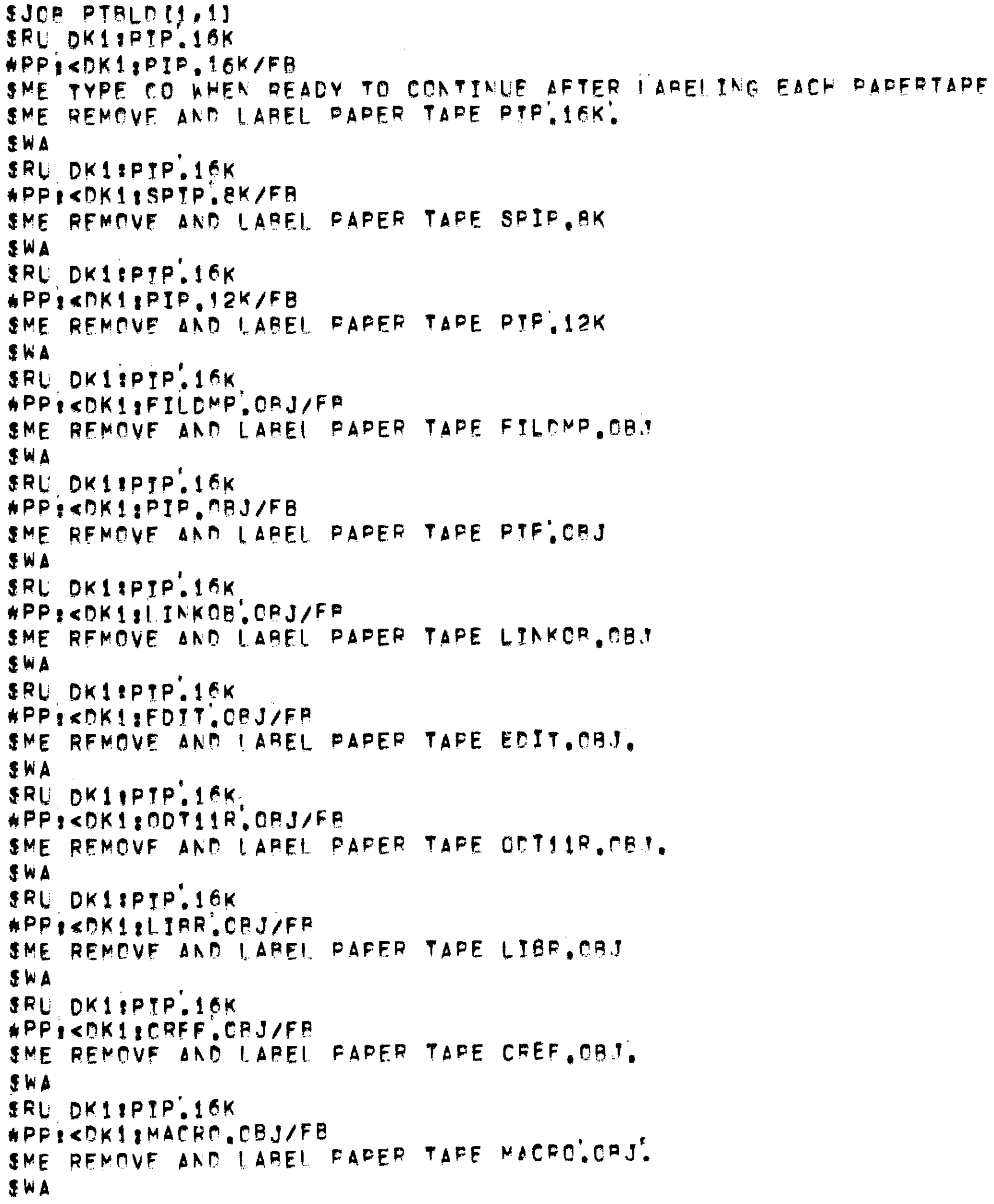
```
$RU MACRO
#DK1:MAIN.OBJ<DK1:COMGLB,MAIN
#DK1:CILIO.OBJ<DK1:COMGLB,CILIO
#DK1:COMDEC.OBJ<DK1:COMGLB,COMDEC
#DK1:EDCOM.OBJ<DK1:COMGLB,EDCOM
#DK1:EDITCL,OBJ<DK1:COMGLB,EDITCL
#DK1:LOAD,OBJ<DK1:COMGLB,LOAD
#DK1:HOKBOT.OBJ<DK1:COMGLB,HOKBOT
#DK1:RFBOOT,OBJ<DK1:RFBOOT
#DK1:RKBOOT,OBJ<DK1:RKBOOT
#DK1:RCBOOT,OBJ<DK1:RCBOOT
#DK1:RPBOOT.OBJ<DK1:RPBOOT
#DK1:TCBOOT,OBJ<DK1:TCBOOT
#DK1:MTBOOT,OBJ<DK1:MTBOOT
$RU PIP
#DK1:CILUS.OBJ<DK1:MAIN,OBJ,CILIO.OBJ
#DK1:CILUS0,OBJ<DK1:COMDEC.OBJ
#DK1:CILUS1,OBJ<DK1:EDCOM,OBJ
#DK1:CILUS2,OBJ<DK1:EDITCL.OBJ
#DK1:TEMP1.OBJ<DK1:LOAD.OBJ,HOKBOT.OBJ,RFBOOT.OBJ,RKBOOT.OBJ
#DK1:TEMP2.OBJ<DK1:RCBOOT.OBJ,RPBOOT.OBJ,TCBOOT.OBJ,MTBOOT.OBJ
#DK1:CILUS3.OBJ<DK1:TEMP1.OBJ,TEMP2.OBJ
#DK1:TEMP1.OBJ,TEMP2.OBJ/DE
```

A.1.2 Putting System Programs on Distribution Media

Three different Batch streams are used by DEC to put system programs on the various distribution media:

1. PTBLD.DIS - Paper tape distribution;
2. DTBLD.DIS - DECtape distribution; and,
3. MTBLD.DIS - Magnetic tape distribution.

A.1.2.1 Paper Tape Build Batch Stream



```
$JOB PTBLD[1,1]
$RU DK1:PIP.16K
#PP:<DK1:PIP.16K/FB
$ME TYPE CO WHEN READY TO CONTINUE AFTER LABELING EACH PAPERTAPE
$ME REMOVE AND LABEL PAPER TAPE PIP.16K.
$WA
$RU DK1:PIP.16K
#PP:<DK1:SPIP.8K/FB
$ME REMOVE AND LABEL PAPER TAPE SPIP.8K
$WA
$RU DK1:PIP.16K
#PP:<DK1:PIP.12K/FB
$ME REMOVE AND LABEL PAPER TAPE PIP.12K
$WA
$RU DK1:PIP.16K
#PP:<DK1:FILDMP.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE FILDMP.OBJ
$WA
$RU DK1:PIP.16K
#PP:<DK1:PIP.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE PIP.OBJ
$WA
$RU DK1:PIP.16K
#PP:<DK1:LINKOB.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE LINKOB.OBJ
$WA
$RU DK1:PIP.16K
#PP:<DK1:EDIT.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE EDIT.OBJ.
$WA
$RU DK1:PIP.16K
#PP:<DK1:ODT11R.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE ODT11R.OBJ.
$WA
$RU DK1:PIP.16K
#PP:<DK1:LIBR.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE LIBR.OBJ
$WA
$RU DK1:PIP.16K
#PP:<DK1:CREF.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE CREF.OBJ.
$WA
$RU DK1:PIP.16K
#PP:<DK1:MACRO.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE MACRO.OBJ.
$WA
```

```
$RU DK1:PIP.16K
#PP:<DK1:MACROP.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE MACROP.OBJ.
$WA
$RU DK1:PIP.16K
#PP:<DK1:MACOVR.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE MACOVR.OBJ
$WA
$RU DK1:PIP.16K
#PP:<DK1:PST.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE PST.OBJ
$WA
$RU DK1:PIP.16K
#PP:<DK1:PSTP.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE PSTP.OBJ
$WA
$RU DK1:PIP.16K
#PP:<DK1:MACROS.LDA/FB
$ME REMOVE AND LABEL PAPER TAPE MACROS.LDA
$WA
$RU DK1:PIP.16K
#PP:<DK1:PIPOV0.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE PIPOV0.OBJ
$WA
$RU DK1:PIP.16K
#PP:<DK1:PIPOV1.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE PIPOV1.OBJ
$WA
$RU DK1:PIP.16K
#PP:<DK1:PIPOV2.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE PIPOV2.OBJ
$WA
$RU DK1:PIP.16K
#PP:<DK1:PIPOV3.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE PIPOV3.OBJ
$WA
$RU DK1:PIP.16K
#PP:<DK1:LINK.8K/FB
$ME REMOVE AND LABEL PAPER TAPE LINK.8K
$WA
$RU DK1:PIP.16K
#PP:<DK1:LINKOB.8K/FB
$ME REMOVE AND LABEL PAPER TAPE LINKOB.8K
$WA
$RU DK1:PIP.16K
#PP:<DK1:LINK.12K/FB
$ME REMOVE AND LABEL PAPER TAPE LINK.12K
$WA
$RU DK1:PIP.16K
#PP:<DK1:LINKOB.12K/FB
$ME REMOVE AND LABEL PAPER TAPE LINKOB.12K
$WA
$RU DK1:PIP.16K
#PP:<DK1:LINK.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE LINK.OBJ
$WA
$RU DK1:PIP.16K
#PP:<DK1:CILUS.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE CILUS.OBJ
$WA
$RU DK1:PIP.16K
#PP:<DK1:CILUS0.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE CILUS0.OBJ
$WA
```

```
$RU DK1:PIP.16K
#PP:<DK1:CILUS1.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE CILUS1.OBJ
$WA
$RU DK1:PIP.16K
#PP:<DK1:CILUS2.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE CILUS2.OBJ
$WA
$RU DK1:PIP.16K
#PP:<DK1:CILUS3.OBJ/FB
$ME REMOVE AND LABEL PAPER TAPE CILUS3.OBJ
$WA
$RU DK1:PIP.16K
#PP:<DK1:PST.MAC/FA
$ME REMOVE AND LABEL PAPER TAPE PST.MAC
$WA
$RU DK1:PIP.16K
#PP:<DK1:SYSMAC.SML/FA
$ME REMOVE AND LABEL PAPER TAPE SYSMAC.SML
$WA
$RU DK1:PIP.16K
#PP:<DK1:GOAP16.DOS/FA
$ME REMOVE AND LABEL PAPER TAPE GOAP16.DOS
$WA
$RU DK1:PIP.16K
#PP:<DK1:GOAP.16K/FA
$ME REMOVE AND LABEL PAPER TAPE GOAP.16K
$WA
$RU DK1:PIP.16K
#PP:<DK1:GOAP.ALL/FA
$ME REMOVE AND LABEL PAPER TAPE GOAP.ALL
$WA
$RU DK1:PIP.16K
#PP:<DK1:GOTA.16L
$ME REMOVE AND LABEL PAPER TAPE GOTA.16L
$WA
$ME ALL OF THE PDP-11 SYSTEM PROGRAMS HAVE NOW BEEN
$ME WRITTEN ONTO PAPER TAPES.
$FI
```

