RT–11
Installation and System Generation Guide

Order No. AA-H376A-TC

March 1980

This document is intended for use by the individual responsible for installing RT-11 at the user site. It provides the information necessary to bootstrap and preserve the distribution volume(s); install mandatory patches; create the working system; and customize, preserve, and test the working system. This manual also describes the system generation process for those users who need to build special monitors and handlers.

SUPERSESSION/UPDATE INFORMATION:	This is a new manual.
OPERATING SYSTEM AND VERSION:	RT-11 V4
SOFTWARE VERSION:	RT-11 V4

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

digital equipment corporation · maynard, massachusetts

First Printing, March 1980

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

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

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

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

The following are trademarks of Digital Equipment Corporation:

DIGITAL
DEC
PDP
DECUS
UNIBUS
COMPUTER LABS
COMTEX
DDT
DECCOMM
ASSIST-11
VAX
DECnet
DATATRIEVE
DECsystem-10
DECtape
DIBOL
EDUSYSTEM
FLIP CHIP
FOCAL
INDAC
LAB-8
DECSYSTEM-20
RTS-8
VMS
IAS
TRAX
MASSBUS
OMNIBUS
OS/8
PHA
RSTS
RSX
TYPESET-8
TYPESET-11
TMS-11
ITPS-10
SBI
PDT

March 1981

This document is intended for use by the individual responsible for installing RT-11 at the user site. It provides information necessary to bootstrap and preserve the distribution volume(s); install mandatory patches; create the working system; and customize, preserve, and test the working system. This manual also describes the system generation process for those users who need to build special monitors and handlers.

RT-11 Installation and System Generation Guide

Order No. AA-H376A-TC

SUPERSESSION/UPDATE INFORMATION: This is a new manual. It includes Update Notice No. 1 (AD-H376A-T1).

OPERATING SYSTEM AND VERSION: RT-11 V4.0

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

digital equipment corporation · maynard, massachusetts

First Printing, March 1980
Updated, March 1981

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

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

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

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

The following are trademarks of Digital Equipment Corporation:

DIGITAL
DEC
PDP
DECUS
UNIBUS
COMPUTER LABS
COMTEX
DDT
DECCOMM
ASSIST-11
VAX
DECnet
DATATRIEVE
DECsystem-10
DECtape
DIBOL
EduSystem
FLIP CHIP
FOCAL
INDAC
LAB-8
DECSYSTEM-20
RTS-8
VMS
IAS
TRAX
MASSBUS
OMNIBUS
OS/8
PHA
RSTS
RSX
TYPESET-8
TYPESET-11
TMS-11
ITPS-10
SBI
PDT

M13500

5/82

Contents

Page

Page

Chapter 3 Installing a System Distributed on a Small Device to Run on a Small Device

Chapter 4 Installing a System Distributed on a Small Device to Run on a Hard Disk

Chapter 5 Installing a System Distributed on Hard Disk to Run on Hard Disk

Chapter 6 Installing a System Distributed on RX02 to Run on RX02

Chapter 7 Installing a System Distributed on Magtape to Run on Hard Disk

Page

PART III SYSTEM GENERATION

Chapter 8 Preparing for System Generation

Chapter 9 Performing the System Generation Process on a Disk System

Chapter 10 Performing the System Generation Process on a Small System

Appendix A SYSGEN Answers That Will Duplicate the Standard Monitors

Appendix B SYSGEN Answers That Create Example Systems

Appendix C Building System Programs

Appendix D SYSGEN Script Language

Appendix E Loading Software Bootstraps

Appendix F RT–11 Conditionals

Appendix G Software Kit Maps

Appendix H Link Maps for Standard (Distributed) Monitors

Figures

Tables

Page

Preface

How to Use This Manual

This manual introduces you to the RT–11 V04 software kit and describes how to install your operating system. It also describes how to perform the system generation process.

Part I of the manual introduces you to installation and system generation concepts and helps you choose a reading path through the other parts. The modular structure of this manual allows you to read only the chapters that you need for your particular combination of hardware and software.

Installation concepts — both planning activities and actual installation procedures — are covered in the six chapters of Part II. You should read Chapter 2 to plan installation, then choose the chapter (3, 4, 5, 6, or 7) that describes installation of a system that most closely resembles your own configuration.

If you discover in Part I that you need to perform the system generation process, Part III will provide you with information. Read Chapter 8 to prepare for system generation and Chapter 9 to learn how to run the SYSGEN program and build the resulting system. System generation on a dual-diskette system is described in Chapter 10. DIGITAL does not recommend a system generation of this kind.

Additional information available in the appendixes includes: SYSGEN answers to duplicate standard monitors, example SYSGEN answers, BATCH streams to build system programs, the SYSGEN script language, toggle-in software bootstraps, RT–11 conditionals, software kit maps, the standard monitor link maps, and customization patches for specially generated monitors.

Before you begin, you should read the *RT–11 Documentation Directory*, which describes the other documents in the RT–11 library. Familiarity with the RT–11 system, as described in the *RT–11 System User's Guide*, is particularly helpful when you perform the procedures in this manual.

If you are a FORTRAN, BASIC–11, FMS–11, CTS–300, or other layered product user, you build your FORTRAN, BASIC, FMS–11, or other system after building the RT–11 system. A layered product is software that is sold separately but requires the RT–11 operating system environment to run. See the appropriate installation manual for instructions on installing such products.

Documentation Conventions

You should become familiar with certain symbolic conventions used in this manual.

1. Examples consist of actual computer output whenever possible. In these examples, user input appears in red where it must be differentiated from computer output.

2. Unless the manual indicates otherwise, terminate all commands or command strings with a carriage return. Where necessary, the symbol RET represents a carriage return, LF a line feed, SP a space, ESC an ESCAPE or ALTMODE, and TAB a tab.

3. To produce certain characters in system commands, you must type a combination of keys concurrently. For example, while holding down the CTRL key, type C to produce the CTRL/C character. Key combinations such as this are documented as CTRL/C, CTRL/O, etc.

4. In descriptions of command syntax, capital letters represent the command name, which you must type. Lower case letters represent a variable for which you must supply a value.

5. In examples, you need to distinguish between the letter O and the number 0. Examples in this manual represent these characters as follows:

   letter O: O

   number 0: 0

6. The sample terminal dialogue provided in this document contains version numbers where they would normally appear. The version numbers given include xx in those fields that can vary from installation to installation. The exact contents of these fields are not of interest in the examples in this manual, as long as appropriate digits appear in the area indicated. The same is true for the FREE BLOCKS messages included in device directories.

   If you submit a software performance report (SPR) to DIGITAL, you must include the complete version number.

7. A decimal point (.) follows a number to indicate that it is a decimal number. A number without a decimal point is an octal number. For example, 128. is 128 (decimal) and 126 is 126 (octal).

Part I
Introduction

Chapter 1
Introduction

Now that your PDP–11 processor or PDT–11 Intelligent Terminal has been installed and your software distribution kit has been delivered, you are ready to get started with RT–11. To begin, you must install RT–11. Installation requires bootstrapping and preserving the distribution volumes, installing mandatory patches, creating the working system from selected components, and customizing, preserving, and testing the working system. In most cases, your system is ready to run once you have installed and tested it. However, if you need certain special features, you may also need to perform the system generation process. In this case, you must run the SYSGEN program, answer its questions, and assemble your own generated monitor(s) and handlers. Figure 1–1 summarizes the flow of the installation and system generation processes.

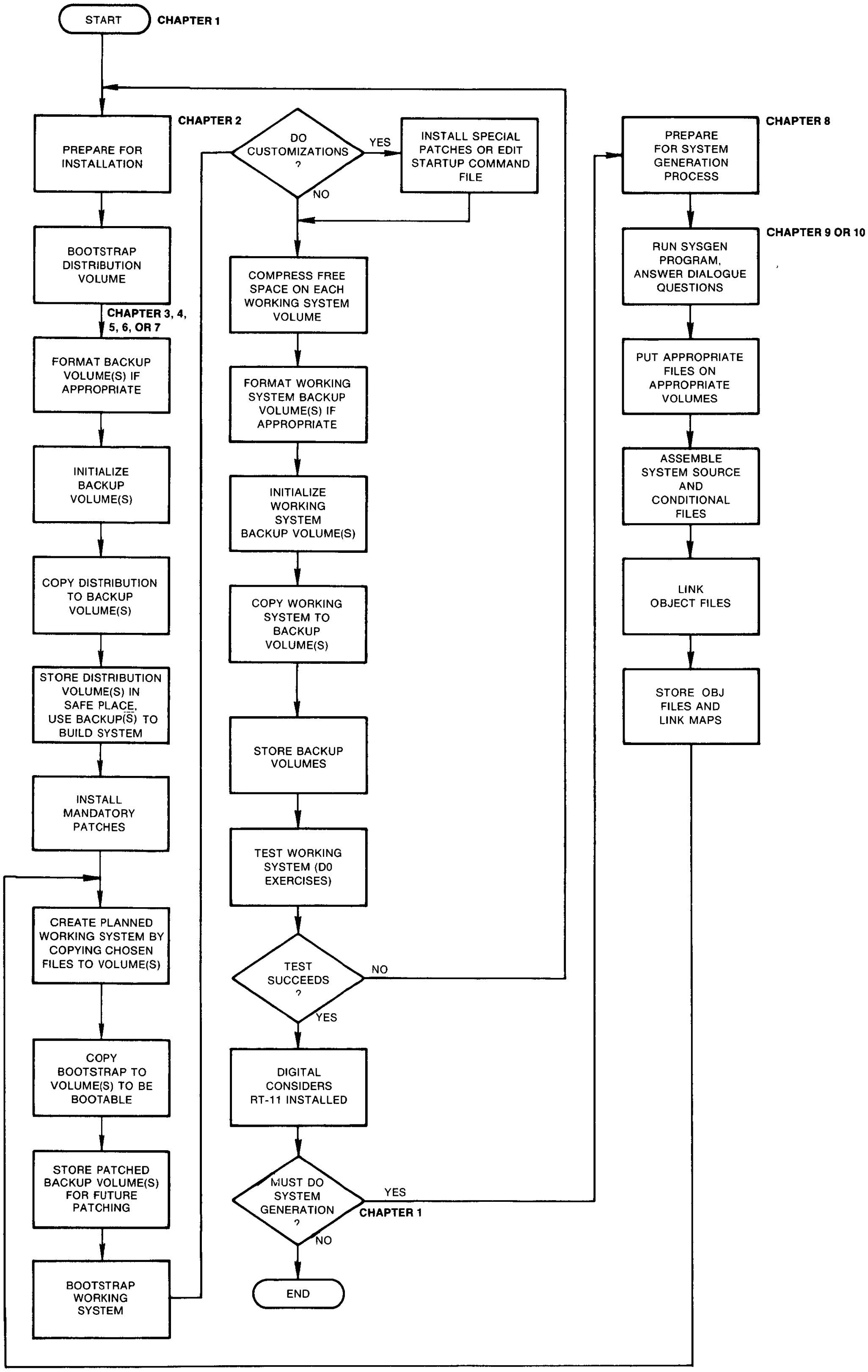
The following sections are intended to help you decide which course of action you should take. Once you have established which steps you must complete, you can identify the chapters you need to read. You should not need to read this entire manual.

1.1 Identifying Your Needs

In order to select the procedures you must perform and the reading path you should follow, answer the following questions, using the information in Sections 1.1.1 through 1.1.3.

- What is your hardware configuration?
- Which monitors do you need?
- Which features do you need?