A.1.2.2 DECtape Build Batch Stream

```
$JOB DTBLD[1,1]
$ME MOUNT SCRATCH DECTAPES ON
$ME UNITS Ø AND 1.SET REMOTE
$ME AND WRITE ENABLE SWITCHES ON BOTH.
$ME TYPE CO WHEN READY TO CONTINUE.
$WA
$RU DK1:PIP.16K
#DTØ:/ZF
#DTØ:<DK1:PIP.16K/FB,PIP.12K
#DTØ:<DK1:SPIP.8K/FB,EDIT.OBJ,ODT11R.OBJ,LIBR.OBJ,FILCMP.OBJ,CREF.C
#DTØ:<DK1:MACRO.OBJ,MACROP.OBJ,MACOVR.OBJ,PST.OBJ,PSTP.OBJ/FB
#DTØ:<DK1:MACROS.LDA,PIP.OBJ,PIPOVØ.OBJ,PIPOV1.OBJ,PIPOV2.OBJ/FB
#DTØ:<DK1:PIPOV3.OBJ,LINK.8K/FB
#DTØ:<DK1:LINKOB.8K/FB,LINK.12K,LINKOB.12K,LINK.OBJ,LINKOB.OBJ
#LP:<DTØ:/DI
#DT1:/ZF
#DT1:<DK1:CILUS.OBJ,CILUSØ.OBJ,CILUS1.OBJ
#DT1:<DK1:CILUS2.OBJ,CILUS3.OBJ
#DT1:<DK1:PST.MAC/FA,SYSMAC.SML,GOTA16.DOS
#DT1:<DK1:GOTA.16K/FA,GOTA.ALL,GOTA.16L,GOTA16.DO
#LP:<DT1:/DI
$ME REMOVE DECTAPES FROM UNITS Ø AND 1 AND LABEL THEM.
$WA
$ME ALL OF THE PDP-11 SYSTEM PROGRAMS HAVE NOW BEEN
$ME WRITTEN ON DECTAPES
$FI
```

A.1.2.3 Magnetic Tape Build Batch Stream

```
$JOB MTBLD[1,1]
$ME MOUNT A MAGNETIC TAPE ON UNIT=Ø,WRITE ENABLED.
$ME TYPE "CO" WHEN READY



$WA
$RU DK1:PIP.16K
*QTØ:GOAM16.DOS<DK1:GOAM16.DOS/FA
*QTØ:GOAM.16K<DK1:GOAM.16K/FA
*QTØ:GOAM.ALL<DK1:GOAM.ALL/FA
*QTØ:GOTA.16L<DK1:GOTA.16L/FA
*QTØ:PIP.16K<DK1:PIP.16K/FB
*QTØ:<DK1:PIP.12K/FB
*QTØ:<DK1:SPIP.8K/FB
*QTØ:EDIT.OBJ<DK1:EDIT.OBJ/FB
*QTØ:ODT11R.OBJ<DK1:ODT11R.OBJ/FB
*QTØ:LIBR.OBJ<DK1:LIBR.OBJ/FB
*QTØ:FILDMP.OBJ<DK1:FILDMP.OBJ/FB
*QTØ:CREF.OBJ<DK1:CREF.OBJ/FB
*QTØ:MACRO.OBJ<DK1:MACRO.OBJ/FB
*QTØ:MACROP.OBJ<DK1:MACROP.OBJ/FB
*QTØ:MACOVR.OBJ<DK1:MACOVR.OBJ/FB
*QTØ:PST.OBJ<DK1:PST.OBJ/FB
*QTØ:PSTP.OBJ<DK1:PSTP.OBJ/FB
*QTØ:MACROS.LDA<DK1:MACROS.LDA/FB
*QTØ:PIP.OBJ<DK1:PIP.OBJ/FB
*QTØ:PIPOVØ.OBJ<DK1:PIPOVØ.OBJ/FB
*QTØ:PIPOV1.OBJ<DK1:PIPOV1.OBJ/FB
*QTØ:PIPOV2.OBJ<DK1:PIPOV2.OBJ/FB
*QTØ:PIPOV3.OBJ<DK1:PIPOV3.OBJ/FB
*QTØ:LINK.8K<DK1:LINK.8K/FB
*QTØ:LINKOB.8K<DK1:LINKOB.8K/FB
*QTØ:LINK.12K<DK1:LINK.12K/FB
*QTØ:LINKOB.12K<DK1:LINKOB.12K/FB
*QTØ:LINK.OBJ<DK1:LINK.OBJ/FB
*QTØ:LINKOB.OBJ<DK1:LINKOB.OBJ/FB
*QTØ:CILUS.OBJ<DK1:CILUS.OBJ/FB
*QTØ:CILUSØ.OBJ<DK1:CILUSØ.OBJ/FB
*QTØ:CILUS1.OBJ<DK1:CILUS1.OBJ/FB
*QTØ:CILUS2.OBJ<DK1:CILUS2.OBJ/FB
*QTØ:CILUS3.OBJ<DK1:CILUS3.OBJ/FB
*QTØ:PST.MAC<DK1:PST.MAC/FA
*QTØ:SYSMAC.SML<DK1:SYSMAC.SML/FA
*LP:<MTØ:/DI
*MTØ:/RW
$ME REMOVE MAGNETICTAPE FROM UNIT=Ø AND LABEL THE MAGNETIC TAPE
$WA
$FI
```

A.2 GETTING ON THE AIR BATCH STREAMS

Recall that the BATCH-11 user builds the system programs on his system device by specifying the following commands:

```
1.                $RUN ⎧PR: ⎫ PIP.16K
                       ⎨DTn:⎬
                       ⎩MTn:⎭
```

2a. Paper Tape

```
#SY:GOAP16.DOS<PR:/FA/SU     GOAP16.DOS tape in reader
#SY:GOAP.16K<PR:/FA/SU       GOAP.16K tape in reader
#SY:GOAP.ALL<PR:/FA/SU       GOAP.ALL tape in reader
#SY:GOTA.16L<PR:/FA/SU       GOTA.16L tape in reader
```

2b. DECtape

```
#SY:<DT1:GOTA16.DOS/FA/SU
#SY:<DT1:GOTA.16K/FA/SU
#SY:<DT1:GOTA.ALL/FA/SU
#SY:<DT1:GOTA.16L/FA/SU
  Go to Step 3
```

2c. Magnetic Tape

```
#SY:<MTØ:GOAM16.DOS/FA/SU
#SY:<MTØ:GOAM.16K/FA/SU
#SY:<MTØ:GOAM.ALL/FA/SU
#SY:<MTØ:GOTA.16L/FA/SU
  Go to Step 3
```

3.

```
#↑C
.KI
        ⎧GOAP16.DOS⎫   specify appropriate
$BA     ⎨GOTA16.DOS⎬   batch stream
        ⎩GOAM16.DOS⎭
```

The batch streams transferred to the system disk, and subsequently read, are listed in the following sections.

A.2.1.1 GOAP16.DOS

```
$JOB GOAP16[1,1]
$CH GOAP.16K
$CH GOAP.ALL
$CH GOTA.16L
$RU PIP
#GOAP.*,GOTA.*/DE
#ODT11R.TMP/RE<ODT11R.OBJ
#*.OBJ/DE
#ODT.OBJ/RE<ODT11R.TMP
#*.8K,*.12K,*.TMP/DE
#[*,*]/DI
$ME POSITION CLØ.SYS IN READER
$WA
#SY:CLØ.SYS<PR:/FA
```

```
$ME POSITION CL1.SYS IN READER
$WA
#SY:CL1.SYS<PR:/FA
$ME POSITION CL2.SYS IN READER
$WA
#SY:CL2.SYS<PR:/FA
$ME POSITION CL3.SYS IN READER
$WA
#SY:CL3.SYS<PR:/FA
$ME POSITION CL4.SYS IN READER
$WA
#SY:CL4.SYS<PR:/FA
$ME POSITION CL5.SYS IN READER
$WA
#SY:CL5.SYS<PR:/FA
$ME POSITION CL6.SYS IN READER
$WA
#SY:CL6.SYS<PR:/FA
$ME POSITION CL7.SYS IN READER
$WA
#SY:CL7.SYS<PR:/FA
$ME POSITION LDUMP.LDA IN READER
$WA
#SY:LDUMP.LDA<PR:/FB
$FINISH
```

A.2.1.2 GOAP.16K

```
$ME AFTER POSITIONING EACH PAPER TAPE IN THE READER
$ME TYPE CO WHEN READY TO CONTINUE.
$ME POSITION PIP.16K IN PAPER TAPE READER
$WA
$GET PR:
$SA PIP.16K
$ME POSITION SYSMAC.SML IN PAPER TAPE READER.
$WA
$RU PIP.16K
#SYSMAC.SML<PR:/FA/SU
$ME POSITION MACRO.OBJ IN PAPER TAPE READER.
$WA
$RU PIP.16K
#MACRO.OBJ<PR:/FB/SU
$ME POSITION PST.OBJ IN PAPER TAPE READER.
$WA
$RU PIP.16K
#PST.OBJ<PR:/FB/SU
$ME POSITION LINKOB.12K IN PAPER TAPE READER.
$WA
$RU PIP.16K
#LINKOB.12K<PR:/FB/SU
$ME POSITION LINK.12K IN PAPER TAPE READER
$WA
$RU PIP.16K
#LINK.12K<PR:/FB/SU
$ME POSITION LINKOB.OBJ IN PAPER TAPE READER
$WA
$RU PIP.16K
#LINKOB.OBJ<PR:/FB/SU
$ME POSITION LINK.OBJ IN PAPER TAPE READER
$WA
$RU PIP.16K
#LINK.OBJ<PR:/FB/SU
```

A.2.1.3 GOAP.ALL

```
$ME AFTER POSITIONING EACH PAPER TAPE IN THE READER
$ME TYPE CO WHEN READY TO CONTINUE.
$ME POSITION EDIT.OBJ IN PAPER TAPE READER
$WA
$RU PIP.16K
#EDIT.OBJ<PR:/FB/SU
$ME POSITION ODT11R.OBJ IN PAPER TAPE READER.
$WA
$RU PIP.16K
#ODT11R.OBJ<PR:/FB/SU
$ME POSITION LIBR.OBJ IN PAPER TAPE READER.
$WA
$RU PIP.16K
#LIBR.OBJ<PR:/FB/SU
$ME POSITION FILDMP.OBJ IN PAPER TAPE READER.
$WA
$RU PIP.16K
#FILDMP.OBJ<PR:/FB/SU
$ME POSITION CREF.OBJ IN PAPER TAPE READER.
$WA
$RU PIP.16K
#CREF.OBJ<PR:/FB/SU
$ME POSITION PIP.OBJ IN PAPER TAPE READER
$WA
$RU PIP.16K
#PIP.OBJ<PR:/FB/SU
$ME POSITION PIPOVØ.OBJ IN PAPER TAPE READER.
$WA
$RU PIP.16K
#PIPOVØ.OBJ<PR:/FB/SU
$ME POSITION PIPOV1.OBJ IN PAPER TAPE READER.
$WA
$RU PIP.16K
#PIPOV1.OBJ<PR:/FB/SU
$ME POSITION PIPOV2.OBJ IN PAPER TAPE READER.
$WA
$RU PIP.16K
#PIPOV2.OBJ<PR:/FB/SU
$ME POSITION PIPOV3.OBJ IN PAPER TAPE READER.
$WA
$RU PIP.16K
#PIPOV3.OBJ<PR:/FB/SU
$ME POSITION CILUS.OBJ IN PAPER TAPE READER.
$WA
$RU PIP.16K
#CILUS.OBJ<PR:/FB/SU
$ME POSITION CILUSØ.OBJ IN PAPER TAPE READER.
$WA
$RU PIP.16K
#CILUSØ.OBJ<PR:/FB/SU
$ME POSITION CILUS1.OBJ IN PAPER TAPE READER.
$WA
$RU PIP.16K
#CILUS1.OBJ<PR:/FB/SU
$ME POSITION CILUS2.OBJ IN PAPER TAPE READER.
$WA
$RU PIP.16K
#CILUS2.OBJ<PR:/FB/SU
$ME POSITION CILUS3.OBJ IN PAPER TAPE READER.
$WA
$RU PIP.16K
#CILUS3.OBJ<PR:/FB/SU
```