Figure 1–1: Flowchart of Installation and System Generation


START
CHAPTER 1
PREPARE FOR INSTALLATION
CHAPTER 2
BOOTSTRAP DISTRIBUTION VOLUME
CHAPTER 3, 4, 5, 6, OR 7
FORMAT BACKUP VOLUME(S) IF APPROPRIATE
INITIALIZE BACKUP VOLUME(S)
COPY DISTRIBUTION TO BACKUP VOLUME(S)
STORE DISTRIBUTION VOLUME(S) IN SAFE PLACE. USE BACKUP(S) TO BUILD SYSTEM
INSTALL MANDATORY PATCHES
CREATE PLANNED WORKING SYSTEM BY COPYING CHOSEN FILES TO VOLUME(S)
COPY BOOTSTRAP TO VOLUME(S) TO BE BOOTABLE
STORE PATCHED BACKUP VOLUME(S) FOR FUTURE PATCHING
BOOTSTRAP WORKING SYSTEM
DO CUSTOMIZATIONS ?
YES
INSTALL SPECIAL PATCHES OR EDIT STARTUP COMMAND FILE
NO
COMPRESS FREE SPACE ON EACH WORKING SYSTEM VOLUME
FORMAT WORKING SYSTEM BACKUP VOLUME(S) IF APPROPRIATE
INITIALIZE WORKING SYSTEM BACKUP VOLUME(S)
COPY WORKING SYSTEM TO BACKUP VOLUME(S)
STORE BACKUP VOLUMES
TEST WORKING SYSTEM (DO EXERCISES)
TEST SUCCEEDS ?
NO
YES
DIGITAL CONSIDERS RT-11 INSTALLED
MUST DO SYSTEM GENERATION ?
YES
CHAPTER 1
NO
END
PREPARE FOR SYSTEM GENERATION PROCESS
CHAPTER 8
RUN SYSGEN PROGRAM, ANSWER DIALOGUE QUESTIONS
CHAPTER 9 OR 10
PUT APPROPRIATE FILES ON APPROPRIATE VOLUMES
ASSEMBLE SYSTEM SOURCE AND CONDITIONAL FILES
LINK OBJECT FILES
STORE OBJ FILES AND LINK MAPS

1.1.1 What Is Your Hardware Configuration?

Your hardware configuration is an important consideration for three reasons:

1. Installation procedures vary according to the hardware configuration.
2. System generation cannot be performed on all hardware configurations.
3. System generation procedures vary according to the hardware configuration.

Chapters 3 through 7 contain detailed procedures for installing RT–11 systems on certain common hardware configurations. Unless you have an unusual configuration, read only the chapter intended for your configuration. (See Table 1–6 to identify your reading path.)

1.1.2 Which Monitor(s) Do You Need?

You have to decide which monitor or monitors you need in order to know what steps to follow in getting started. Your distribution kit includes a single-job monitor, a base-line single-job monitor, and a foreground/background monitor. The extended memory monitor is available only through the system generation process. Sections 1.1.2.1 through 1.1.2.4 describe the various monitors, Table 1–1 summarizes the monitor differences, and Table 1–2 lists the features available in the monitors. The sections and the two tables give you the information you need to make a decision about your monitor requirements.

Table 1-1: Monitor Differences

Characteristics	Monitor			
	BL	SJ	FB	XM[1]
Memory Limits	12 to 30K words	12 to 30K words	16 to 30K words	32 to 124K words
Size of resident monitor	1857 words	1996 words	4220 words	varies
Support FG Job	no	no	yes	yes
Multi-Terminal Support	no	optional[2]	optional[2]	optional[2]
Timer Facilities (.MRKT, .CMKT—including midnight date/time rollover)	no	optional[2]	yes	yes
Memory Parity Support	no	optional[2]	optional[2]	optional[2]
BATCH Support	no	optional[3]	optional[3]	optional[3]
VT11/VS60 Support	no	optional[2]	optional[2]	optional[2]
Error Messages on Fatal System Errors	no	optional[2]	yes	yes
Devices Supported	RK05 disk, RX01 diskette, DECtape, line printer, serial line printer, terminal, paper tape, null device	all	all	all

[1] The XM monitor is not a standard monitor. You must perform the system generation process to create this monitor.

[2] You can perform the system generation process to create a special monitor and handlers that include this support.

[3] The standard monitors include BATCH support, but if you select a particular monitor during the system generation process, this support is not enabled (unless you also respond "yes" to the BATCH question).

Table 1–2: Features Available in Distributed Monitors

	Monitor		
Feature	BL	SJ	FB
Timer support (.MRKT and .CMKT—including midnight date/time rollover)	no	no	yes
Error messages on system I/O errors	no	yes	yes
Idle loop light pattern	no	no	no
Multi-terminal support	no	no	no
60HZ Clock frequency	yes	yes	yes
50HZ Clock frequency instead of 60HZ	no	no	no
Programmable clock instead of line as system clock	no	no	no
Startup command file	yes	yes	yes
Memory parity support	no	no	no
System job support	no	no	no
Month rollover support	no	no	no
Keyboard monitor command subset	no	no	no
Floating point support	no	yes	yes
Error message on power fail	no	yes	yes
BATCH	no	yes	yes
Error logging	no	no	no
RK11/RK05[1]	yes	yes	yes
RL01/RL02[1]	no	yes — 2 drives	yes—2 drives
RJS03/RJS04[1]	no	RJS03—can patch for RJS04	RJS03—can patch for RJS04
RK06/07[1]	no	yes	yes
RF11[2]	no	yes	yes
RP11/RPR02/RP03[3]	no	yes	yes
RX01[1]	yes	yes	yes
RX02[1]	no	yes	yes
DECtape[1]	yes	yes	yes
DECtape II[1]	no	yes	yes
PDT–11 intelligent terminal	no	yes	yes
File-structured magtape—TM11	no	yes — 2 units[3]	yes—2 units[3]

(continued on next page)

Table 1–2: Features Available in Distributed Monitors (Cont.)

	Monitor		
Feature	BL	SJ	FB
File-structured magtape—TJU16	no	yes—2 units[3]	yes—2 units[3]
File-structured magtape—TS11	no	yes—1 unit[3]	yes—1 unit[3]
Hardware magtape TM11[4]	no	yes	yes
Hardware magtape TJU16[4]	no	yes	yes
Hardware magtape TS11[4]	no	yes	yes
TA11 cassette[1]	no	yes	yes
VT11/VS60	no	yes VT11	yes VT11
Line printer	yes	yes	yes
Serial line printer	yes	yes	yes
High-speed reader/punch	yes	yes	yes
CR11 card reader	no	yes	yes
Null handler	yes	yes	yes
Extra device slots	no	no	no

[1] See Section 2.8.13 to install devices.

[2] RT–11 no longer supports RF11 disks; however, the handler is included in the software kit.

[3] You can perform the system generation process to create a monitor and handlers that support more than two magtape units.

[4] See Section 2.8.13.2 to install hardware magtape support.

1.1.2.1 Base-line Single-Job Monitor — The base-line single-job monitor (BL) is a version of the single-job monitor. The BL monitor does not support optional monitor and device functions. It has the minimum residency requirement of any RT–11 monitor and is intended for use in applications that require small monitor size and minimal executive support.

The base-line monitor supports MAC8K, the special overlaid MACRO assembler for applications that require a small assembler.

The differences between the base-line monitor and the standard single-job monitor are summarized in Table 1–3.

Table 1-3: Base-line Single-Job Versus Single-Job Monitor

Base-line SJ	Standard SJ
Supports only the following devices: RK05 disk, RX01 diskette, DECtape, null device, line printer, serial line printer, terminal, paper tape, and PDT-11 intelligent terminal.	Supports all RT-11 devices, with almost all normal devices installed.
Does not support graphic display.	Supports graphic display terminal (VT11/VS60) as ASCII console terminal.
System halts on power failure or system device I/O error.	System prints error message on power failure or system device I/O error.
Supports only one terminal.	Can be generated to support more than one terminal.
Does not support BATCH.	Supports BATCH.
Minimal memory and system device requirements for monitor.	Requires slightly larger memory and system device for monitor.
Does not include floating point support.	Includes floating point support.

If your application program is very large and you require the smallest possible monitor, the base-line single-job monitor may be your best choice. The base-line monitor can perform all the system commands and can run most of the utilities. For highly interactive applications, you should use the standard single-job monitor. The more complete error processing and device support available in the standard single-job monitor provide a base that is easier to use and more flexible.

1.1.2.2 Single-Job Monitor — Of all the standard RT-11 monitors, the single-job monitor (SJ) has the fastest response times at interrupt and keyboard levels and the minimum memory requirements (except for the base-line single-job monitor, which is a stripped down version of the single-job monitor). If your application involves interactive program development, maximum-throughput real-time data acquisition, or continuous execution of a single end-user application program, the single-job monitor is the best choice.

The single-job monitor supports all hardware devices (except memory management hardware) and runs all the system utilities except QUEUE and error logging. It runs in any supported configuration with at least 12K words of memory but cannot make use of more than 28K words of memory (30K on an LSI-11 processor).

1.1.2.3 Foreground/Background Monitor — The foreground/background monitor (FB) is the smallest RT-11 monitor that supports multiprogramming. It allows you to execute a completely independent foreground job at a higher software priority level than the background, while you use the remaining system facilities to support the background. The RT-11 foreground job is not intended for a two-user time-sharing system. Rather, it best supports a stable, event-driven real-time or I/O application that can execute with a minimum of user interaction while the bulk of the system's business is conducted in the background.

The background in the foreground/background environment appears just like the single-job monitor; all the facilities available to you as an SJ user are available to you as an FB user in the background.

The FB monitor provides you with more features than the SJ monitor. These include:

1. The ability to run a foreground job
2. Enhanced terminal service
3. Extended executive facilities for real-time applications

The minimum memory requirement for the FB monitor is 16K words. The FB monitor resides in 8K or fewer words of memory and cannot use more than 28K words of memory (30K on an LSI–11 processor).

If your application includes the need for a software priority real-time application to run concurrently with normal system development and data-processing applications, the FB monitor is the suitable choice. If you do not require concurrent real-time execution, you can conserve system resources by using the SJ monitor.

1.1.2.4 Extended Memory Monitor — The extended memory monitor (XM) is the largest and most powerful of the RT–11 executives. It has all the facilities of the FB monitor; in addition, it can support up to 124K words of memory. The extended memory monitor provides a set of programmed requests that allow advanced applications to make use of additional memory above 28K (using the PDP–11 memory management unit).

The XM monitor is significantly larger than the FB monitor. The XM monitor's requirements that device handlers and the USR be resident add even more to the monitor memory overhead. Consequently, if you do not have more than 28K words of memory, choose one of the other monitors. Also, if your application involves program development or execution of an end-user application that does not make use of extended memory facilities, you should opt for the FB or SJ monitor.

The XM monitor requires 32K words of memory, a memory management unit (KT11 hardware or a PDP–11/23 processor), and the Extended Instruction Set (EIS) option to operate. The XM monitor is not distributed on the RT–11 kit. You must perform the system generation process to create this monitor and its device handlers.

1.1.3 Which Features Do You Need?

The following sections summarize the RT–11 features available, so that you can decide which features you need and how to obtain them. You can secure some of the features through simple customization procedures during a standard installation process. Other features are available only when you go through the system generation process.

1.1.3.1 Is Installation All You Need to Do? — You need to decide whether installation is all you need to do or whether you will also need to perform the system generation process. Table 1–4 summarizes the features that you can add to your system by performing simple customizations during installation. Chapter 2 describes these customizations in detail and provides instructions for implementing them. Many users will not need to make any of these customizations.