A.2.1.4 GOTA.16L

NOTE

GOTA.16L is the same for all media.

```
$RUN LINKOB.12K
$RUN LINK.12K
#LINK.LDA<LINK.OBJ/E
#LINKOB.LDA<LINKOB.OBJ/E
$RU LINKOB.LDA
$RU LINK.LDA
#MACRO<MACRO,PST/E
#EDIT<EDIT/E
#LIBR<LIBR/E
#FILDMP<FILDMP/T:37460/E
#CREF<CREF/E
#PIP<PIP.OBJ/CC/OV:4/E
#PIPOVØ<PIPOVØ.OBJ/CC/E
#PIPOV1<PIPOV1.OBJ/E
#PIPOV2<PIPOV2.OBJ/CC/E
#PIPOV3<PIPOV3.OBJ/E
#CILUS<CILUS.OBJ/CC/OV:4/E
#CILUSØ.OVR<CILUSØ.OBJ/E
#CILUS1.OVR<CILUS1.OBJ/E
#CILUS2.OVR<CILUS2.OBJ/E
#CILUS3.OVR<CILUS3.OBJ/CC/E
```

A.2.2 DECtape Batch Streams

A.2.2.1 GOTA16.DOS

```
$JOB GOTA16[1,1]
$ME MOUNT DEC-11-OCSPA-A-UC1 ON DTØ: WRITE LOCKED
$ME MOUNT DEC-11-OCSPA-A-UC2 ON DT1: WRITE LOCKED
$ME TYPE "CO" WHEN READY
$WA
$CH GOTA.16K
$CH GOTA.ALL
$CH GOTA.16L
$RU PIP
#GOTA.*/DE
#ODT11R.TMP/RE<ODT11R.OBJ
#*.OBJ/DE
#ODT.OBJ/RE<ODT11R.TMP
#*.8K,*.12K/DE
#[*,*]/DI
$ME MOUNT DEC-11-OBMUA-A-UC ON DTØ: WRITE LOCK
$WA
#SY:<DTØ:*.SYS/FA
#SY:<DTØ:LDUMP.LDA
$ME ALL DONE
$FINISH
```

A.2.2.2 GOTA.16K

```
$RU DTØ:PIP.16K
#SY:<DT1:SYSMAC.SML/FA/SU
#SY:<DTØ:MACRO.OBJ/FB/SU
#SY:<DTØ:PST.OBJ/FB/SU
#SY:<DTØ:LINKOB.12K/FB/SU
#SY:<DTØ:LINK.12K/FB/SU
#SY:<DTØ:LINKOB.OBJ/FB/SU
#SY:<DTØ:LINK.OBJ/FB/SU
```

A.2.2.3 GOTA.ALL

```
$RU DTØ:PIP.16K
#SY:<DTØ:EDIT.OBJ/FB/SU
#SY:<DTØ:ODT11R.OBJ/FB/SU
#SY:<DTØ:LIBR.OBJ/FB/SU
#SY:<DTØ:FILDMP.OBJ/FB/SU
#SY:<DTØ:CREF.OBJ/FB/SU
#SY:<DTØ:PIP.OBJ/FB/SU
#SY:<DTØ:PIPOVØ.OBJ/FB/SU
#SY:<DTØ:PIPOV1.OBJ/FB/SU
#SY:<DTØ:PIPOV2.OBJ/FB/SU
#SY:<DTØ:PIPOV3.OBJ/FB/SU
#SY:<DT1:CILUS.OBJ/FB/SU
#SY:<DT1:CILUSØ.OBJ/FB/SU
#SY:<DT1:CILUS1.OBJ/FB/SU
#SY:<DT1:CILUS2.OBJ/FB/SU
#SY:<DT1:CILUS3.OBJ/FB/SU
```

A.2.2.4 GOTA.16L

```
$RUN LINKOB.12K
$RUN LINK.12K
#LINK.LDA<LINK.OBJ/E
#LINKOB.LDA<LINKOB.OBJ/E
$RU LINKOB.LDA
$RU LINK.LDA
#MACRO<MACRO,PST/E
#EDIT<EDIT/E
#LIBR<LIBR/E
#FILDMP<FILDMP/T:3/460/E
#CREF<CREF/E
#PIP<PIP.OBJ/CC/OV:4/E
#PIPOVØ<PIPOVØ.OBJ/CC/E
#PIPOV1<PIPOV1.OBJ/E
#PIPOV2<PIPOV2.OBJ/CC/E
#PIPOV3<PIPOV3.OBJ/E
#CILUS<CILUS.OBJ/CC/OV:4/E
#CILUSØ.OVR<CILUSØ.OBJ/E
#CILUS1.OVR<CILUS1.OBJ/E
#CILUS2.OVR<CILUS2.OBJ/E
#CILUS3.OVR<CILUS3.OBJ/CC/E
```

A.2.3 Magnetic Tape Batch Streams

A.2.3.1 GOAM16.DOS

```
$JOB GOAM16[1,1]
$CM GOAM.16K
$CM GOAM.ALL
$CM GOTA.16L
$RU PIP
#GOAM.*,GOTA.*/DE
#ODT11R.TMP/RE<ODT11R.OBJ
#*.OBJ/DE
#ODT.OBJ/RE<ODT11R.TMP
#*.8K,*.12K,*.TMP/DE
#[*,*]/DI
```

A.2.3.2 GOAM.16K

```
$RU MT0:PIP,16K
#SYSMAC.SML<MT0:SYSMAC.SML/FA/SU
#MACRO.OBJ<MT0:MACRO.OBJ/FB/SU
#PST.OBJ<MT0:PST.OBJ/FB/SU
#LINKOB.12K<MT0:LINKOB.12K/FB/SU
#LINK.12K<MT0:LINK.12K/FB/SU
#LINKOB.OBJ<MT0:LINKOB.OBJ/FB/SU
#LINK.OBJ<MT0:LINK.OBJ/FB/SU
```

A.2.3.3 GOAM.ALL

```
$RU MT0:PIP,16K
#EDIT.OBJ<MT0:EDIT.OBJ/FB/SU
#ODT11R.OBJ<MT0:ODT11R.OBJ/FB/SU
#LIBR.OBJ<MT0:LIBR.OBJ/FB/SU
#FILDMP.OBJ<MT0:FILDMP.OBJ/FB/SU
#CREF.OBJ<MT0:CREF.OBJ/FB/SU
#PIP.OBJ<MT0:PIP.OBJ/FB/SU
#PIPOV0.OBJ<MT0:PIPOV0.OBJ/FB/SU
#PIPOV1.OBJ<MT0:PIPOV1.OBJ/FB/SU
#PIPOV2.OBJ<MT0:PIPOV2.OBJ/FB/SU
#PIPOV3.OBJ<MT0:PIPOV3.OBJ/FB/SU
#CILUS.OBJ<MT0:CILUS.OBJ/FB/SU
#CILUS0.OBJ<MT0:CILUS0.OBJ/FB/SU
#CILUS1.OBJ<MT0:CILUS1.OBJ/FB/SU
#CILUS2.OBJ<MT0:CILUS2.OBJ/FB/SU
#CILUS3.OBJ<MT0:CILUS3.OBJ/FB/SU
```

A.2.3.4 GOTA.16L

```
$RUN LINKOB.12K
$RUN LINK.12K
#LINK.LDA<LINK.OBJ/E
#LINKOB.LDA<LINKOB.OBJ/E
$RU LINKOB.LDA
$RU LINK.LDA
#MACRO<MACRO,PST/E
#EDIT<EDIT/E
#LIBR<LIBR/E
#FILDMP<FILDMP/T:3/400/E
#CREF<CREF/E
#PIP<PIP.OBJ/CC/OV:4/E
#PIPOVØ<PIPOVØ.OBJ/CC/E
#PIPOV1<PIPOV1.OBJ/E
#PIPOV2<PIPOV2.OBJ/CC/E
#PIPOV3<PIPOV3.OBJ/E
#CILUS<CILUS.OBJ/CC/OV:4/E
#CILUSØ,OVR<CILUSØ.OBJ/E
#CILUS1,OVR<CILUS1.OBJ/E
#CILUS2,OVR<CILUS2.OBJ/E
#CILUS3,OVR<CILUS3.OBJ/CC/E
```

A.3 BUILDING SYSTEM PROGRAMS

The process of completely assembling and linking system programs is accomplished by the batch stream CUSP.BLD, shown below. Note that the assembly procedures have been discussed in section A.1.1. Sections A.3.1 and A.3.2 describe the batch streams that link and test the system programs. It should also be noted that the batch streams used by DEC assume that the batch stream device is an RK11 disk, mounted on unit 1. This specification should be modified by the user, if necessary, to accommodate his choice of batch stream device.

```
$JOB CSPBLD[1,1]
$CM DK1:LINK.ASM
$CM DK1:LIBR.ASM
$CM DK1:EDIT.ASM
$CM DK1:CREF.ASM
$CM DK1:PIP,ASM
$CM DK1:MACRO.ASM
$CM DK1:ODT.ASM
$CM DK1:FILDMP.ASM
$CM DK1:CILUS.ASM
$CM DK1:MACRO.LNK
$CM DK1:LIBR.LNK
$CM DK1:EDIT.LNK
$CM DK1:CREF.LNK
$CM DK1:PIP.LNK
$CM DK1:LINK.LNK
$CM DK1:FILDMP.LNK
$CM DK1:CILUS.LNK
$CM DK1:LIST.DIR
```

To link system programs, specify the following $JOB statement to BATCH:

```
$JOB CSPLNK[1,1]
```

This causes the following batch stream to be read:

```
$JOB CSPLNK[1,1]
$CA DK1:MACRO.LNK
$CA DK1:LIBR.LNK
$CA DK1:EDIT.LNK
$CA DK1:CREF.LNK
$CA DK1:PIP.LNK
$CA DK1:LINK.LNK
$CA DK1:FILDMP.LNK
$CA DK1:CILUS.LNK
$CA DK1:LIST.DIR
```

A.3.1.1 MACRO.LNK

```
$RU LINK
#DK1:MACRO,LP:<DK1:MACRO,PST/E
#DK1:MACROP,LP:<DK1:MACROP,MACOVR,PSTP/T:37460/E
#DK1:MACROO,LP:<DK1:MACRO,MACOVR,PST/T:57460/E
```

A.3.1.2 LIBR.LNK

```
$RU LINK
#DK1:LIBR,LP:<DK1:LIBR/E
```

A.3.1.3 EDIT.LNK

```
$RU LINK
#DK1:EDIT,LP:<DK1:EDIT/E
```

A.3.1.4 CREF.LNK

```
$RU LINK
#DK1:CREF,LP:<DK1:CREF/E
```

A.3.1.5 PIP.LNK

```
$RU LINK
#DK1:SPIP.8K,LP:<DK1:SPIPEX.COR,TRNFER.COR,GLSUBS.COR
#DK1:ENTER.COR,DELETE.COR,DIREC.COR/T:37460/E
#DK1:PIP.16K,LP:<DK1:PIPEX.COR,TRNFER.COR,GLSUBS.COR,ALLOC.COR
#DK1:ENTER.COR,ZERO.COR,RENAME.COR,UNLOC.COR,FASTMT.COR
#DK1:PROTEC.COR,DELETE.COR
#DK1:DIREC.COR/T:77460/E
#DK1:PIP.12K,LP:<DK1:PIPEX.COR,TRNFER.COR,GLSUBS.COR,ALLOC.COR
#DK1:ENTER.COR,ZERO.COR,RENAME.COR,UNLOC.COR,FASTMT.COR
#DK1:PROTEC.COR,DELETE.COR
#DK1:DIREC.COR/T:57460/E
#DK1:PIP,LP:<DK1:PIP.OBJ/CC/OV:4/E
#DK1:PIPOVØ,LP:<DK1:PIPOVØ.OBJ/CC/E
#DK1:PIPOV1,LP:<DK1:PIPOV1.OBJ/E
#DK1:PIPOV2,LP:<DK1:PIPOV2.OBJ/CC/E
#DK1:PIPOV3,LP:<DK1:PIPOV3.OBJ/E
```

A.3.1.6 LINK.LNK

```
$RU LINK
#DK1:LINK,LP:<DK1:LINK/E
#DK1:LINKOB,LP:<DK1:LINKOB/E
#DK1:LINK.8K,LP:<DK1:LINK/T:37460/E
#DK1:LINKOB,8K,LP:<DK1:LINKOB/T:37460/E
#DK1:LINK.12K,LP:<DK1:LINK/T:57460/E
#DK1:LINKOB,12K,LP:<DK1:LINKOB/T:57460/E
```

A.3.1.7 FILDMP.LNK

```
$RU LINK
#DK1:FILDMP,LP:<DK1:FILDMP/T:37460/E
```

A.3.1.8 CILUS.LNK

```
$RU LINK
#DK1:CILUS,LP:<DK1:CILUS.OBJ/CC/OV:4/E
#DK1:CILUSØ.OVR,LP:<DK1:CILUSØ.OBJ/E
#DK1:CILUS1.OVR,LP:<DK1:CILUS1.OBJ/E
#DK1:CILUS2.OVR,LP:<DK1:CILUS2.OBJ/E
#DK1:CILUS3.OVR,LP:<DK1:CILUS3.OBJ/CC/E
```

APPENDIX B

SYSTEM CORE REQUIREMENTS

B.1 MONITOR CORE SIZE ALGORITHM

The amount of core occupied by a BATCH-11 or DOS-11 Monitor at a given time is determined by the needs of the system at that time. Thus, this amount fluctuates depending on the circumstances of the moment. To determine the core requirements of the system, add the core sizes of the following items (all values are given in decimal):

1. Resident Monitor (refer to section B.2);
2. One driver for each device that has been INITed, but not RLSEd. (NOTE: Do not include the driver for the system device. This is part of the Resident Monitor's requirements);
3. One buffer for each open dataset (refer to section B.2.4);
4. One dataset data block (DDB) for each open dataset - 16 words;
5. One file information block (FIB) for each open file (a file is a dataset on a file-structured device) - 16 words;
6. One bit map for each unit on which a file is open for output (64 words);
7. One auxiliary buffer, during operations like OPEN, RUN, etc. (refer to Appendix C of the DOS Reference Manual) - 256 words.

The size actually required by the Monitor is likely to be greater than that derived by simply following the algorithm. The reason is that memory allocated in the middle of the Monitor, for FIB's, DDB's, etc. is not freed when the associated dataset is released; thus, this space is not available for the DDB or buffer of a dataset subsequently INIT'ed. To conserve space for user programs, it is recommended that datasets that will be released first, be INIT'ed last. (Buffer space is allocated at the top of the Monitor, as datasets are INIT'ed; this space is accessible when the dataset is CLOSE'd.)

Typically, a standard DOS-11 Monitor requires from about 2142 to 2213 words permanently resident, depending on the system disk used (see B.2.1). A BATCH-11 Monitor requires approximately 500 words more. Minimum sizes of the Resident Monitors are:

DOS-11	BATCH-11
RC11 = 2142	2642
RF11 = 2145	2645
RK11 = 2206	2706

A "typical" installation (e.g., DECtape, keyboard, line printer) will require approximately 13ØØ more words, in addition to the Monitor, for device drivers and buffers.

B.2 MODULE SIZES

The following sections define the number of words occupied by the Resident Monitor, the Temporary Monitor, transient modules, and device drivers.

B.2.1 Resident Monitor

The number of words needed for the Resident Monitor depends on the system device, and the number of user-defined resident EMT's, and non-standard devices (VAR).

System Device		Monitor Size (Decimal)
RK11	=	1710 + VAR
RF11	=	1649 + VAR
RC11	=	1646 + VAR

VAR = (NPHYDEV + UDDEV) * 4 + UDEMT + SRESEMT

where:

NPHYDEV = total number of standard devices defined for, and attached to, the system;

UDDEV = total number of user-defined devices;

UDEMT = highest EMT value defined by the user (in decimal), minus 63 (decimal); when none are defined, UDEMT=Ø;

SRESEMT = sum of the sizes of all EMT's made permanently resident.

For a standard configuration, VAR is less than or equal to 44.

The modules comprising the Resident Monitor, and their respective sizes, are:

Module	Size (Decimal)	
RMN1	412	
RMN2	204	
RMN3	257	
RWN	552	
RMN5	135 + VAR	
CLOCK	13	
DK	137	Only one may be resident, depending on which disk is the system device.
DF	76	
DC	73	

B.2.2 Temporary Monitor

TMON (for BATCH-11) = 38Ø4

TMON (for DOS-11) = 2816

B.2.3 Transient Modules

Module	Size (Decimal)
ALO	251
DLN	256
FCR	238
FCL	256
DCN	254
DEL	256
LUK	232
CBA	256
LBA	170
DIR	215
FOP	250
CKX	203
GMA	143
APP	252
AP2	168
REN	212
PRO	158
MTO	254
CSX	255
REC	251
INR	428
RLS	183
TRA	177
BLO	143
XIT	100
SPC	46
STT	30
OPN	313
CLS	227
GUT	118
CSM	256
CVT	204
LDR	622
LD2	193
SPO	46
DMP	124
RUN	255
CDT	235
BSF	147

B.2.4 Device Drivers

Driver	Size	Buffer Size
DK (low density)	128	256
(high density)	138	256
DF	77	64
DC	74	64
MT	455	256
QT	483	256
KB	600	32
LP	152	48
DT	190	256
CR (ASCII only)	251	48
PR	67	32
PP	78	32

B.3 STACK SIZE

The most stack space required by the BATCH-11 or DOS-11 Monitor occurs during OPEN processing; 56 words are needed at this time. In addition, 10 words of stack space must be allowed for each possible interrupt (a maximum of 4). Thus, 100 words of core should be reserved, to provide an ample safety margin.

APPENDIX C

MONITOR EMT MODULES

MODULE	NUMBER	GLOBAL NAME
(RES)	0	
WAIT	1 (RES)	N/A
WRITE	2 (RWN)	RWN
(RES)	3 (NYA)	N/A
READ	4 (RWN)	RWN
(RES)	5	N/A
INIT	6	INR
RELEASE	7	RLS
TRAN	10	TRA
BLOCK	11	BLO
SPECIAL FCTNS	12	SPC
STATUS	13	STT
DIRECTORY SRCH	14	DIR
ALLOCATE	15	ALO
OPEN	16	OPN
CLOSE	17	CLS
RENAME	20	REN
DELETE	21	DEL
APPEND	22	APP
(RES)	23	
PROTECT	24	PRO
RECORD I/O	25	REC
SPARES	26,27	
KSB OVERLAY	30 (RES)	N/A
KBL	31	N/A
EDP	32	N/A
KBI	33	N/A
GNM	34	N/A
BCL	35	N/A
SPARES	36,37	
MSB OVERLAY	40 (RES)	N/A
GENERAL UTILITIES	41	GUT
CONVERSION	42	CVT
FILE OPEN	43	FOP
FILE CREATE	44	FCR
FILE CLOSE	45	FCL
FILE LOOKUP	46	LUK
BLOCK ALLOCATE	47	LBA
GET BIT MAP	50	GMA
CONTIGUOUS ALLOCATOR	51	CBA
CHECK ACCESS	52	CKX
DELETE LINKED	53	DLN
DELETE CONTIGUOUS	54	DCN
APPEND 2	55	AP2
COMMAND SYNTAX	56	CSX
COMMAND SEMANTICS	57	CSM
EXIT	60	XIT
PROGRAM LOADER	61	LDR
LOADER 2	62	LD2
MAG. TAPE OPEN	63	MTO
BINARY DUMP	64	DMP
RUN	65	RUN
ASCII DATE/TIME	66	CDT
BATCH STREAM FLUSH	67	BSF

APPENDIX D

SYSTEM DISTRIBUTION KITS

D.1 PAPER TAPE KIT

The Paper Tape distribution kit contains the following paper tapes (not including FORTRAN tapes.)

BATCH/DOS VØ8 KIT - FOR PAPERTAPE

CONTENTS	TAPE
SYSLOD.LDA	DEC-11-OSLDA-A-PL
DOS:	
MONLIB.LCL	DEC-11-ODMLA-A-PO1 to PO5
RMON.LDA	DEC-11-ODRMA-A-PL
TAIL.LDA	DEC-11-ODTLA-A-PL1 to PL5
RMON.STB	DEC-11-ODSTA-A-PL
CONFIG.MAC	DEC-11-ODCOA-A-PA
CFMAC.MAC	DEC-11-ODCFA-A-PA
SYSTEM.MAC	DEC-11-ODSYA-A-PA
BATCH:	
MONLIB.LCL	DEC-11-OBMLA-A-PO1 to PO5
RMON.LDA	DEC-11-OBRMA-A-PL
TAIL.LDA	DEC-11-OBTLA-A-PL1 to PL5
RMON.STB	DEC-11-OBSTA-A-PL
CONFIG.MAC	DEC-11-OBCOA-A-PA
CFMAC.MAC	DEC-11-OBCFA-A-PA
SYSTEM.MAC	DEC-11-OBSYA-A-PA
CLØ.SYS thru CL7.SYS	DEC-11-OBCLA-A-PA1 thru PA8
LDUMP.LDA	DEC-11-OBLDA-A-PL
CMDS.MAC	DEC-11-OBCMA-A-PA
TMON.OBJ	DEC-11-OBTMA-A-PR
CCL.OBJ	DEC-11-OBCCA-A-PR
BATMAN.OBJ	DEC-11-OBBMA-A-PR
LKP.OBJ	DEC-11-OBLKA-A-PR

CONTENTS	LABEL
SYSTEM PROGRAMS:	
EDIT	DEC-11-UEDTA-A-PR
ODT11R	DEC-11-UODTA-A-PR
LIBR	DEC-11-ULIBA-A-PR
FILDMP	DEC-11-UFDPA-A-PR
CREF	DEC-11-LCREA-A-PR
MACRO:	
MACRO.OBJ	DEC-11-LMCOA-A-PR
MACRO.OBJ	DEC-11-LMOPA-A-PR
MACROVR.OBJ	DEC-11-LMOVA-A-PR
PST.OBJ	DEC-11-LMSTA-A-PR
PSTP.OBJ	DEC-11-LMTPA-A-PR
PST.MAC	DEC-11-LMPSA-A-PA
SYSMAC.SML	DEC-11-LMSYA-A-PA
MACROS.LDA	DEC-11-LMOSA-A-PL
CILUS:	
CILUS.OBJ	DEC-11-UCILA-A-PR1
CILUSØ.OBJ	DEC-11-UCILA-A-PR2
CILUS1.OBJ	DEC-11-UCILA-A-PR3
CILUS2.OBJ	DEC-11-UCILA-A-PR4
CILUS3.OBJ	DEC-11-UCILA-A-PR5
LINK:	
LINK.12K	DEC-11-UL12A-A-PL1
LINKOB.12K	DEC-11-UL12A-A-PL2
LINK.8K	DEC-11-UL8KA-A-PL1
LINKOB.8K	DEC-11-UL8KA-A-PL2
LINK.OBJ	DEC-11-ULNKA-A-PR1
LINKOB.OBJ	DEC-11-ULNKA-A-PR2
PIP:	
PIP.16K	DEC-11-UP16A-A-PL
PIP.12K	DEC-11-UP12A-A-PL
SPIP.8K	DEC-11-UP8KA-A-PL
PIP.OBJ	DEC-11-UPIPA-A-PR1
PIPOVØ.OBJ	DEC-11-UPIPA-A-PR2
PIPOV1.OBJ	DEC-11-UPIPA-A-PR3
PIPOV2.OBJ	DEC-11-UPIPA-A-PR4
PIPOV3.OBJ	DEC-11-UPIPA-A-PR5