Table 1–4: Features Available Through Simple Customizations

Feature	Section	Procedure
Adding subroutines to SYSLIB.OBJ	2.3.5	To add subroutines to the default system library, SYSLIB.OBJ, use the librarian. The RT–11 linker searches SYSLIB to resolve undefined globals at the end of a link operation. Normally, you should customize SYSLIB to contain the system subroutines, FORTRAN OTS routines, and application-specific subroutines.
Substituting RJS04 support for RJS03	2.8.1	RT–11 RJS03/4 support is initialized for RJS03 when distributed. To use RJS04, you must patch the DS handler to change the handler's device size entry.
Changing characters SRCCOM uses to indicate insertions and deletions	2.8.2	Normally, when you use the DIFFERENCES/CHANGEBAR command or SRCCOM to compare two files, SRCCOM uses the vertical bar character to indicate insertions and the bullet (alphabetic o) character to indicate deletions. You can patch SRCCOM to use different characters.
Changing CSR addresses that FORMAT utility program supports	2.8.3	If any devices that FORMAT supports are installed at nonstandard addresses, you must patch FORMAT to change the addresses.
✓ Changing default output device from line printer to terminal	2.8.4	If your configuration does not include a line printer, you can change the default output device (which certain monitor commands use) from line printer to terminal. To make this change, you can edit the startup command file.

(continued on next page)

Table 1–4: Features Available Through Simple Customizations (Cont.)

Feature	Section	Procedure
Changing location of VT11/VS60 floating vectors	2.8.5	If you add certain other devices to a configuration that includes a VT11 or VS60, you must patch the monitor to move the VT11 or VS60 vector addresses.
Changing the number of directory columns	2.8.6	When you use the DIRECTORY command, the directory prints in two columns. When you use the DIRECTORY/BRIEF command, the directory contains five columns with less information. You can patch DIR to change the default number of columns.
Changing the number of /Q p-sects LINK allows	2.8.7	You can patch LINK to change the number of absolute base address p-sects that LINK allows. Normally, LINK's /Q option lets you specify the absolute base addresses of up to eight p-sects in your program.
Changing the size of LINK's library module list	2.8.8	To change the default size of LINK's list of library modules, you can patch LINK. LINK normally creates a list of 252 (octal) modules to be included from libraries during the link operation. If you are a DIBOL user, you must make this patch.
Changing the size of the queue	2.8.9	You can patch QUEUE to change the size of the work file QUEUE uses to queue files to be sent to a device. If you increase the size of the work file, QUEUE can handle more files at one time.
Modifying EDIT	2.8.10	*Size of text window*: In certain applications using VT11/VS60 terminals, the text window overflows onto the scrolled editing commands, making a portion of the screen difficult to read. You can alleviate this problem by patching EDIT to reduce the size of the text window. *Terminals with nonstandard ESCAPE code*: If you have a terminal with nonstandard ESCAPE code (that is, the terminal generates 175 octal or 176 octal rather than 33 octal), you must patch EDIT so that it operates correctly.
Setting initial console fill characteristics	2.8.11	The LA30S DECwriter requires filler characters to follow each carriage return, and the 600, 1200, and 2400 baud VT05 terminals require filler characters to follow each line feed. If your configuration includes any of these terminals, you must install a patch.

(continued on next page)

Table 1–4: Features Available Through Simple Customizations (Cont.)

Feature	Section	Procedure
Extracting the overlay handlers from SYSLIB.OBJ	2.8.12	If you intend to use overlaid programs, you need the overlay handlers that are included in SYSLIB. If you are a MACRO-only user and need only the overlay handlers, you can extract them from SYSLIB for use as a separate library.
Installing other devices	2.8.13	When you boot the system device, the bootstrap installs the devices in your hardware configuration if the appropriate device handler is on the system device and if there are enough device slots. If you want to install a device that the bootstrap does not install, use the REMOVE and INSTALL commands. You can also: *Change LP, LS, or DD addresses*: If any of these devices is installed at nonstandard CSR and vector addresses, you can use the monitor SET command to change the addresses. *Install hardware magtape*: If you need hardware magtape support, you can rename the hardware handler to MT.SYS, MM.SYS, or MS.SYS. *Set magtape parity and density*: If you need to set magtape parity or density (other than the standard 800 bpi, odd parity), you can use the monitor SET command for TM11 or TJU16 magtape (but not TS11). *"Change" RP02 support to RP03*: To use RP03 disk drives, you must think of each RP03 drive as two logical units of 40000 blocks.
Modifying BATCH to save space	2.8.14	If you need to save disk space, you can store certain system programs on DK: rather than on SY: and cause BATCH to access them there. Patch BATCH so that it invokes system programs with the monitor RUN command (which assumes DK: as the default) rather than with the R command (which always uses SY:).
Modifying LINK to change default SYSLIB device	2.8.15	By patching the linker, you can change the device on which the default system library, SYSLIB.OBJ, resides.
Modifying the help text	2.8.16	To change the help text that prints when you use the HELP command, you must create your own help text macro library, process that library with LIBR, and copy the library and the file HELP.EXE to the same volume.

(continued on next page)

Table 1–4: Features Available Through Simple Customizations (Cont.)

	Feature	Section	Procedure
X	Modifying listing page length in MACRO and CREF	2.8.17	If your line printer uses line printer paper that is not 10.5 inches long, or if you do not have a line printer, you can patch MACRO and CREF to change the listing page length from the standard 60 lines.
?	Preventing fatal system errors from causing a reset	2.8.18	You can patch the monitor to prevent it from performing a hard reset of errors when a fatal system error occurs. The normal reset stops I/O, protecting media from being corrupted. Some hardware errors may be more easily diagnosed without this reset.
X	Running RT–11 in less memory than is available	2.8.19	You can install a patch if your application requires that RT–11 run in less memory than is available.
+	Setting upper limit on file size	2.8.20	If your application requires an upper limit to the size of a file, you can patch the monitor to set the limit you require.
X	Specifying 50-cycle clock rate	2.8.21	To use a 50-cycle clock rate rather than the 60-cycle clock of the standard monitors, you must patch the monitor.
X	Specifying number of RF11 platters	2.8.22	RT–11 RF11 fixed-head disk support is initialized for one platter when distributed. To use more than one platter, patch the RF handler to change the handler's device size entry.
X	Using CAPS–11 to load files	2.8.23	If you need to use CAPS–11 to load RT–11 files, you must patch the cassette handler, CT.SYS, to alter header records.
?	Setting upper limit on memory size	2.8.24	If your PDP–11 does not generate a bus timeout trap, RT–11 does not load into memory properly. You can patch the monitor to cause the bootstrap to never look for more than 28K words of memory.
X	Suppressing the bootstrap message	2.8.25	If you want to prevent the monitor idenfication message from printing when you bootstrap a monitor, you can patch that monitor.
+	Suppressing the startup indirect command file	2.8.26	If you want to prevent the startup command file from executing when you bootstrap a monitor, you can patch that monitor.
+	Suppressing the startup indirect command file echo	2.8.27	If you want the startup indirect command file to execute when you bootstrap a monitor but you do not want the command lines in the file to echo on the terminal (appear on the terminal), you can patch the monitor.
LATER VOICE?	Changing the bootstrap message	2.8.28	If you want to change the monitor identification message that appears when you bootstrap a monitor, you can patch that monitor.

(continued on next page)

Table 1–4: Features Available Through Simple Customizations (Cont.)

Feature	Section	Procedure
Changing the default device for indirect command files	2.8.29	If you want to change the default device for indirect command files, you can patch the monitor. Normally, the monitor looks for the command file on DK:.
Changing the default file type for indirect command files	2.8.30	If you want to change the default file type for indirect command files, you can patch the monitor. Normally, indirect command files have the default file type .COM.
Changing the default device for the FRUN command	2.8.31	If you want to change the default device for the FRUN command, you can patch the monitor. Normally, the monitor looks for the foreground program on device DK: when you type FRUN filnam.
Changing the default file type for the FRUN command	2.8.32	If you want to change the default file type for the FRUN command, you can patch the monitor. Normally, the default file type for foreground programs is .REL.
Changing the default device for the EDIT command	2.8.33	If you want to change the default device for the EDIT command, you can patch the monitor. Normally, the monitor looks for EDIT.SAV on device DK:.
Changing the default file name for the EDIT/EDIT command	2.8.34	If you want the monitor to run a program other than EDIT.SAV when you type the EDIT command, you can patch the monitor to change the default file name for the EDIT/EDIT command.
Changing the default file name for the EDIT/TECO command	2.8.35	If you want the monitor to run a program other than TECO.SAV when you type the EDIT/TECO command, you can patch the monitor to change the default file name for the EDIT/TECO command.
Using examine and deposit above the background job	2.8.36	If you want to be able to examine and modify the monitor and the I/O page, you can patch the monitor to remove a restriction on the use of the examine (E) and deposit (D) commands. Normally, the monitor allows you to examine and modify only locations inside the background job's area.
Changing the default device for QUEMAN	2.8.37	If you want to send a file to a device other than LP: when you use the PRINT command in an FB system (with QUEUE running as a foreground job), you can patch the monitor to change the default. The patch also changes the default for the DELETE/ENTRY command.

(continued on next page)

Table 1–4: Features Available Through Simple Customizations (Cont.)

Feature	Section	Procedure
Changing the indirect command file nesting depth	2.8.38	If you want to change the depth to which you can nest indirect command files, you can patch the monitor. Normally, RT–11 allows you to nest indirect command files three deep.
Changing the threshold for resuming output-stalled jobs	2.8.39	If you have a foreground job that produces much terminal output and stalls frequently (waiting for room in the terminal output buffer), you can patch the monitor to improve throughput.
Changing the default number of directory segments	2.8.40	If you want DUP to use a different default number of directory segments when you initialize volumes, you can patch DUP. DUP normally uses the number of directory segments specified in a table in DUP.
Changing MACRO–11 listing control directive defaults	2.8.41	If you want to change the MACRO–11 listing control directive defaults from .LIST (list) to .NLIST (no list), you can patch a word in MACRO–11.
Changing MACRO–11 function control directive defaults	2.8.42	If you want to change the MACRO–11 function control directive defaults from .ENABL (enable) to .DISABL (disable), you can patch a word in MACRO–11.

1.1.3.2 Do You Need to Perform the System Generation Process? — Table 1–5 summarizes the features that you can add to your system only by performing the system generation process. Refer to the SYSGEN dialogue itself, reproduced in Chapter 8 (along with descriptive text), for further information about these features. You may need to perform the system generation process also if you want to remove features to reduce the size of the monitor and improve response time.

You should not attempt to perform the system generation process unless your hardware configuration meets certain requirements. DIGITAL *supports* automatic system generation (under license) only on a system with at least 16K words of memory and 2000 free blocks of disk storage. The minimum configuration that DIGITAL *recommends* for system generation is a system with at least two disk drives and 24K words of memory. DIGITAL also supports system generation on dual-diskette systems (with at least 28K words of memory and a line printer or hard-copy terminal) if you use the manual method described in Chapter 10. However, DIGITAL does not recommend this very lengthy procedure.

The numbers in Table 1–5 correspond to the actual numbered SYSGEN questions. Chapters 9 and 10 provide step-by-step instructions for performing system generation on several types of configurations.