DEVICE DRIVER SOURCES:

CONTENTS	LABEL
DC.MAC	DEC-11-ODDCA-A-PA
DF.MAC	DEC-11-ODDFA-A-PA
DK.MAC	DEC-11-ODDKA-A-PA
CR.MAC	DEC-11-ODCRA-A-PA
MT.MAC	DEC-11-ODMTA-A-PA
KB.MAC	DEC-11-ODKBA-A-PA
PR.MAC	DEC-11-QDPRA-A-PA
PP.MAC	DEC-11-ODPPA-A-PA
LP.MAC	DEC-11-ODLPA-A-PA
DT.MAC	DEC-11-ODDTA-A-PA

D.2 DECTAPE KIT

Six DECtapes are distributed, viz.:

CONTENTS	LABEL
DOS VØ8 MONITOR TAPE: or, BATCH VØ8 MONITOR TAPE	DEC-11-ODMOA-A-UC or, DEC-11-OBMOA-A-UC
BATCH/DOS VØ8 SYSTEM PROGRAM TAPE:	DEC-11OCSPA-A-UC1 thru UC2
FORTRAN COMPILER V4A TAPE:	DEC-11-LFCTA-A-UC
FORTRAN OTS V2ØA TAPE:	DEC-11-LFOLA-A-UC
BATCH/DOS VØ8 DEVICE DRIVER TAPE:	DEC-11-ODRVA-A-UA

D.3 MAGNETIC TAPE KIT

A single magnetic tape (7- or 9-channel) contains either the DOS-11 or BATCH-11 system. The Monitor files are stored under account 2ØØ,2ØØ.

CONTENTS	LABEL
DOS VØ8/FORTRAN V4A/OTS v2ØA/ SYSTEM PROGRAMS MAGTAPE	DEC-11-ODFSA-A-MC
BATCH VØ8/FORTRAN V4A/OTS V2ØA/ SYSTEM PROGRAMS MAGTAPE	DEC-11-OBFSA-A-MC

D.4 PAPER TAPE BATCH STREAMS

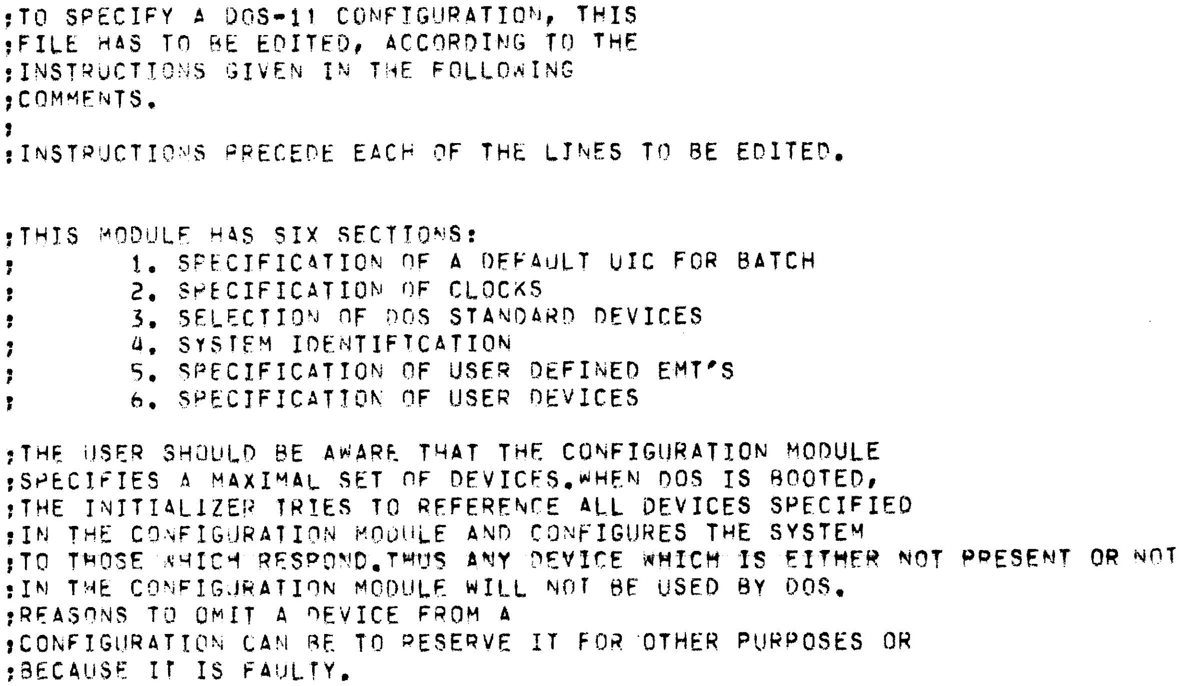
GOAP16.DOS	DEC-11-GOAPA-A-PA1
GOAP.ALL	DEC-11-GOAPA-A-PA2
GOAP.16K	DEC-11-GOAPA-A-PA3
GOTA.16L	DEC-11-GOTAA-A-PA