Table 1–5: Features Available Only Through System Generation Process

	Feature	SYSGEN Question Number	Function
✗	Asynchronous terminal status	13	Provides a program with the updated status of terminals in multi-terminal systems. MU BASIC–11 and CTS–300 require this support.
✗	Device I/O time-out	7	Permits device handlers to issue a mark time programmed request. DECNET applications require this support, but RT–11 does not.
✗	Double density only RX02	39	Permits you to use only double-density RX02 diskettes on the system. This feature makes the RX02 handler smaller.
✗	Error logging	29	Keeps a statistical record of device, memory parity, and memory cache errors. An error logging job is created when you select this support. The EL job retrieves information that is later available to you in summary report format. Only the FB and XM monitors support error logging.
✗	Extended memory monitor	5	Allows you to extend programs' logical addressing space. The extended memory monitor, a version of the FB monitor that supports additional programmed requests, requires memory management hardware and the Extended Instruction Set option.
✓ ?	Extra device slots	104	Permits you to add devices to the system after it is built. The number of logical assignments you can make is equal to the number of devices plus empty device slots in the system.
✗	Idle loop light pattern	11	Causes FB or XM monitor's scheduler idle loop to display a moving light pattern on certain processors.
?	Keyboard monitor command subsets	18, 19, 20, 21	Allows you to choose one, two, or three subsets of the keyboard monitor commands instead of all the commands.
?	Memory parity	26	Causes the system to print an error message when a memory parity error occurs if your configuration includes memory parity hardware. If you have this hardware but do not select this support, the system halts when memory errors occur.

(continued on next page)

Table 1–5: Features Available Only Through System Generation Process (Cont.)

	Feature	SYSGEN Question Number	Function
X	Month rollover	17	Adds support that automatically rolls over the date at the end of the month and the end of the year. Normally, you must reset the date and time. This support is useful for applications that run continuously and over a long period of time.
LATER	Multi-terminal	12	Permits you to use two or more terminals with SJ, FB, or XM monitor. MU BASIC–11 requires this support.
X	Multi-terminal time-out	14	Causes the monitor to reset at regular intervals any terminal that goes off line. This support minimizes the impact of static in multi-terminal systems. MU BASIC–11 requires this support.
X	Programmable clock as system clock	23	Allows you to substitute as the system clock the KW11–P programmable clock for the usual line clock. However, the programmable clock would not then be available for program use.
X	Ring buffer size	15, 16	Allows you to change the size of the input and output ring buffers. The input ring is a buffer in the monitor that holds characters you type at a terminal until a program requests them. The output ring is a buffer in the monitor that holds characters until the terminal can print them. The default input ring buffer size is 134. characters, and the default output ring buffer size is 40. characters.
LATER	Second RX01, RX02, or TU58 controller	32, 38, 45	Adds support for a second RX01, RX02, or TU58 controller, allowing a total of four units in the configuration instead of the usual two.
✓	SJ message on system I/O errors	8	Causes the SJ monitor to issue an error message instead of simply halting. This feature helps to reduce confusion when an error occurs. The FB and XM monitors always issue error messages.

(continued on next page)

Table 1–5: Features Available Only Through System Generation Process (Cont.)

Feature	SYSGEN Question Number	Function
SJ timer	6	Configures the SJ monitor to support mark time and cancel mark time programmed requests. Otherwise, only the FB and XM monitors support these requests, which provide timer capabilities.
.SPCPS programmed request	10	The save/set main-line PC and PS (.SPCPS) programmed request changes the flow of control of main-line code by saving the PC and PS and changing the main-line PC to a new value. This support can be generated for only the FB and XM monitors. .SPCPS is especially useful to control switching among users in multi-user applications.
System jobs	9	Assembles the FB or XM monitor to support as many as eight simultaneously active jobs instead of the usual two. Both the error logging subsystem and the device queue program (QUEUE) can run as system jobs.

1.2 Choosing a Reading Path

The modular organization of this manual is intended to limit your reading to those chapters that answer your needs. Table 1–6 gives you the information you need to select these chapters. To use the table, find your distribution device and your target system device (the device that you will want to bootstrap and that will contain the monitors and system components), and answer the question about system generation. You will then find in the rightmost column the relevant chapters for your configuration.

Also skim the table of contents for an overview of the additional information offered in the appendixes.

NOTE

> If you are unfamiliar with RT–11 software, you should probably use the distributed monitors uncustomized for awhile. Once you have gained familiarity with the system, you will be better equipped to perform customizations or system generation procedures. Be aware, however, that some customiza-

tions may be essential for your particular application, and, consequently, you may experience problems when you use software without needed support. If you are an inexperienced user, be sure to study Tables 1–4 and 1–5, but you may want to skip Section 2.8 (Customizations) until you have gained some experience.

Table 1–6: Chapters You Should Read

Distribution Medium	System Device in Target System	System Generatioı Process	Chapters
Single- Density Diskette RX01 PDT–11/150	Single-density diskette	NO	1, 2, 3
		YES	1, 2, 3, 8, 10
	Hard disk RL01 RL02 RK05 RK06 RK07 RP02 RP03 RJS03 RJS04 RF11	NO	1, 2, 4
		YES	1, 2, 4, 8, 9
DECtape II Cartridge TU58 PDT–11/130	DECtape II cartridge	NO	1, 2, 3
	Hard disk RL01 RL02 RK05 RK06 RK07 RP02 RP03 RJS03 RJS04 RF11	NO	1, 2, 4
		YES	1, 2, 4, 8, 9

(continued on next page)

Table 1–6: Chapters You Should Read (Cont.)

Distribution Medium	System Device in Target System	System Generation Process	Chapters
Hard Disk RL01 RL02 RK05	Hard disk RL01 RL02 RK05 RK06 RK07 RP02 RP03 RJS03 RJS04 RF11	NO	1, 2, 5
		YES	1, 2, 5, 8, 9
Double-Density Diskette RX02	Double-density diskette	NO	1, 2, 6
		YES	1, 2, 6, 8, 10
Magtape TM11 TJU16	Hard disk RL02 RL02 RK05 RK06 RK07 RP02 RP03	NO	1, 2, 7
		YES	1, 2, 7, 8, 9

Part II
Installation

Chapter 2
Preparing for Installation

You should take certain steps, summarized in the following list, before you actually begin to install your system.

1. Survey the general installation procedure
2. Study the contents of your software kit
3. Select the components you need in your working system
4. Plan the arrangement of components
5. Identify any mandatory patches
6. Acquire sufficient media
7. Format any new media
8. Choose customizations you need to make

You can use the worksheet at the end of this chapter to jot down the components you select, the arrangement of files you plan, and any other information you need in order to perform the installation procedures.

2.1 Survey of Installation

Although the specific steps you must perform to install RT–11 depend on your configuration, all users must perform some *general* steps.

First, you must bootstrap the distribution volume and preserve the volume (or volumes) by making backup copies. Then you must install mandatory patches, if any have been published, in all affected components. You can then create the working system from chosen components (eliminating components you do not need) and install the bootstrap on the working system volume. At this point, you can make simple customizations that do not require performing the system generation process. When you have customized the system, you should compress the working system volume or volumes and preserve them on backup volumes. Once you are satisfied with the working system you create, you should test it to make sure that it works properly.

2.2 Software Kit Contents

Familiarity with the contents of the software kit you received helps to make the installation procedure go smoothly. You should learn which files are included in the kit and what each file does. Study Table 2–1 for a summary of the RT–11 components distributed in the kits and refer to the *RT–11 System User's Guide* for a thorough description of each utility. Appendix G of this manual maps the specific files on each volume in the various kits. RT–11 is distributed on RK05 disk, RL01 disk, RL02 disk, RX01 diskette, RX02 diskette, DECtape II cartridge, and magtape. The organization of files on volumes depends on the distribution medium.

Table 2–1: RT–11 Software Components

Type of Software	Description
Monitors	
RT11BL.SYS	Base-line single-job monitor
RT11SJ.SYS	Single-job monitor
RT11FB.SYS	Foreground/background monitor
Device Handlers	
CR.SYS	Card reader handler for SJ, FB monitors
CT.SYS	TA11 cassette handler for SJ, FB monitors
DD.SYS	DECtape II handler for SJ, FB monitors
DL.SYS	RL11/RL01/RL02 handler for SJ, FB monitors
DM.SYS	RK611/RK06/RK07 handler for SJ, FB monitors
DP.SYS	RP11/RPR02/RP03 handler for SJ, FB monitors
DS.SYS	RJS03/4 handler for SJ, FB monitors
DT.SYS	DECtape handler for SJ, FB monitors
DX.SYS	RX11/RX01 Single-density diskette handler for SJ, FB monitors
DY.SYS	RX211/RX02 handler for SJ, FB monitors
LP.SYS	Parallel line printer handler for SJ, FB monitors
LS.SYS	Serial line printer handler for SJ, FB monitors
MM.SYS	File-structured TJU16 handler for SJ, FB monitors

(continued on next page)

Table 2–1: RT–11 Software Components (Cont.)

Type of Software	Description
MMHD.SYS	Hardware TJU16 handler for SJ, FB monitors
MS.SYS	File-structured TS11 handler for SJ, FB monitors
MSHD.SYS	Hardware TS11 handler for SJ, FB monitors
MT.SYS	File-structured TM11 handler for SJ, FB monitors
MTHD.SYS	Hardware TM11 handler for SJ, FB monitors
NL.SYS	Null handler for SJ, FB monitors
PC.SYS	High-speed paper tape handler for SJ, FB monitors
PD.SYS	PDT–11 skeleton handler
RF.SYS	RF11 handler for SJ, FB monitors
RK.SYS	RK11/RK05 handler for SJ, FB monitors
TT.SYS	Terminal handler for SJ monitor
Other System Files	
BA.SYS	BATCH handler
SWAP.SYS	External monitor swap blocks
Utility Programs	
BINCOM.SAV	Binary compare utility
CREF.SAV	Cross-reference utility
DIR.SAV	Directory utility
DUMP.SAV	File dump utility
DUP.SAV	Disk maintenance utility
EDIT.SAV	Text editor
FILEX.SAV	Foreign file exchange utility
FORMAT.SAV	Disk formatting utility
HELP.SAV	Help utility
KED.SAV	Keypad editor for VT100 terminal
K52.SAV	Keypad editor for VT52 terminal
LIBR.SAV	Librarian
LINK.SAV	Linker
MACRO.SAV	MACRO assembler
MAC8K.SAV	Overlaid MACRO assembler for space constrained applications
MDUP.SAV	Magtape utility
MDUP.MM	Magtape bootstrap utility for TJU16
MDUP.MS	Magtape bootstrap utility for TS11

(continued on next page)

Table 2–1: RT–11 Software Components (Cont.)

Type of Software	Description
MDUP.MT	Magtape bootstrap utility for TM11
PATCH.SAV	Patching utility for V03B or earlier program memory images
PAT.SAV	Patching utility for program object modules
PIP.SAV	File transfer utility
QUEMAN.SAV	User interface with QUEUE utility
RESORC.SAV	System resource display utility
SIPP.SAV	Save image patch utility
SLP.SAV	Source language patch utility
SRCCOM.SAV	Source compare utility
SYSMAC.SML	System macro library
Miscellaneous	
BATCH.SAV	BATCH processor
DISMT1.COM	Indirect file to duplicate distribution tape 1 (bootable magtape)
DISMT2.COM	Indirect file to duplicate distribution tape 2
HELP.MLB	Help text
HELP.EXE	Help program
ERROUT.SAV	Error log program
QUEUE.REL	Device queue manager
README.TXT	Message text file
STARTF.COM	FB startup command file
STARTS.COM	SJ startup command file
V4USER.TXT	Distribution kit message text file
RTBL.MAP	Distributed base-line single-job monitor link map
RTSJ.MAP	Distributed single-job monitor link map
RTFB.MAP	Distributed foreground/background monitor link map
System Generation Files	
SYSGEN.SAV	System generation dialogue utility
SYSGEN.CND	SYSGEN input file
SYSTBL.CND	SYSGEN input file
Debuggers	
ODT.OBJ	Debugging aid
VDT.OBJ	Debugging aid for virtual jobs and multi-terminal jobs

(continued on next page)

Table 2–1: RT–11 Software Components (Cont.)

Type of Software	Description
Graphics Software	
VTMAC MAC	Display handler macro file
VTHDLR.OBJ	VT11 VS60 display handler
PLOT55.OBJ	VT55 graphics terminal software
TEST55.FOR	Demonstration program for VT55
Libraries and Subroutines	
GETSTR FOR	FORTRAN subroutine source
PUTSTR FOR	FORTRAN subroutine source
SYSLIB.OBJ	System FORTRAN-callable subroutines
SYSMAC.MAC	System macro library source
Bootstraps	
MBOOT.BOT	Magtape primary bootstrap
MSBOOT.BOT	Magtape secondary bootstrap
Demonstration Programs	
DEMOBG MAC	Demonstration source
DEMOED.TXT	Demonstration source
DEMOFG.MAC	Demonstration source
DEMOF1 FOR	Demonstration source
DEMOX1.MAC	Demonstration source
Source Files	
BA.MAC	BATCH handler source file for system generation
BSTRAP.MAC	Bootstrap source file for system generation
CR.MAC	Card reader handler source file for system generation
CT.MAC	Cassette handler source file for system generation
DD.MAC	DECtape II handler source file for system generation
DL.MAC	RL01/2 handler source file for system generation
DM.MAC	RK06/07 handler source file for system generation
DP.MAC	RP11 handler source file for system generation
DS.MAC	RJS03/4 handler source file for system generation
DT.MAC	DECtape handler source file for system generation
DX.MAC	RX01 handler source file for system generation

(continued on next page)

Table 2–1: RT–11 Software Components (Cont.)

Type of Software	Description
DY.MAC	RX02 handler source file for system generation
EDTGBL.MAC	Monitor edit log and global definition file for system generation
ELCOPY.MAC	Error log job source file for system generation
ELINIT.MAC	Error log job source file for system generation
ELTASK.MAC	Error log job source file for system generation
ERROUT.MAC	Error log job source file for system generation
ERRTXT.MAC	Error log job source file for system generation
FB.MAC	FB conditional source file for system generation
FSM.MAC	Magtape file support source file for system generation
KMON.MAC	Keyboard monitor source file for system generation
KMOVLY.MAC	Keyboard monitor overlay source file for system generation
LP.MAC	Parallel line printer handler source file for system generation
LS.MAC	Serial line printer handler source file for system generation
MTTEMT.MAC	Multi-terminal programmed request source file for system generation
MTTINT.MAC	Multi-terminal interrupt service source file for system generation
NL.MAC	Null handler source file for system generation
PC.MAC	High-speed paper tape handler source file for system generation
PD.MAC	PDT–11 handler source file for system generation
RF.MAC	RF11 handler source file for system generation
RK.MAC	RK05 handler source file for system generation
RMONFB.MAC	FB/XM resident monitor source file for system generation
RMONSJ.MAC	SJ resident monitor source file for system generation
SJ.MAC	SJ conditional source file for system generation
SYCND.BL	Conditional file DIGITAL used to build standard base-line single-job monitor
SYCND.DIS	Conditional file DIGITAL used to build standard SJ, FB monitors
SYCND.HD	Hardware magtape handler conditional file
SYSTBL.BL	Conditional file DIGITAL used to build standard base-line single-job monitor

(continued on next page)

Table 2-1: RT-11 Software Components (Cont.)

Type of Software	Description
SYSTBL.DIS	Conditional file DIGITAL used to build standard SJ, FB monitors
TJ.MAC	TJU16 handler source file for system generation
TM.MAC	TM11 handler source file for system generation
TS.MAC	TS11 handler source file for system generation
TT.MAC	TT.SYS source file for system generation
USR.MAC	USR source file for system generation
XM.MAC	XM conditional source file for system generation
XMSUBS.MAC	XM monitor subroutines for system generation
Unsupported Software	
TECO.SAV	TECO editor
VT52.TEC	TECO macro
EDIT.TEC	TECO macro
LOCAL.TEC	TECO macro
VTEDIT.TEC	TECO macro
SEARCH.TEC	TECO macro
TYPE.TEC	TECO macro
VEG.TEC	TECO macro
INSERT.TEC	TECO macro
SORT.TEC	TECO macro
TECO.TEC	TECO macro
SQU.TEC	TECO macro
TECO.INI	TECO macro
SPEED.SAV	Utility for setting terminal baud rate

2.3 Selecting the Components for Your Working System

Because space on volumes is limited, you should include in your system only those components that are essential to your application. To decide which components these should be, study the space limitations of your particular device, then consider the suggestions given for each software component (Sections 2.3.1 through 2.3.16).

Before selecting components, you should become familiar with your medium's space limitations. In particular, examine the software kit maps in Appendix G to establish how many blocks are occupied by the components residing on each volume and how much free space is available on the volume.

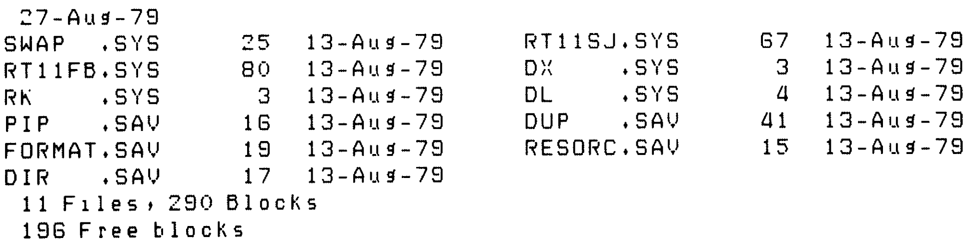
Table 2–2: Device Size

Device	Device Name	Size in Blocks
RX01 Diskette	DX	494
PDT–11/150 Diskette	PD	494
DECtape II cartridge	DD	512
PDT–11/130 cartridge	PD	512
DECtape	DT	578
RX02 Diskette	DY	988
RF11 Disk	RF	1024 per platter
RJS03/4 Disk	DS	1024/ 2048
RK05 Disk	RK	4800
RL01/2 Disk	DL	10210/ 20450
RP02/3 Disk	DP	40000/ 80000[1]
RK06/7 Disk	DM	27102/ 53724

[1] If your configuration includes RP03 support, you access each RP03 drive as two logical units of 40000 blocks.

You should keep in mind the number of blocks various components occupy when you select the components for your working system. Also keep them in mind when you decide how to arrange these components on volumes. In addition, remember that you may need space for data storage on your system volume and other volumes in the working system.

Although you will probably find it more efficient to select components for the working system before actually starting installation, you can also examine volume directories during the installation process to get the component size information. Once you have booted the system, begun the installation process, and backed up the distribution media, you can examine a backup volume's directory (or the directories of each backup volume if there are more than one). To examine a directory, you will use the DIRECTORY command. The following is a sample directory.



```
 27-Aug-79
SWAP  .SYS    25  13-Aug-79     RT11SJ.SYS    67  13-Aug-79
RT11FB.SYS    80  13-Aug-79     DX    .SYS     3  13-Aug-79
RK    .SYS     3  13-Aug-79     DL    .SYS     4  13-Aug-79
PIP   .SAV    16  13-Aug-79     DUP   .SAV    41  13-Aug-79
FORMAT.SAV    19  13-Aug-79     RESORC.SAV    15  13-Aug-79
DIR   .SAV    17  13-Aug-79
 11 Files, 290 Blocks
 196 Free blocks
```

You can also find the total free space on an initialized blank volume by obtaining a directory of the volume.

Once you have examined the software kit maps, read the following sections. The suggestions given in these sections may help you make your selections. The following software components are described.

1. Monitor
2. SWAP.SYS File
3. System Device Handler
4. Other Device Handlers
5. Default System Library
6. Exercises
7. Help Package
8. Line Printer Handler
9. MACRO Assembler
10. MAC8K Assembler
11. Queue Package
12. Source Files
13. Startup Command File
14. Text Editor
15. Utility Programs
16. VTMAC.MAC and VTHDLR.OBJ

As you make selections, jot down your choices on the worksheet at the end of the chapter.

2.3.1 Monitor

In general, you need only one monitor on a working system. If you do need more than one monitor, build a different system volume for each monitor. When your system device is a large disk, you may have room for several monitors.

2.3.2 SWAP.SYS File

You need the file SWAP.SYS on a system volume to serve as temporary storage for part of a program in memory when KMON, the USR, or both must swap over that program. When KMON or the USR are no longer needed, the system reads this external swap file back into main memory.

2.3.3 System Device Handler

You need the system device handler on any system volume. For example, if you build a system with RK05 as the system device, the file RK.SYS must be on the system disk.

2.3.4 Other Device Handlers

In addition to the system device handler, you need the device handlers for the other peripheral devices in your configuration. You do not need handler files for any devices you do not have. You must have TT.SYS (the terminal handler) on your system volume if you plan to use a non-multiterminal SJ monitor. However, the FB, XM, and multiterminal SJ monitors each contain an integral, resident TT handler, so you need not have TT.SYS on your system volume if you plan to use any of those monitors.

2.3.5 Default System Library

To use the LINK utility program, you may need the file SYSLIB.OBJ. The file SYSLIB.OBJ is the default system library, which the RT-11 linker searches to resolve any undefined globals at the end of a link operation.

Generally, SYSLIB for your application should contain the system subroutines (the file SYSLIB.OBJ found in the software kit), installation-specific libraries of application subroutines, and the FORTRAN OTS routines. If it must contain application subroutines and language routines, you must customize SYSLIB to include these routines. If you intend to link overlaid files, you need SYSLIB, since it contains the overlay handlers. If you are a MACRO-only user, requiring only the overlay handlers in SYSLIB (and not the other routines), you can create a separate library containing only the overlay handlers by using the EXTRACT option with the LIBRARY command. You can name the new, smaller library SYSLIB. Section 2.8.12 describes the procedure for creating such a library. Section 2.8.12 also describes adding the overlay handlers to another library (for example, FORLIB or the DIBOL library) when the other library already exists.

To add modules to SYSLIB from a file xxxxxx.OBJ, use the following command, in which you need both the REMOVE and INSERT options. INSERT inserts the new library or module in the old one; REMOVE removes the duplicate global, $OVRH, from the library directory. You must remove $OVRH from the library directory for SYSLIB to function properly. This global appears in both overlay handlers, OHANDL and VHANDL, because VHANDL includes the program code found in OHANDL. VHANDL processes both unmapped and virtual overlays, while OHANDL processes only unmapped overlays.

```
.LIBRARY/INSERT/REMOVE SYSLIB.OBJ xxxxxx.OBJRET
Global? $OVRHRET
Global? RET
.
```

Refer to the FORTRAN user's guide for instructions on creating in-line code versions of GETSTR and PUTSTR and replacing threaded code modules in a library with in-line code modules.

2.3.6 Exercises

If you intend to perform the exercises in the *Introduction to RT–11*, you need the following components on the system volume:

SWAP.SYS
RT11SJ.SYS
RT11FB.SYS
xx.SYS (system device handler)
TT.SYS
LP.SYS or LS.SYS (if appropriate)
other handlers if appropriate
DIR.SAV
PIP.SAV
DUP.SAV
LINK.SAV
EDIT.SAV
SRCCOM.SAV
RESORC.SAV
LIBR.SAV
MACRO.SAV
CREF.SAV
SYSMAC.SML
ODT.OBJ
DEMOED.TXT
DEMOX1.MAC
DEMOBG.MAC
DEMOFG.MAC

To do FORTRAN exercises in the *Introduction to RT–11*, you also need:

FORTRA.SAV
SYSLIB.OBJ with FORLIB.OBJ included in it
DEMOF1.FOR

To do BASIC exercises in the *Introduction to RT–11*, you also need:

BASIC.SAV

If the system volume is single-density diskette or DECtape II, you will not have room for all components on the system volume. See Section 2.4.5 for the file arrangement you should use.