APPENDIX E

STANDARD CONFIGURATION MODULE

E.1 DESCRIPTION

The Configuration Module source file is named SYSTEM.MAC. It contains code that defines a standard BATCH/DOS-11 system configuration. It also contains comments that summarize how to modify it to fit a particular user's installation.



```
;TO SPECIFY A DOS-11 CONFIGURATION, THIS
;FILE HAS TO BE EDITED, ACCORDING TO THE
;INSTRUCTIONS GIVEN IN THE FOLLOWING
;COMMENTS.
;
;INSTRUCTIONS PRECEDE EACH OF THE LINES TO BE EDITED.


;THIS MODULE HAS SIX SECTIONS:
;        1. SPECIFICATION OF A DEFAULT UIC FOR BATCH
;        2. SPECIFICATION OF CLOCKS
;        3. SELECTION OF DOS STANDARD DEVICES
;        4. SYSTEM IDENTIFICATION
;        5. SPECIFICATION OF USER DEFINED EMT'S
;        6. SPECIFICATION OF USER DEVICES

;THE USER SHOULD BE AWARE THAT THE CONFIGURATION MODULE
;SPECIFIES A MAXIMAL SET OF DEVICES.WHEN DOS IS BOOTED,
;THE INITIALIZER TRIES TO REFERENCE ALL DEVICES SPECIFIED
;IN THE CONFIGURATION MODULE AND CONFIGURES THE SYSTEM
;TO THOSE WHICH RESPOND.THUS ANY DEVICE WHICH IS EITHER NOT PRESENT OR NOT
;IN THE CONFIGURATION MODULE WILL NOT BE USED BY DOS.
;REASONS TO OMIT A DEVICE FROM A
;CONFIGURATION CAN BE TO RESERVE IT FOR OTHER PURPOSES OR
;BECAUSE IT IS FAULTY.
```

```
;BATCH USES A DEFAULT UIC WHEN NO UIC
;WAS SPECIFIED, THE DEFAULT UIC IS [1,2].
;TO CHANGE THE DEFAULT UIC [1,2] TO [X,Y],
;REPLACE DFU    1,2     WITH
;        DFU    X,Y
         DFU    1,2
;THERE ARE 2 POSSIBLE CLOCKS:
; 1 KW11-L THE LINE CLOCK
; 2 KW11-P THE PROGRAMMABLE CLOCK
;THE PROGRAMMABLE CLOCK WILL BE RUN IN LINE MODE.
;TO DELETE THE LINE CLOCK FROM ALL POSSIBLE
;CONFIGURATIONS, DELETE THE NEXT LINE
FTCLL=1
;TO DELETE THE PROGRAMMABLE CLOCK FROM
;ALL POSSIBLE CONFIGURATIONS, DELETE
;THE NEXT LINE.

FTCLP=1                 ;PROGRAMMABLE CLOCK

;IN THE NEXT LINES ALL STANDARD PDP-11
;DEVICES ARE DEFINED. THIS IS DONE BY MEANS
;OF DEFINING A SYMBOL FTXX, WHERE XX
;IS THE DEVICE MNEMONIC.
;TO DELETE DEVICE XY DELETE THE LINE
;FTXY=1 BELOW.  DO THIS FOR ALL
;DEVICES NOT WANTED IN YOUR CONFIGURATION.
FTDC=1          ;RC DISK
FTKB=1          ;TELETYPE
FTPR=1          ;HIGH SPEED PAPER TAPE READER
FTDT=1          ;DECTAPE
FTCR=1          ;CARD READER
FTMT=1          ;MAGNETIC TAPE




FTDF=1          ;RF DISK
FTDK=1          ;RK DISK
FTLP=1          ;LINE PRINTER
FTPP=1          ;HIGH SPEED PAPER TAPE PUNCH
        PAGE
;THE SYSTEM GENERATED CAN BE IDENTIFIED
;BY A USER DEFINED IDENTIFICATION LINE PRINTED
;WHEN THE SYSTEM IS INITIALIZED.
;INSERT THE IDENTIFICATION LINE BETWEEN
;THE ANGLE BRACKETS (NOTE: INCLUDE
;NO ANGLE BRACKETS IN YOUR TEXT)
; ON THE NEXT LINE.

        IDENT <>

;I.E. IDENT <RF SYSTEM NO RK>
```

```
;IN THE NEXT LINES THE IDENTIFICATION PRINTED AFTER THE V
;IN THE MONITOR MESSAGE IS DEFINED. DO NOT CHANGE THIS.
        MONVRS  <Ø8-Ø2>

        .PAGE
;IF NO USER DEFINED EMT'S OR NON
;STANDARD DEVICES ARE TO BE ADDED TO THE
;SYSTEM, THEN THE EDITING OF THIS FILE
;IS FINISHED.
;IN THE NEXT SECTION NON-STANDARD EMT'S
;ARE DEFINED. THESE HAVE TO FALL IN THE RANGE
;14Ø-177. ALL OTHER VALUES ARE RESERVED
;FOR DEC.
;
;
;IN THE NEXT LINE THE SYMBOL UDEMT
;MUST BE EQUATED TO THE DIFFERENCE
;BETWEEN THE HIGHEST EMT VALUE DEFINED AND
;77 (ALL NUMBERS ARE OCTAL).
;A ZERO MEANS THERE ARE NO NON STANDARD
;EMT'S.
;E.G. IF THE HIGHEST EMT IS 166 THEN UDEMT=67.

UDEMT=Ø
;
;TWO DIFFERENT TYPES OF EMT'S CAN BE
;DEFINED:
; 1. SWAPPABLE EMT'S
; 2. EMT'S LOADED RESIDENT DURING
;     SYSTEM INITIALIZATION
;FIRST DEFINE THE TOTAL NUMBER OF SWAPPABLE EMT'S
;(IN OCTAL) IN THE NEXT LINE (UDEMSW=...).

UDEMSW=Ø                        ;REPLACE Ø BY NUMBER OF SWAPPABLE
                                ;EMT'S

;NEXT DEFINE THE NUMBER OF EMT'S TO BE LOADED
;RESIDENT DURING SYSTEM INITIALIZATION.

UDEMRS=Ø                        ;REPLACE Ø BY NUMBER OF
                                ;USER EMT'S TO BE LOADED
                                ;RESIDENT.
;AFTER THE DEFINITION OF THE NUMBER OF USER EMT'S,
;THE VALUES OF THE EMT'S MUST BE SPECIFIED.
;THIS IS DONE IN THE MACROS GEMUØ-GEMUS
;DO THIS IN THE FOLLOWING WAY:
; 1 DO NOT DEFINE MORE THAN 13 (OCTAL)
;   EMT'S IN A SINGLE MACRO.
; 2 DO NOT DEFINE BOTH RESIDENT EMT'S
;   AND SWAPPABLE EMT'S IN THE SAME MACRO
; 3 START BY DEFINING THE VALUES OF SWAPPABLE
;   EMT'S IN MACRO GEMUØ, IF THERE ARE
;   MORE THAN 13 (OCTAL) VALUES CONTINUE
;   THE DEFINITION IN MACRO GEMU1
; 4 THE TITLE OF EMT MODULES MUST BE .TITLE EM.XXX,
;   WHERE XXX IS THE OCTAL NUMBER OF THE EMT.
;   E.G. EMT 166 HAS A TITLE STATEMENT:
;  .TITLE       EM.166
; 5 THE LOAD MODULES SHOULD BE INSERTED IN THE CIL.
;  THE EMT CODE SHOULD BE POSITION INDEPENDENT AND
;  LINKED AT A BOTTOM OF Ø.
```

```
;AFTER THE MACRO DEFINITION, VALUFS OF USER DEFINED EMT'S
;CAN BE INSERTED BETWEEN THE ANGLE BRACKETS.
;NOTE SEE RULES:
;E.G.    .IRP     P,<14Ø,146>                ;DEFINES EMT'S 14Ø AND 146
         .MACRO   GEMUØ    VAL
         .IRP     P,<>
         .IIF EQ,PHASE SUBST<PROMRT+<P*2>>,VAL
         .IIF NE,PHASE R5Ø<EMPRIM+<P*2>>,P
         .ENDM
         .ENDM    GEMUØ
;AFTER THE MACRO DEFINITION, VALUES OF USER DEFINED EMT'S
;CAN BE INSERTED BETWEEN THE ANGLE BRACKETS.
;NOTE SEE RULES:
;E.G.    .IRP     P,<14Ø,146>                ;DEFINES EMT'S 14Ø AND 146
         .MACRO   GEMU1    VAL
         .IRP     P,<>
         .IIF EQ,PHASE SUBST<PROMRT+<P*2>>,VAL
         .IIF NE,PHASE R5Ø<EMPRIM+<P*2>>,P
         .ENDM
         .ENDM    GEMU1
;AFTER THE MACRO DEFINITIONE VALUES OF USER DEFINED EMT'S
;CAN BE INSERTED BETWEEN THE ANGLE BRACKETS
;NOTE SEE RULES:
;E.G.    .IRP     P,<14Ø,146>                ;DEFINES EMT'S 14Ø AND 146
         .MACRO   GEMU2    VAL
         .IRP     P,<>
         .IIF EQ,PHASE SUBST<PROMRT+<P*2>>,VAL
         .IIF NE,PHASE R5Ø<EMPRIM+<P*2>>,P
         .ENDM
         .ENDM    GEMU2
;AFTER THE MACRO DEFINITION, VALUES OF USER DEFINED EMT'S
;CAN BE INSERTED BETWEEN THE ANGLE BRACKETS.
;NOTE SEE RULES:
;E.G.    .IRP     P,<14Ø,146>                ;DEFINES EMT'S 14Ø AND 146
         .MACRO   GEMU3    VAL
         .IRP     P,<>
         .IIF EQ,PHASE SUBST<PROMRT+<P*2>>,VAL
         .IIF NE,PHASE R5Ø<EMPRIM+<P*2>>,P
         .ENDM
         .ENDM    GEMU3
;AFTER THE MACRO DEFINITION, VALUES OF USER DEFINED EMT'S
;CAN BE INSERTED BETWEEN THE ANGLE BRACKETS.
;NOTE SEE RULES:
;E.G.    .IRP     P,<14Ø,146>                ;DEFINES EMT'S 14Ø AND 146
         .MACRO   GEMU4    VAL
         .IRP     P,<>
         .IIF EQ,PHASE SUBST<PROMRT+<P*2>>,VAL
         .IIF NE,PHASE R5Ø<EMPRIM+<P*2>>,P
         .ENDM
         .ENDM    GEMU4
         PAGE
;AFTER THE MACRO DEFINITION, VALUES OF USER DEFINED EMT'S
;CAN BE INSERTED BETWEEN THE ANGLE BRACKETS.
;NOTE SEE RULES:
;E.G.    .IRP     P,<14Ø,146>                ;DEFINES EMT'S 14Ø AND 146
         .MACRO   GEMU5    VAL
         .IRP     P,<>
         .IIF EQ,PHASE SUBST<PROMRT+<P*2>>,VAL
         .IIF NE.PHASE R5Ø <EMPRIM+<P*2>>,P
         .ENDM
         .ENDM    GEMU5
```

```
;IF EMT'S HAVE BEEN ADDED AND THE LINKER SHOULD
;BE ABLE TO LOAD THEM RESIDENT, THE TABLE
;STARTING AT LABEL GLOTAB  HAS TO BE EXTENDED.
;  THE TABLE GLOTAB DEFINES THE RADIX5Ø NAMES OF EMT'S.
;WHEN THERE REMAIN UNDEFINED GLOBALS,THE LINKER SEARCHES
;THE TABLE GLOTAB, AND THE TABLE INDEX
;DIVIDED BY TWO OF ANY TABLE ENTRY MATCHING AN UNDEFINED
;GLOBAL WILL BE THE NUMBER OF A EMT TO BE LOADED
;RESIDENT.
;TO ADD THE GLOBAL NAMES OF USER DEFINED EMT'S
;USE THE FOLLOWING RULES:
;1  SUBTRACT 1ØØ FROM THE LOWEST DEFINED EMT
;2  IF THIS NUMBER IS X,THEN EXTEND THE TABLE WITH X ZEROES
;3  ADD NEXT .RAD5Ø      /EMX/
;  WHERE EMX IS THE  GLOBAL NAME
;4 IF MORE THAN ONE EMT HAS BEEN DEFINED,REPLACE 1ØØ BY THE VALUE
;  OF THE EMT JUST DEFINED +1, AND TAKE THE NEXT EMT VALUE AS
; LOWEST  EMT VALUE, AND REPEAT STEP 1.
;
;NOTE:  ALL NUMBERS ARE OCTAL
;
GLOTAB:
        .WORD   GLOLST-GLOTAB             ;LENGTH OF TABLE HEADER INCLUDED
        .RAD5Ø  /EMT/           ;  Ø,WAITR
        .RAD5Ø  /EMT/           ;  1,WAIT
        .RAD5Ø  /RWN/           ;2 WRITE
        .RAD5Ø  /RWR/           ;3 NYA
        .RAD5Ø  /RWN/           ;4 READ
        .RAD5Ø  /RWR/
        .RAD5Ø  /INR/           ;  6,DATASET INITIALIZATION
        .RAD5Ø  /RLS/           ;  7,DATASET RELEASE
        .RAD5Ø  /TRA/           ; 1Ø,TRAN PRCESSOR
        .RAD5Ø  /BLO/
        .RAD5Ø  /SPC/
        .RAD5Ø  /STT/
        .RAD5Ø  /DIR/
        .RAD5Ø  /ALO/
        .RAD5Ø  /OPN/
        .RAD5Ø  /CLS/
        .RAD5Ø  /REN/
        .RAD5Ø  /DEL/
        .RAD5Ø  /APP/
        .RAD5Ø  /GAR/
        .RAD5Ø  /PRO/
        .RAD5Ø  /REC/
        .WORD   Ø
        .WORD   Ø
        PAGE

        .WORD   Ø,Ø,Ø,Ø,Ø,Ø     ;EMT'S 3Ø-35
        .WORD   Ø
        .WORD   Ø
        .RAD5Ø  /SPE/           ; 4Ø,MSB OVERLAY PROCESSOR, RESIDENT
        .RAD5Ø  /GUT/           ; 41, GENERAL UTILITIES
        .RAD5Ø  /CVT/           ; 42,CONVERSION PROCESSOR
        .RAD5Ø  /FOP/
        .RAD5Ø  /FCR/
        .RAD5Ø  /FCL/
        .RAD5Ø  /LUK/
        .RAD5Ø  /LBA/
        .RAD5Ø  /GMA/
        .RAD5Ø  /CBA/
        .RAD5Ø  /CKX/
        .RAD5Ø  /DLN/
        .RAD5Ø  /DCN/
        .RAD5Ø  /AP2/
        .RAD5Ø  /CSX/
        .RAD5Ø  /CSM/
        .RAD5Ø  /XIT/
        .RAD5Ø  /LDR/
        .RAD5Ø  /LD2/
        .RAD5Ø  /MTO/
        .RAD5Ø  /DMP/
        .RAD5Ø  /RUN/           ; 65, RUN PROCESSOR
        .RAD5Ø  /CDT/
        .RAD5Ø  /BSF/

;ADD USER DEFINED GLOBALS IN FRONT OF THE NEXT LINE

GLOLST:

;IF NO NON STANDARD DEVICES HAVE TO BE
;DEFINED, THE EDITING OF THIS FILE IS READY.
;OTHERWISE INSERT THE NUMBER (OCTAL)
;OF NONSTANDARD DEVICES ON THE NEXT LINE.

UDDEV=Ø                 ;REPLACE THIS BY OCTAL NUMBER
                        ;OF NON STANDARD DEVICES.

;FOR EACH NON STANDARD DEVICE,THE
;MNEMONIC AND INTERRUPT VECTOR ADDRESS
;HAVE TO BE SPECIFIED.
;THE SPECIFICATION OF THESE ITEMS HAS TO BE DONE
;BY MEANS OF THE MACROS GDVUO, GVDU1
;AND GDVU2 USING THE FOLLOWING RULES.
; 1 START DEFINING BY USING GDVUØ, NEXT
;   GDVU1, AND THERFAFTER GDVU2
; 2 IN EACH MACRO, SPECIFICATIONS
;   FOR ONLY 3 DEVICES CAN BE GIVEN
; 3 IF THERE ARE 3 OR MORE SPECIFICATIONS
;   TO BE DONE, SPECIFY 3 IN THE CURRENT MACRO.
;THE MACROS GDVUØ, GDVU1 AND GDVU2
;PROVIDE THE FACILITY FOR 9 DEVICES.  WHEN
;MORE DEVICES HAVE TO BE ADDED, ADD
;ADDITIONAL MACROS WITH NAMES STARTING
;WITH GDVU3,-....
;HAVING THE SAME LAYOUT AS
;GDVUØ, GDVU1 AND GDVU2.
        PAGE
;HERE THE MNEMONIC AND INTERRUPT VECTOR
;ADDRESS OF UP TO 3 DEVICES CAN BE SPECIFIED.
;IF THE MNEMONICS AND INTERRUPT VECTOR
;ADDRESS ARE RESPECTIVELY:
;X1,X2,X3 AND IVA1,IVA2 AND IVA3
;THEN EDIT THE LINE AFTER THE MACRO DEFINITION TO READ:
;       .IRP    P,<<X1,IVA1>,<X2,IVA2>,<X3,IVA3>>
        .MACRO  GDVUØ
        .IRP    P,<>
        UDEVG   P
        .ENDM
        .ENDM   GDVUØ
;HERE THE MNEMONIC AND INTERRUPT VECTOR
;ADDRESS OF UP TO 3 DEVICES CAN BE SPECIFIED.
;IF THE MNEMONICS AND INTERRUPT VECTOR
;ADDRESS ARE RESPECTIVELY:
;X1,X2,X3 AND IVA1,IVA2 AND IVA3
```

```
;THEN EDIT THE LINE AFTER THE MACRO DEFINITION TO READ:
;        .IRP     P,<<X1,IVA1>,<X2,IVA2>,<X3,IVA3>>
         .MACRO   GDVU1
         .IRP     P,<>
         UDEVG    P
         .ENDM
         .ENDM    GDVU1
;HERE THE MNEMONIC AND INTERRUPT VECTOR
;ADDRESS OF UP TO 3 DEVICES CAN BE SPECIFIED.
;IF THE MNEMONICS AND INTERRUPT VECTOR
;ADDRESS ARE RESPECTIVELY:
;X1,X2,X3 AND IVA1,IVA2 AND IVA3
;THEN EDIT THE LINE AFTER THE MACRO DEFINITION TO READ:

         .IRP     P,<<X1,IVA1>,<X2,IVA2>,<X3,IVA3>>
         .MACRO   GDVU2
         .IRP     P,<>
         UDEVG    P
         .ENDM
         .ENDM    GDVU2
```