2.3.7 Help Package

Your distribution kit includes three help files: HELP.SAV, HELP.MLB, and HELP.EXE. HELP.EXE is the help program. To run, the help program requires a help text macro library file. DIGITAL supplies HELP.MLB, which is such a library. HELP.SAV consists of HELP.EXE and HELP.TXT

(HELP.MLB after the librarian LIBR processes it) merged into a ready-to-use utility. Unless you want to change the help text, HELP.SAV is the only file you need. You can delete the other help files from the working system.

Section 2.8.16 describes how to customize the help text to your specific needs.

2.3.8 Line Printer Handler

The software kit includes the line printer handler LP.SYS and the serial line printer handler LS.SYS. If your hardware configuration includes a serial line printer instead of the usual parallel line printer, you should include only LS.SYS in your working system.

NOTE

> If your hardware configuration is a PDT–11 Intelligent Terminal system, you must use the parallel line printer handler, LP.SYS, although your line printer may be a serial line printer.

You can use the serial line printer in the same way you use a parallel line printer. On your working system volume, rename the original LP.SYS file to something else in order to save it. Then, rename LS.SYS to LP.SYS.

```
.RENAME/SYSTEM LS.SYS LP.SYS(RET)
.BOOT SY:(RET)
```

You must reboot the system after renaming the handler so the proper handler will be installed in the device table.

Then, when you use the PRINT command, the system sends output to the serial line printer. Since you will store both the distribution kit and the backup volume(s) you make during the installation process, you always have copies of both handlers.

If your serial line printer is installed at nonstandard vector and control status register addresses, you can use the SET command to change the addresses. See Section 2.8.13.1. Note, however, that once you have renamed LS.SYS to LP.SYS, you must use the device name LP: in a SET command. The available SET options would still be those for the LS handler.

2.3.9 MACRO Assembler

If you intend to use the MACRO assembler, you need the files MACRO.SAV and SYSMAC.SML (the system macro library) on the system volume. See the *RT–11 System User's Guide* for a description of the assembly process.

2.3.10 MAC8K Assembler

The MAC8K assembler is a special, overlaid version of the MACRO assembler. It is smaller than MACRO; consequently, it is useful if your application program is very large and you have space problems if you use MACRO.

MAC8K is considerably slower than MACRO and has fewer features. You need include only one of these assemblers in your working system. DIGITAL recommends that you choose MACRO if possible. You need the file SYSMAC.SML (the system macro library) on the system volume when you use MAC8K.

2.3.11 Queue Package

The Queue package is a particularly useful utility that runs with the FB or XM monitors only. It sends files to any RT–11 device but is particularly useful in queueing files for printing.

To use the Queue package, you need the files QUEUE.REL (which queues and prints the files you specify) and QUEMAN.SAV (which processes command lines and sends the information to QUEUE.REL). When you run QUEUE, it creates a temporary work file (QUFILE.TMP) that contains the queue of files.

QUEUE runs as a foreground job. If you want to use QUEUE as a system job, you must perform the system generation process to generate support for system jobs and then give the name QUEUE.SYS to the system job (rename QUEUE.REL). Refer to the *RT–11 System User's Guide* for more information about the Queue package. Section 2.8.9 in this manual describes how to increase the size of the queue of files.

2.3.12 Source Files

Normally, you do not need any source files in your working system except demonstration sources. Otherwise, the source files (file type .MAC) are required only for the system generation process (See Chapters 8, 9, and 10).

You do need the demonstration sources in the working system to perform the exercises in Chapters 3 through 7 of this manual or in the *Introduction to RT–11*. Once you have finished these exercises, the demonstration source files are no longer useful and you can delete them.

2.3.13 Startup Command File

The standard RT–11 monitors look for a startup command file (STARTx.COM) whenever you bootstrap the system. If they find one, the monitors execute its commands (to print a message, assign devices, and so on). If the monitors do not find a startup command file, they print a message indicating that the startup file has not been found. You do not need a startup command file unless you want one; you can delete the distributed startup files or create your own. Refer to the *RT–11 System User's Guide* for more information about startup command files.

2.3.14 Text Editor

RT–11 gives you a choice of text editors. EDIT is a character-oriented text editor that you can use with either a hard-copy terminal or a video terminal. The Keypad Editor (KED or K52) is an easy to use, cursor-oriented editor that you can use only on a VT100 or VT52 video terminal. TECO is a powerful (although sometimes difficult-to-use) editor that DIGITAL does not support, although it is included in the software kit.

Generally, you need only one editor in your working system, although you may want to try out one or two of these editors before choosing the one you prefer. Refer to the *RT–11 System User's Guide,* the *PDP–11 Keypad Editor's User's Guide,* and the *TECO User's Guide* for descriptions of these editors.

You need the file EDIT.SAV if you choose EDIT as your text editor.

If you choose the Keypad Editor, the file you need depends on the type of terminal you have. You need the file KED.SAV if your terminal is a VT100 and K52.SAV if your terminal is a VT52.

If you choose TECO, you need TECO.SAV. You may also want to include some of the distributed TECO macros.

2.3.15 Utility Programs

You do not need any system utility programs you do not intend to use; but remember that most of the keyboard monitor commands need certain system programs in order to work. For example, PIP, DUP, and DIR are necessary for most of the keyboard commands to function. Appendix B of the *RT–11 System User's Guide* summarizes the keyboard monitor commands and lists the system utility programs that each command requires. Refer to the *RT–11 System User's Guide* also for descriptions of the functions of the remaining utility programs. If your system device is a large disk, you will probably find it useful to have most of the utility programs on your working system device. On the other hand, if you have a small system device, you will not have room for all the utilities. In addition to PIP, DUP, and DIR, you probably need to use an editor, LINK, FORMAT (for certain devices), and HELP frequently, and you may want them on the system volume.

However, the utilities you use less frequently can occupy a second volume, which you can insert in and run from Unit 1.

2.3.16 VTMAC.MAC and VTHDLR.OBJ

You do not need VTMAC.MAC and VTHDLR.OBJ unless you write applications that use a VT11/VS60 graphics display.

NOTE

> VTHDLR, which is the display handler, supports only VT11 hardware. Therefore, you may not be able to use all the features of a VS60 display.

2.4 Planning the Arrangement of Components

Planning the arrangement of components will enable you to use your system most efficiently. If your system volume is one of the smaller devices, this planning is especially important. Consider the following suggestions before you build a working system; they can help you plan an efficient system.

1. Assign the default device to the data device.
2. Create a separate utilities volume.
3. Create "bad blocks" on DECtape II cartridges to avoid excessive rewinds.
4. Create several system volumes.
5. Create volumes for use with the *Introduction to RT–11*.
6. Limit components on the system volume.

More details on each of these suggestions follow. (You can use the worksheet at the end of this chapter to record the arrangement as you plan it.)

2.4.1 Assigning the Default Device to the Data Device

Once you have installed the system and it is running, you can assign the default device DK: to the data device (Unit 1). This means that most temporary files, data files, and so on, default to the second unit, minimizing demand for system device capacity.

The command to assign the default device to Unit 1 is as follows:

```
ASSIGN xx1: DK:RET
```

You can include this command in your startup command file (see the *RT–11 System User's Guide*) to assign the default device to xx1: whenever you boot the system. Note that the procedures in this manual assume DK: is the system device unless indicated otherwise.

Be sure to make any adjustments in procedure if you assign the default.

2.4.2 Creating a Separate Utilities Volume

Create a separate utilities volume for the utility programs you expect to use infrequently. This technique will provide you with a system volume containing all the components necessary to execute the majority of keyboard commands and perform common program preparation functions. Then, when you need a seldom-used utility, you can insert the utilities volume in Unit 1 instead of the data volume. You can run a nonoverlaid utility directly from the utilities volume (or you can copy the utility temporarily to the system volume).

NOTE

> The PIP and DUP utilities must always reside on the system volume.

To run a nonoverlaid utility from the utilities volume, use the following commands, where xx is the physical device name and aaaaaa is the utility program's name.

```
.RUN xx1:aaaaaa(RET)
*
```

Replace the utilities volume in Unit 1 with the data volume, and issue the appropriate commands to the utility.

```
*(CTRL/C)
.
```

However, if you run an overlaid utility from Unit 1, the volume containing that utility must remain in Unit 1 at all times (DUP is the only exception to this rule). Therefore, you should generally include the overlaid utilities on your system volume. The overlaid components are: PIP, DUP, MACRO, LINK, LIBR, KED, K52, FORMAT, MAC8K, HELP, and TECO.

ODT.OBJ is also useful on the system volume to debug MACRO programs.

An example of this kind of arrangement of volumes into system volume and utilities volume follows:

System volume
SWAP.SYS
RT11SJ.SYS
DX.SYS
TT.SYS
LP.SYS
BA.SYS
PIP.SAV
DUP.SAV
DIR.SAV
EDIT.SAV
MACRO.SAV
LINK.SAV
LIBR.SAV
ODT.OBJ
HELP.SAV — required for the HELP command

Utilities volume
BATCH.SAV
FILEX.SAV
SIPP.SAV
BINCOM.SAV
SRCCOM.SAV

DUMP.SAV
RESORC.SAV
PAT.SAV
FORMAT.SAV

2.4.3 Creating "Bad Blocks" on TU58 DECtape II

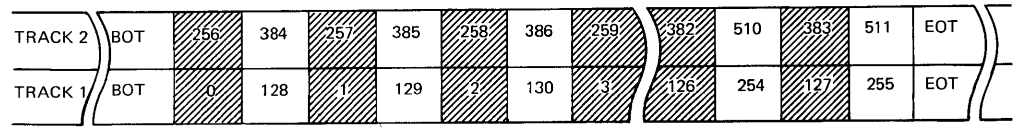
If your volumes are DECtape II cartridges, you may encounter slow response time due to excessive rewinds of the tape. Because of the way DECtape II stores data in records, you can actually improve system performance by creating dummy bad blocks in strategic locations. DECtape II writes data records in a specific sequence and pattern; to write an entire cartridge, for example, it performs the following actions.

1. Writes alternate data records on the first track.
2. Rewinds to return to the beginning of tape (BOT) mark.
3. Writes data records skipped on the first pass of the first track.
4. Rewinds.
5. Writes alternate data records on the second track.
6. Rewinds.
7. Writes data records skipped on the first pass of the second track.

Performance degradation occurs when a file (particularly a monitor file) overlaps from the end-of-tape to the beginning-of-tape—for example, it extends from the last portion of the second pass on track 1 to the first portion of the first pass on track 2.

There are three locations where you can create the dummy bad blocks. (Figure 2–1 illustrates the locations of blocks on the tape.) Create a bad block at the beginning of the second pass on track 1 (block 128.), at the beginning of the first pass on track 2 (block 256.), and at the beginning of the second pass on track 2 (block 384.). In this way, you can prevent the system from writing across rewinds, since RT–11 requires contiguous free space in which to write files. However, this technique prevents you from creating any file over 127 blocks long and also increases fragmentation.

Figure 2–1: Block Locations on DECtape II



To create these dummy bad blocks, insert an initialized blank volume (write enabled) in Unit 1 and type the following commands:

```
.CREATE/START:128. xx1:FIL1.BAD(RET)
.CREATE/START:256. xx1:FIL2.BAD(RET)
.CREATE/START:384. xx1:FIL3.BAD(RET)
.
```

Repeat this procedure on all the cartridges for your working system. Then, when you build your system, use the volumes on which you have created these bad blocks.

NOTE

> If you create these dummy bad blocks, you should consider them a permanent part of the cartridge (unless you reinitialize it). You can use the DELETE command to remove dummy bad blocks only if you have not compressed the cartridge with the SQUEEZE command. SQUEEZE renames bad blocks in such a way that you cannot type the file's name to delete the file.

2.4.4 Creating Several System Volumes

Create several system volumes, each devoted to a particular function. You can then change the system volume as normal job flow changes the functions you need. To change system volumes, wait for a logical stopping point in the job flow; do not arbitrarily remove the system volume in the middle of an operation.

2.4.5 Creating Volumes for Use with the *Introduction to RT–11*

If you intend to perform the exercises in *Introduction to RT–11*, you need certain components on your working system. If your system device is one of the small devices (RX01, RX02, PDT–11, or DECtape II), you need to build four volumes, which are listed with the files they contain. Be sure to copy the bootstrap to each volume.

System Volume
SWAP.SYS
RT11SJ.SYS
RT11FB.SYS
xx.SYS (system device handler)
TT.SYS
LP.SYS or LS.SYS (if appropriate)
PIP.SAV
DUP.SAV
DIR.SAV
LINK.SAV
EDIT.SAV
SRCCOM.SAV
RESORC.SAV
LIBR.SAV
ODT.OBJ
MACRO.SAV
SYSMAC.SML
CREF.SAV

DEMOED.TXT
DEMOF1.FOR
DEMOX1.MAC
DEMOBG.MAC
DEMOFG.MAC

FORTRAN Language Volume
SWAP.SYS
RT11SJ.SYS
xx.SYS (system device handler)
TT.SYS
LP.SYS or LS.SYS (if appropriate)
PIP.SAV
DUP.SAV
DIR.SAV
EDIT.SAV
FORTRA.SAV
DEMOF1.FOR

LINK Volume
SWAP.SYS
RT11SJ.SYS
xx.SYS (system device handler)
LP.SYS or LS.SYS (if appropriate)
TT.SYS
PIP.SAV
DUP.SAV
DIR.SAV
LINK.SAV
SYSLIB.OBJ (including FORLIB.OBJ)

BASIC Volume
SWAP.SYS
RT11SJ.SYS
xx.SYS (system device handler)
LP.SYS or LS.SYS (if appropriate)
TT.SYS
PIP.SAV
DUP.SAV
DIR.SAV
LINK.SAV
EDIT.SAV
BASIC.SAV
SYSLIB.OBJ

2.4.6 Limiting Components on System Volume

Limit the system volume to necessary and frequently used system components. Place on the system volume only one monitor file and only the handler files for devices in your configuration. For example, a typical DECtape II system volume need have only:

SWAP.SYS
RT11SJ.SYS
DD.SYS (system device handler)
TT.SYS
LP.SYS (if your system includes a line printer)

The device handler is not included in the monitor file (as it was in Version 3B) so that you must include the system device handler on any system volume. If you use the foreground/background monitor RT11FB.SYS, you do not need TT.SYS because the terminal service is resident in that monitor.

However, if you intend to use the keyboard monitor commands, you need to include at least the utility programs PIP, DUP, and DIR, which are necessary for most of the keyboard commands to function. The programs EDIT, LINK, and HELP are also used frequently.

2.5 Mandatory Patches

To identify mandatory patches for installation, study the *RT–11 Software Dispatch Review*. Mandatory patches correct software errors discovered since the software was released. The *RT–11 Software Dispatch Review*, which is included in your software kit, summarizes all the patches to RT–11 Version 4 published to date in the *RT–11 Software Dispatch*. If you are a Category A license customer, you will receive issues of the dispatch as they are published for 90 days after installation (see the review for a description of its distribution policy). After the 90-day period, you must purchase Binary Program Update Service to receive copies of the *RT–11 Software Dispatch*.

Be sure to select the patches for only RT–11. You should install patches for each layered product when you install the layered product. A layered product is software (for example, BASIC–11) that is sold separately but that requires the RT–11 operating system environment in which to run.

Most published patches are binary patches that you install using the Save Image Patch Program (SIPP). In the event that monitors require patches, the patches may be published in both binary and source form. If a device handler requires a patch, only a source patch is published (in which case you have to reassemble and relink the handler). If you later perform the system generation process to create your own monitors and handlers, you must use source files with patches applied. Therefore, during the installation procedures, you must install both binary patches and source patches.

The *RT–11 Software Dispatch Review* designates patches as mandatory or optional and describes the criteria for these definitions.

Once you have identified the appropriate patches, mark them, copy them, or set them aside in some way so that they will be available when you perform the installation procedures.

2.6 Acquiring Sufficient Media

When you perform the installation procedures, you need blank media on which to make backup copies of the distribution, create a working system, and make backup copies of the working system. For example, if you receive RT–11 on RX01 diskettes, you need at least 11 blank diskettes, for the following purposes.

7	to back up distribution
2 (or more)	for working system
2 (an equal number)	to back up working system
?	for your application's data storage requirements

11+	total

Before you begin installation, read over the installation chapter that pertains to your configuration and establish how many cartridges, diskettes, disks, or magtapes you need. Then you can secure additional media, if required.

2.7 Formatting New Media

You may need to format certain types of media before you initialize them. RK05 disks are not formatted during manufacture. Diskettes are available in single-density but not double-density format. While the RX02 diskette drive can read and write single-density diskettes, you must reformat the diskettes for double-density in order to back up the distribution media and to use the device fully.

Once you have bootstrapped RT–11, you can use the FORMAT utility program to format your new volumes (as described in the installation chapters). See the *RT–11 System User's Guide* for a description of FORMAT.

However, if you are installing an RT–11 system that was distributed to you on magtape to run on RK05 disk, you need to format a disk before you boot RT–11. In this case, you can use the toggle-in formatting program as described in the following procedure; you should format an RK05 disk before you turn to the installation procedures.

If your configuration includes a pushbutton console emulator rather than a switch register, refer to your hardware user's guide for instructions on using the emulator. By using the emulator, deposit the RK05 disk formatting program listed in Table 2–3. The toggle-in procedure is as follows.

Mount the disk (write enabled) to be formatted in RK05 Unit 0. (This formatting program does not work on another unit.) Perform the following steps to deposit the program in memory, verify that you deposited it correctly, and run the program to format the disk.

1. Set the ENABLE/HALT switch to HALT.
2. Set the first address, 001000, in the switch register (see Table 2–3).
3. Press the LOAD ADDRESS switch.
4. Set the contents for the first address (from Table 2–3) in the switch register.
5. Lift the DEPOSIT switch. The computer automatically advances to the next address.
6. Set the contents for the next address (from Table 2–3) in the switch register.
7. Lift the DEPOSIT switch.
8. Repeat steps 6 and 7 until you have deposited all the instructions.

Now verify that you deposited the formatting program properly.

1. Set the first address, 001000, in the switch register.
2. Press the LOAD ADDRESS switch.
3. Press the EXAM switch to display the contents of that address in the data register.
4. Compare the number in the data register with the value for that address in Table 2–3.
5. If the values are the same, press EXAM again to display the contents of the next address. If the values are not the same, repeat the entire procedure for depositing the bootstrap. Verify the contents of all the addresses in this way. If any instruction is incorrect, repeat the entire deposit procedure.

Now run the format program.

1. Set the first address, 001000, in the switch register.
2. Press the LOAD ADDRESS switch.
3. Set the ENABLE/HALT switch to ENABLE.
4. Press the START switch.
5. Let the program execute for 60 seconds, then set the ENABLE/HALT switch to HALT to stop the program.

The disk is now formatted.

Table 2–3: RK05 Disk Formatting Program

Location	Contents
001000	012737
001002	006003
001004	177404
001006	105737
001010	177404
001012	100375
001014	000137
001016	001000

2.8 Customizations

Although the RT–11 components as distributed need no alteration for most applications, you can alter them in some ways. Many alterations require your going through the system generation process. Others require less time-consuming procedures, such as patching the distributed monitors. Compare Tables 1–4 and 1–5, and read Sections 2.8.1 through 2.8.42 to establish what you need to do. Identify any non-system-generation customizations you need to make and jot them down on the worksheet at the end of this chapter. You can perform the procedure for each selected customization during the installation process (Chapter 3, 4, 5, 6, or 7). System generation is necessary only if you need a customization that cannot be achieved with one of the procedures described in the following sections. Chapters 8 through 10 describe the system generation process.

NOTE

> Refer to Appendix I for additional patches that further customize specially generated monitors.

DIGITAL strongly recommends that when you install customization patches you use the SIPP utility. Also use the feature of SIPP that creates an indirect command file when it installs the patch. In this way, you can store copies of the indirect command files so that you can easily install the customization patches again, if necessary. This might be necessary, for instance, if you replace a customized component with a copy of that component from the patched backup volume after you install a published mandatory patch. When you invoke SIPP and it responds with an asterisk, enter the following:

```
.RUN SIPP(RET)
*filnam.COM=filnam.typ/L(RET)
```

The equal sign and file name cause SIPP to create the indirect file, and /L causes SIPP to not install the patch in the input file. You can use the indirect file to install the patch whenever necessary.

NOTE

> In the patches reproduced in this manual, lower-case alphabetic x represents unknown characters. These characters vary according to the specific software component.

2.8.1 RJS04 Support for RJS03

If your configuration includes an RJS03/4 disk drive, you can use all the space available on it only if you make this customization. RT–11 is distributed with RJS03/4 disk support initialized for RJS03. To allow complete use of space on RJS04, modify the device size entry in the DS handler as follows:

```
.RUN SIPP(RET)
*DS.SYS(RET)
Base?      0(RET)
Offset?   54(RET)

  Base          Offset        Old   New?
   000000        000054    002000   4000(RET)
   000000        000056    100016   (CTRL/Y)(RET)
*(CTRL/C)

.
```

2.8.2 Changebar Listings with SRCCOM

You can patch SRCCOM to change the default characters that SRCCOM uses to indicate insertions and deletions on listings. Normally, when you use the DIFFERENCES/CHANGEBAR command or SRCCOM's /D option to compare two files, SRCCOM places vertical bars next to each line that has been added to the new file and bullets (lower case alphabetic o) next to lines that have been deleted. If you want to use characters other than the vertical bar and bullet characters, you can patch SRCCOM.

In the following patch, n is the ASCII code for the character you want to use to indicate insertions and m is the ASCII code for the character you want to use to indicate deletions.

```
.RUN SIPP(RET)
*SRCCOM.SAV(RET)
Base?      0(RET)
Offset?   1000(RET)

  Base          Offset        Old   New?
   000000        001000    076157   ;A(RET)
   000000        001000       <o>   ;Am(RET)
   000000        001001       <|>   ;An(RET)
   000000        001002      <^C>   (CTRL/Y)(RET)
*(CTRL/C)

.
```

2.8.3 Control Status Register (CSR) Addresses Used by FORMAT

If devices that the FORMAT utility program supports (DM, RK, DX, DY, DP, and DL) are installed at nonstandard addresses, you must patch the FORMAT utility to change the CSR addresses it uses. Normally, FORMAT uses the standard CSR addresses to access devices on the system.

To patch FORMAT, establish which device has nonstandard CSR addresses. If more than one device has nonstandard addresses, apply the patch more than once. Locate the device in the following table, and use the values from the table in the patch.