APPENDIX F

SYSLOD NOTES

F.1 USE OF SWITCHES

The System Loader program provides a number of switches, which may be specified on the output side of the command string. Switches may be used only on the output side of the command string, in the general form:

```
#output CIL/SW1.../SWN<input LICIL
```

By including a switch, the user indicates that a particular action is to be performed by SYSLOD. Switches are summarized in Table F-1.

TABLE F-1

SYSLOD SWITCHES

Switch	Function
/ZERO	Zero the disk (initialize the BATCH/ DOS-11 file structure).
/BOOT	Boot the output CIL into core.
/HOOK	Hook the CIL to the bootstrap.
/BLOCKS:nnn	Allocate nnn blocks for the CIL.
/NS:lbs:stb	Non-standard. Load a CIL onto a disk that does not have standard DOS file structure.
/FORMAT	This switch causes an RK11 disk to be formatted when DKn: is the outnut device.

NOTE: The initial two letters of each switch are all that are required; the remainder are ignored. If no switches are used, or if the /ZERO switch or /NS switch is not used, the output CIL replaces the CIL that is currently hooked to the bootstrap. SYSLOD searches the bootstrap for the starting block and logical block size of the current CIL, and uses these values when creating and loading the new CIL.

F.2 SWITCH DESCRIPTIONS

F.2.1 /ZERO Switch

The /ZERO switch clears every block on the output device, initializes the BATCH/DOS-11 file structure, and writes a new bootstrap in block Ø. The output CIL is installed on the disk as the first and only file under the [1,1] User Identification Code (UIC).

Example:

```
#DK:MONCIL.CIL/ZE<DT4:MONCIL.LCL
```

F.2.2 /NS Switch

The NS (Non-Standard) switch is used when the CIL is either to be loaded on a disk that does not have a standard DOS-11 file structure, or when the CIL itself does not adhere to the DOS-11 file structure. All relevant information must be supplied by the user, either specifically or by default. The /NS switch has the form:

/NS:*lbs*:*stb*

where:

lbs = logical block size (a decimal value designating words-per-block). If lbs is not given, the standard DOS logical block size is used:

```
RC11, RF11 = 64
RK11       = 256
```

lbs can be used without stb specified.

stb = starting block number, at which the CIL is to begin. Default = 4. Stb cannot be used alone. Lbs must also be specified. The value specified for stb must be 4 or greater, because blocks Ø through 3 are occupied by the Bootstrap.

The SYSLOD program puts the CIL onto the output disk, with the logical block size specified, and at the starting block specified. If there is not enough contiguous disk space available, SYSLOD prints an error message, and does not attempt to load the CIL.

The value of lbs, if specified, must be either 64 or 256, or must match the value given in the CIL Line. This ensures that SYSLOD will allocate enough blocks for the CIL Index.

If the CIL is to be hooked, the bootstrap requires that the value of lbs be a multiple of 64.

The /NS switch and /ZERO switch are mutually exclusive. An error message is produced if they appear in the same command string. All other SYSLOD switches can be specified, however. See Table F-2.

Example:

```
#DF:JHR.CIL/NS:256:4Ø<DTØ:JHR.LCL
```

The input dataset is a Linked CIL named JHR.LCL, resident on DECtape unit Ø. SYSLOD puts the output CIL (JHR.CIL) on the RF11 disk, starting at logical block 4Ø, with a logical block size of 256 words per block.

F.2.3 /BOOT Switch

The /BOOT switch causes the output CIL to be booted into core and executed, after it has been loaded onto the disk by SYSLOD. It is only necessary to specifically request that this be done when the /NS switch is used (refer to Section 2.1.4.2). If the /NS switch is not used, /BOOT is assumed by default.

Example:

```
#DK1:MONCIL.CIL/NS:256:4Ø/BO<DTØ:MONCIL.LCL
```

The Monitor CIL is specified to be loaded onto RK11 unit 1. The file structure is assumed to be unknown. Thus, it is necessary to specify /BO to boot in the CIL, and begin its execution.

F.2.4 /HOOK Switch

The /HOOK switch causes the bootstrap on the output disk to be updated with information pertaining to the output CIL, so the bootstrap will boot that CIL into core when it is executed. If the /NS switch is not used, the /HOOK switch is assumed by default.

Example:

```
#DC:MONCIL.CIL/NS:256:3Ø/HO<DT:MONCIL
```

The CIL (MONCIL.CIL) is loaded onto the RC11 disk, with a logical block size of 256 words, starting at logical block 3Ø. The bootstrap on the disk is updated to reflect the following information about MONCIL.CIL.

a. Start block of first core image in the CIL.
b. Size (number of bytes) of first core image in the CIL.
c. Load address in core of first core image in the CIL.
d. Start address of first core image in the CIL.
e. Pointer to first block of the CIL (block 3Ø).

F.2.5 /BLOCKS Switch

The /BLOCKS switch is used to allocate a specified number of logical blocks for the output CIL. In this way, extra disk space can be reserved for later extension to the CIL. The switch is specified as

```
/BL[OCKS]:NNNNN
```

where NNNNN is a decimal number, up to five digits long, specifying the number of logical blocks to be allocated.

The /BLOCKS switch cannot be used alone. It must be used in conjunction with either the /ZERO or the /NS switch. When used with the /ZERO switch, /BLOCKS allocates more than the minimum number of blocks required for the CIL. When used with the /NS switch, /BLOCKS ensures that only the number of blocks specified are written into by the CIL. For example, assume that the user knows that there are 1ØØ blocks of contiguous space available, starting at block 5ØØ on the disk. He wishes to load the CIL into this space, but is not sure how many blocks are needed. To make sure that block 6ØØ will not be destroyed by the CIL, he would specify the following command string:

```
#DK1:SYS.CIL/NS:256:5ØØ/BL:1ØØ<PR:
```

SYSLOD computes the number of blocks needed before it attempts to load the CIL. If more than 1ØØ are needed, it prints the message

```
LICIL TOO BIG, NOT ENOUGH RESERVED BLOCKS !
```

on the teleprinter, and restarts, typing # at the beginning of the next line.

If the user has specified enough blocks, the CIL is loaded as specified. If the /BLOCKS switch was not used, the CIL would be loaded, starting at block 5ØØ, up to the number of blocks required, heedless of the contents of the blocks into which it was being loaded. The minimum-number-of-blocks value is determined by examining the CIL Line, which was produced by CILUS at the time the LICIL was generated.

F.2.6 /FORMAT Switch

This switch may be included only when the output disk is an RK11 unit. Its function is to put maintenance format information on the disk, such as the start of physical blocks, sectors, etc. If the /FO switch is specified for an RC11 or RF11 disk, it has no effect.

Example:

```
#DKØ:MONLIB.CIL/FO<MTØ:MONLIB.LCL
```

F.2.7 Switch Combinations

Table F-2 illustrates the legal combination of SYSLOD switches.

TABLE F-2

LEGAL SYSLOD SWITCH COMBINATIONS

	ZERO	NS	BLOCKS	HOOK	BOOT	FORMAT
ZERO		I	L	L,D	L,D	L
NS	I		L	L	L	L
BLOCKS	L	L		L	L	L
HOOK	L,D	L	L		L	L
BOOT	L,D	L	L	L		L
FORMAT	L	L	L	L	L	

LEGEND

L - Legal
I - Illegal
D - Default

NOTES:

1. /ZERO and /NS are mutually exclusive.
2. /HOOK and /BOOT are set by default unless /NS is used.
3. /HOOK and /BOOT can be used with any switches.
4. /BLOCKS requires that either /ZERO or /NS be used.
5. /FORMAT is usable only with RK11 disks.

F.3 SYSLOD ERROR MESSAGES

F.3.1 Recoverable Errors

The following errors are diagnosed and reported by SYSLOD. Once the error message is printed, SYSLOD will restart by identifying itself again, and printing the # (input request) character on the teleprinter. The user should retry the command, making the indicated corrections.

1. SYNTAX ERROR

   This message is printed if the command input line has a syntax error. The offending line is printed up to the first error character, with a question mark following the error character.

   EXAMPLE:

   ```
   DK1:FILE$LDA<DT1:FILE.LCL
   ```

   This line would be echoed as follows:

   ```
   DK1:FILE$? )
   ```

2. TOO MANY DATASETS

   SYSLOD accepts one input dataset and one output dataset. This message occurs when more than one input or output dataset is specified in the command line.

3. TOO MANY SWITCHES

   SYSLOD does not accept any switches from the input dataset. If more output switches are used than can be accommodated, SYSLOD issues this error message.

4. UNKNOWN SWITCH

   If a switch is typed that is not one of those recognized by SYSLOD, this message is issued.

5. SWITCH ERROR

   This error message is reported if a switch is used incorrectly. E.g.,

/ZE:1	ZE[RO] does not take arguments
/BL/HO	BL[OCK] requires a decimal argument

6. SWITCH CONTEXT ERROR

   This error is reported when switches are specified incorrectly for their definitions, e.g.,

   a. If BL[OCKS] or FO[RMAT] is specified /ZE or /NS must be specified also.

b. /NS and /ZE cannot both be specified.

7. ERROR IN SWITCH ARGUMENT

   Decimal arguments given with /NS or /BL are too large for 16 bits.

8. NON-EXISTENT DISK OR DISK NOT READY

   A legal disk name was given for the output dataset, but the device either is not part of the system configuration or is not ready.

9. UNKNOWN DISK NAME SPECIFIED

   SYSLOD accepts the following disk names as part of the output dataset:

DC	1 to 4 disks
DF	1 to 8 disks
DK	HIGH or LOW density

10. ERROR WHILE FORMATTING RK DISK

    Any disk error while formatting the RK disk will cause this message.

F.3.2 Non-recoverable Errors

There are two types of non-recoverable errors. First, the 'F' errors, which are the same as those in DOS-11. Refer to the DOS Monitor Programmer's Manual.

The second type of non-recoverable errors are described below.

1. INPUT IS NOT A LICIL

   The first line of the input file was not COMD section #4 (see F.4, Note c). This will usually occur when the user tries to SYSLOD a load (.LDA) module instead of a LICIL.

2. END OF FILE BEFORE CIL LINE READ

   This occurs when SYSLOD is looking for the CIL line but can't find it. Typically this occurs because the user put the wrong paper tape in the tape reader, when he has many tapes comprising the whole LICIL. When the input is from DECtape or magnetic tape, this message usually indicates that the file has been destroyed, and must be rebuilt.

3. BOOTSTRAP NOT IN BLOCK Ø

   This can only occur in 'replace' mode (/NS and /ZE[RO] not specified). In this mode, SYSLOD finds the CIL to be re-replaced on the disk by looking at the bootstrap parameters. If the first block number of the CIL hooked to the bootstrap (i.e. LOCATION 176 OF BOOT) is Ø then block Ø is certainly not a hooked bootstrap.

4. BOOTSTRAP NOT HOOKED TO CIL; CANNOT REPLACE

   This can only occur in 'replace' mode. If the first block indicator of the bootstrap (LOCATION 176) is non-Ø, it must be pointing to a CIL. If the first formatted binary line of the 'hooked' file is not COMD section #3 it is not a CIL.

5. BLOCK SIZE DISCREPANCY BETWEEN CILUS AND SYSLOD

   This occurs when the /NS switch is used with CILUS or SYSLOD but not both, (i.e., exactly the same parameters must be specified with SYSLOD that were specified with CILUS).

6. LICIL TOO BIG, NOT ENOUGH RESERVED BLOCKS

   a. In replace mode this occurs when the new LICIL is larger than the old.

   b. If not replace mode, the number of reserved blocks (/BL:Nnnnnn) is not large enough to hold the CIL.

7. 1st LINE NOT COMD SECTION #4 or 1

   After the CIL line is read, SYSLOD starts loading the LICIL. The first formatted binary line after the CIL line must be COMD section #4 or COMD section #1. After each core image is loaded, SYSLOD sets itself for loading a new core image. If the beginning of the new core image is not COMD section #4 or #1, this error message will result.

8. COMD SECTION #4 SEQUENCE ERROR

   This will occur when the user has a LICIL made up of many paper tapes. If the tapes are not read in the correct sequence, this error will result.

9. INPUT ERROR

   After a READ, the status in the buffer header indicated that one of the following errors occurred:

   a. invalid line error

   b. checksum error

   c. character parity error or illegal binary format

   d. device parity error

10. LOGICAL BLOCK SIZE ERROR

    This occurs when the logical block size given with the /NS switch is not an even multiple of the default logical block size. E.g., a block size of 1ØØ for RF11 is not an even multiple of the default size of 64.

11. END OF DISK BEFORE CIL COMPLETE

    The last block number of the output disk has been written, but the CIL is not complete.

F.3.3 Non-fatal Error

The following error is reported by SYSLOD but does not stop the loading process.

1. DISK ERROR AT DECIMAL BLOCK #NNNN, SYSLOD WILL CONTINUE

   After a TRAN out, the status word of the TRAN BLOCK is examined. If it shows an error, the TRAN is retried 10 times. If the error persists, the offending block number is printed, after which SYSLOD proceeds.

F.4 NOTES

A. If the default block size is not used the following restrictions apply:

   1. The argument block size must not exceed 256 words.
   2. The block size must be an even multiple of the default size for the disk.

B. If magnetic tape is the input medium, and the wrong filename or UIC is given, SYSLOD will halt at 24432_8 instead of printing FØ12.

C. The COMD (COMmunication Directory) contains a code number that identifies the kind of information that follows. SYSLOD expects the first formatted binary line it receives on input to be identified as code #4 (indicating that it is a LICIL).

APPENDIX G

CILUS NOTES

G.1 CILUS SWITCHES

Switches may appear in a CILUS command on either the output side or the input side.

G.1.1 Output Switches

Output switches are:

```
/HOOK  -  Hook the output CIL to the bootstrap.
/BOOT  -  Boot the output CIL.
```

G.1.2 Input Switches

Input switches include:

```
/LOAD  -  Loads a CIL or LICIL onto disk or DECtape.

/LIST  -  Produces a directory listing of a CIL or
          LICIL.

/BUILD -  Creates a CIL or LICIL.

/E     -  Indicates the end of input datasets.
```

G.2 CILUS COMMAND STRINGS

The general form of a CILUS command string is:

```
#CIL,LISTING,LICIL<PRIMARY,SECONDARY/E
```

All output datasets are optional, but at least one output dataset must appear in the command string. No default is assumed. The order of appearance determines the function of the datasets. I.e., if all three are specified, then (from left to right) they are defined to be the CIL dataset, the listing dataset, and the LICIL dataset. If a leading dataset is omitted, it must be indicated by a comma in its place, e.g.:

```
#CIL,,LICIL<PRIMARY,SECONDARY/E
```

indicates that the listing dataset is omitted. Legal output dataset devices are shown in Table G-1.

Table G-1

CILUS OUTPUT DEVICES

Dataset Type	Disk	DECtape	Magtape	Papertape	Printer/ Keyboard
CIL	X	X			
LICIL	X	X	X	X	
Listing	X	X	X	X	X

NOTES

1. A CIL cannot be stored on paper tape or magnetic tape because DOS-11 does not support contiguous files on these devices.
2. The LICIL format provides the means for distribution of CIL's on papertape and magnetic tape, as well as on disk and DECtape.

Output switches can be specified for either the CIL or the LICIL, but not for both in the same command string.

Input datasets consist of at least a primary dataset (required), and one or more secondary datasets (optional). Secondary input datasets can be continued on as many lines as necessary, terminated by the /E switch.

When LOADing or LISTing, only a primary input dataset is allowed (which must be a CIL or LICIL). For example:

```
#OUTCIL,LP:<INCIL/LOAD
```

This loads INCIL under the name OUTCIL, on the system disk (by default) and produces a listing of OUTCIL's directory on the line printer.

```
#LP:<INCIL/LIST
```

This produces a directory of INCIL's contents on the line printer.

The /E switch is not needed for either /LOAD or /LIST.

The /BUILD switch requires a primary input dataset, and accepts optional secondary inputs. The input datasets can be CIL's, LICIL's, or load modules. They are built in the order specified, to produce the specified output. The /BUILD switch must appear immediately after the primary dataset; e.g.,

```
#LIB,LP:,SY:LIBLCL<INLIB/BU,MOD1,MOD2/E
```

The contents of INLIB, MOD1, and MOD2 (in that order) are built into the CIL named LIB. A line printer listing, and a LICIL (LIBLCL) are also output (to the system device, as indicated by SY:). The /E switch is required.

G.3 EDIT COMMANDS

A CIL can be edited by means of the following commands:

```
INSERT
REPLACE
DELETE
PURGE
EXECUTE
```

To edit a CIL, specify a command string with no action switch on the input side (i.e., no /LOAD, /LIST, or /BUILD). CILUS responds by printing * on the next line. Edit commands can be used, as follows (NOTE: only the first letter of each command is required):

INSERT

```
*I P1:S1
```

where P1 - the name of a module in the primary input dataset;

S1 - the name of a module in the secondary input dataset.

Result: S1 is inserted into the primary input dataset ahead of P1.

```
*I P1:S1,S1A,S1B

    where P1  - Primary dataset module

          S1-S1B - Secondary dataset modules.

    Result:     S1, S1A, S1B are put into primary data-
                set (in that order) ahead of P1.
                NOTE:  S1, S1A, S1B must also appear
                in this order in the secondary data-
                set.

*I S1,S1A,S1B

    where S1, S1A, S1B - Secondary dataset modules.

    Result:     Because no primary input dataset
                module is specified, S1, S1A, and
                S1B are appended to the primary
                input dataset, in the order in
                which they appear in the secondary
                input dataset (their order in the
                command is ignored).
```

REPLACE

```
*R P1:S1

    where P1  - Primary input dataset module to be
                replaced.

          S1  - Secondary input dataset module that
                replaces P1.
```

DELETE

```
*D P1,P2,...,Pn

    where       Delete can be used to eliminate
                modules from the primary input
                dataset only(P1,P2,...Pn).

*PURGE

                This command is used to correct
                errors made while specifying edit
                commands.  If a wrong module was
                specified to be deleted, replaced,
                etc., specifying *P clears all pre-
                vious edit command input.  The edit
                commands can now be entered correctly.

*EXECUTE

                When all edit commands have been
                properly entered, use this command
                to cause the editing to be performed.
```

APPENDIX H

RESERVED FILENAME EXTENSIONS

Extension	Attribute
ALG	ALGOL source file
BAS	BASIC source file
BAK	Backup file
BLI	BLISS source file
CBL	COBOL source file
CIF	Core Image File
CIL	Core Image Library
CMD	Command file
CRF	Input to cross-referencing program
DAT	DATA file for FORTRAN job
DDT	Reserved for DDT
DGN	Diagnostic message file
FTN	FORTRAN source file
FCL	FOCAL source list
LBO	Library of object modules (other types of libraries may also be implemented)
LCL	Linked core image library
LDA	Load module, Absolute
LDR	Load module, Relocatable
LOG	Logging file
LSP	LISP source file
LST	Listing file
MAC	MACRO assembler source file
MAP	MAP file
MFD	Master file directory
OBJ	Object module
OPR	Program generation information
OVR	Overlay
PAL	PAL assembler source file
PL1	PL/1 source file
RNO	Reserved for RUNOFF program
ROL	Reserved for ROLLIN program
RPG	RPG source file
SNO	SNOBOL source file
SPC	SPEC format text
STB	Symbol Table (Link-11 output)
SYM	File of symbols
SYS	System management
TMP	Temporary scratch file
UFD	User file directory

INDEX

HOW TO OBTAIN SOFTWARE INFORMATION

Announcements for new and revised software, as well as programming notes, software problems, and documentation corrections are published by Software Information Service in the following newsletters.

Digital Software News for the PDP-8 & PDP-12
Digital Software News for the PDP-11
Digital Software News for the PDP-9/15 Family

These newsletters contain information applicable to software available from Digital's Program Library, Articles in Digital Software News update the cumulative Software Performance Summary which is contained in each basic kit of system software for new computers. To assure that the monthly Digital Software News is sent to the appropriate software contact at your installation, please check with the Software Specialist or Sales Engineer at your nearest Digital office.

Questions or problems concerning Digital's Software should be reported to the Software Specialist. In cases where no Software Specialist is available, please send a Software Performance Report form with details of the problem to:

Software Information Service
Digital Equipment Corporation
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754

These forms which are provided in the software kit should be fully filled out and accompanied by teletype output as well as listings or tapes of the user program to facilitate a complete investigation. An answer will be sent to the individual and appropriate topics of general interest will be printed in the newsletter.

Orders for new and revised software and manuals, additional Software Performance Report forms, and software price lists should be directed to the nearest Digital Field office or representative. U.S.A. customers may order directly from the Program Library in Maynard. When ordering, include the code number and a brief description of the software requested.

Digital Equipment Computer Users Society (DECUS) maintains a user library and publishes a catalog of programs as well as the DECUSCOPE magazine for its members and non-members who request it. For further information please write to:

DECUS
Digital Equipment Corporation
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754

BATCH/DOS-11 System
Manager's Guide
DEC-11-OMGRA-A-D
January 1973

READER'S COMMENTS

Digital Equipment Corporation maintains a continuous effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback -- your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability and readability.

____

____

____

____

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

____

____

____

____

____

How can this manual be improved?

____

____

____

____

____

Other comments?

____

____

____

____

____

Please state your position. ______________________________ Date: __________

Name: __________________________ Organization: __________________

Street: __________________________ Department: __________________

City: ________________ State: ________________ Zip or Country __________

Fold Here

Do Not Tear - Fold Here and Staple

digital

Digital Equipment Corporation
Software Information Services
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754

digital

DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASSACHUSETTS 01754