Device	sss	bbbb
DM	1	7742
RK	2	7664
DY (first controller)	3	7664
DY (second controller)	3	7666
DP	4	7662
DL	5	7662
DX (first controller)	6	7664
DX (second controller)	6	7666

In the patch, nnnnnn is the new CSR address.

```
.RUN SIPP(RET)
*FORMAT.SAV(RET)
Segment? sss(RET)
Base?    bbbbbb(RET)
Offset?  0(RET)

Segment    Base    Offset  Old     New?
000sss    bbbbbb   000000 xxxxxx nnnnnnn(RET)
000sss    bbbbbb   000002 xxxxxx (CTRL/Y) (RET)
*(CTRL/C)
.
```

2.8.4 Terminal as Default Output Device Instead of Line Printer

If your configuration does not have a line printer, you can cause monitor commands to default to the terminal instead of the line printer. Since several monitor commands default the output device to LP, you should edit the startup command file to cause all system references to the device LP: to use the terminal. To change the defaults of such commands (for example, DUMP and PRINT) you need to add an ASSIGN TT: LP: command to the startup file. Then, every time you bootstrap the system, the reassignment of the default device takes place.

2.8.5 VT11/VS60 Floating Vectors

Under certain circumstances, you may need to change the VT11/VS60 vector address. VT11/VS60 display processor vectors are normally located at 320 to 332. However, the floating vector region on the PDP–11 is situated in locations 300 to 476. Therefore, you may have to move the VT11/VS60 vectors if you add other devices.

If the vectors for your VT11/VS60 change, install the following patch to modify the monitor for a different VT11/VS60 vector address. Once you have made this patch, all DIGITAL supplied software that accesses the display will function properly on the system without further patching.

In the patch, monitr.SYS is the name of the monitor file that you want to modify, $RMON is the value of that symbol from the monitor link map, and nnn is the location of the first VT11/VS60 vector on your system. Find the monitor link map for the monitor you want to alter (see Appendix H), and use the value of $RMON in the patch. Note that nnn must be an even value between 70 and 464.

```
.RUN SIPP(RET)
*monitr.sys(RET)
Base?     $RMON(RET)
Offset?   354(RET)

  Base        Offset       Old   New?
   $RMON       000354   000320   nnn (RET)
   $RMON       000356   000000   (CTRL/Y) (RET)
*(CTRL/C)
.
```

2.8.6 The Number of Directory Columns

You can patch DIR to change the number of columns in the directory that lists when you use the DIRECTORY command. Normally, the directory contains two columns of file names and types, file size in blocks, and date of creation. If you use the /FAST or /BRIEF options to DIRECTORY, DIR lists only file names and types in five columns. You can also use the /COLUMNS:n option to specify the number of columns in the directory. However, if you want to change the default number of columns in the directory, install the following patch.

In the patch, ffffff is an octal number (in the range 1 to 11) for the number of directory columns when you use the DIRECTORY/FAST command. The value nnnnnn is an octal number (in the range 1 to 11) for the number of columns when you use the DIRECTORY command.

```
.RUN SIPP(RET)
*DIR.SAV(RET)
Base?     0(RET)
Offset?   1000(RET)

  Base        Offset       Old  New?
   000000      001000   000005  ffffff(RET)
   000000      001002   000002  nnnnnn(RET)
   000000      001004   000005  (CTRL/Y) (RET)
*(CTRL/C)
.
```

2.8.7 The Number of /Q Program Sections LINK Allows

You can patch LINK to change the number of absolute base address p-sects (/Q p-sects) that LINK allows. Normally, the /Q option to LINK lets you specify the absolute base addresses of up to eight p-sects in your program. You need absolute base address p-sects to prepare programs in absolute loading format for use in read only memory (ROM) storage. Refer to the *RT–11 System User's Guide* for more information about absolute base address p-sects and about LINK in general.

The limit of eight such p-sects, however, is the default number, and you can change it by patching LINK. LINK uses the number of p-sects to set up the /Q buffer area and to establish how many times it should ask the question:

> Load section: address?

Note however, that LINK allocates the buffer space even if you do not use the /Q option when you perform the link. LINK calculates the size of the buffer to be three times the contents of QSWVAL.

To change the number of /Q p-sects LINK allows, use SIPP to patch LINK.SAV as follows. In the patch, nnn is the number of p-sects you want. The number nnn must be in the range 1 to 177 (octal).

```
.RUN SIPP<RET>
*LINK.SAV<RET>
Segment? 0<RET>
Base?    2172<RET>
Offset?  0<RET>

Segment    Base    Offset     Old    New?
000000    002172   000000   000010   nnn<RET>
000000    002172   000002   001742   <CTRL/Y> <RET>
*<CTRL/C>
.
```

2.8.8 The Size of LINK's Library Module List

If you want to permanently change the default size of LINK's list of library modules, you can patch LINK. LINK creates a list of 252 (octal) modules to be included from libraries during the link operation. Because the size of each entry in this list is larger in RT–11 Version 4 than in previous versions, the list may not be large enough for your application. You can use the LINK /P option at link time to increase the size of this list. If you want to permanently change the default size of the list (and avoid using the /P option), you can install a patch in the linker. You can still override the new default at link time by using /P.

Note that if you use this patch to increase the default size of the list, the maximum number of global symbols allowed in the link will be reduced.

NOTE

> If you are a DIBOL user, you must install this patch. Make the default number of modules LINK holds 400 (octal) or greater.

In the patch, nnnnnn is the number of modules the list should hold.

```
.RUN SIPP<RET>
*LINK.SAV<RET>
Segment? 1<RET>
Base?    4376<RET>
Offset?  114<RET>

Segment    Base     Offset    Old     New?
000001     004376   000114   000252   nnnnnn<RET>
000001     004376   000116   175616   <CTRL/Y> <RET>
*<CTRL/C>
.
```

2.8.9 The Size of the QUEUE Work File

QUEUE, the device queue foreground program (or system job) uses a work file five (octal) blocks long. This work file allows you to queue approximately 127 files at once. If your application requires larger queues, you can patch QUEUE.REL to change the default size of the work file.

In the following patch, nnn is the default size of the work file in octal blocks. To compute the approximate size of the work file that would be required for the number of files you need queued at once, use the formula:

nnn = (maximum number of file specifications in queue at one time + 1)/32 + 1

```
.RUN SIPP<RET>
*QUEUE.REL<RET>
Base?    1000<RET>
Offset?  0<RET>

  Base      Offset     Old   New
   001000    000000 000005  nnn<RET>
   001000    000002 000000  <CTRL/Z> <RET>

Offset?  1646<RET>

  Base      Offset     Old   New?
   001000    001646 000004  (nnn-1)<RET>
   001000    001650 000200  (nnn-1)*32<RET>
   001000    001652 000000  <CTRL/Y> <RET>
*<CTRL/C>
.
```

2.8.10 EDIT

To customize the editor, EDIT, you can reduce the size of the editor's text window and you can cause the editor to operate correctly on terminals with nonstandard ESCAPE codes.

2.8.10.1 Size of the Text Window — If your configuration includes a VT11/VS60 display processor, you may need to reduce the size of the editor's text window to correct an overflow problem. The editor works in such a way that when you use a VT11 or VS60, the window into the buffer and the scrolled command lines are separate "pictures." On rare occasions, if the text window around the cursor contains long lines and several line feed (or form feed) characters, the window may overflow onto the scrolled editing commands, making that portion of the screen difficult to read. While this problem does not usually occur, if it does, you can make the obscure lines clear by advancing the cursor several lines.

However, if the problem is troublesome for your particular application, you can remove it by reducing the size of the window displayed. Use the following patch to make this change. In the patch, nnn is the number of lines to be displayed above and below the cursor. To eliminate the problem, make nnn smaller than 12.

```
.RUN SIPP(RET)
*EDIT.SAV(RET)
Base?     0(RET)
Offset?   17704(RET)

  Base        Offset      Old   New?
   000000      017704   000012  nnn(RET)
   000000      017706   010200  (CTRL/Y)(RET)
*(CTRL/C)

.
```

2.8.10.2 Terminals with Nonstandard ESCAPE Code — You can patch the editor to allow it to operate correctly on terminals with nonstandard escape code. Certain older terminals generate 175 (octal) or 176 (octal), rather than the standard 33 (octal), when you type the ESCAPE or ALTMODE key. Because codes 175 (octal) and 176 (octal) represent legitimate characters on more modern terminals, EDIT does not recognize ESCAPE as the command terminator in the older terminals.

If you have an older terminal, you can correct the problem by making the following patch, so that you can use ESCAPE as documented in the EDIT chapter of the *RT–11 System User's Guide*. In the patch, nnn represents the octal code that your terminal generates when you type ESCAPE on it.

```
.RUN SIPP(RET)
*EDIT.SAV(RET)
Base?     0(RET)
Offset?   11053(RET)

  Base        Offset      Old   New?
   000000      011053     033   nnn(RET)
   000000      011054     016   (CTRL/Y)(RET)
*(CTRL/C)

.
```

The character $ echoes on the terminal, regardless of the octal value used for the escape code. However, EDIT recognizes only the escape code you patch into it, not both.

2.8.11 Initial Console Fill Characteristics

If you have certain types of terminals, you need to establish fill characters to follow each carriage return. Because standard monitors are set to generate no fill characters, you must patch the monitor if you have one of these terminals. For example, the serial LA30 (LA30S) DECwriter requires that filler characters follow each carriage return; the 600, 1200, and 2400 baud VT05 terminals require that filler characters follow each line feed. RT–11 has a mechanism that permits you to specify any number of fills to follow any character.

If your console terminal requires fill characters, you can install the following patch so that the fill characteristics are set automatically whenever you bootstrap the monitor. DIGITAL recommends that you install this patch if appropriate, since it ensures that the bootstrap message appears with the proper fill characters.

In the patch, monitr.SYS is the name of the monitor file that you want to modify, ..TFIL is the value of that symbol from the monitor link map, and nnn is the value from the table. Find the monitor link map for the monitor you want to alter (see Appendix H), and use the value of ..TFIL in the patch.

```
.RUN SIPP(RET)
*monitr.SYS(RET)
Base?     0(RET)
Offset?   ..TFIL(RET)

  Base        Offset        Old    New?
   000000     ..TFIL       000000  nnn(RET)
   000000     ..TFIL+2     000100  (CTRL/Y)(RET)
*(CTRL/C)

.
```

Terminal	Loc 56	Loc 57	Resulting octal word (nnn)
LA30S 110 baud	015(8)	002(8)	1015
LA30S 150 baud	015(8)	004(8)	2015
LA30S 300 baud	015(8)	012(8)	5015
VT05 600 baud	012(8)	001(8)	412
VT05 1200 baud	012(8)	002(8)	1012
VT05 2400 baud	012(8)	004(8)	2012

If the monitor you patch is the hardware bootable monitor, write a new system bootstrap with the COPY/BOOT command.

2.8.12 Extracting the Overlay Handlers from SYSLIB

You can extract the overlay handlers from the default system library, SYSLIB, if those are the only components of SYSLIB you need. Remember that you need the overlay handlers that are included in SYSLIB if you intend to use overlaid programs.

Create a separate library containing only the overlay handlers by using the EXTRACT option to the LIBRARY command, as described in the following procedure:

1. Extract the unmapped overlay handler (for LINK/O overlays) from SYSLIB:

```
.LIBRARY/EXTRACT SYSLIB OHANDL(RET)
Global? $OVRH(RET)
Global? (RET)
.
```

2. Extract the virtual overlay handler (for LINK/V overlays) from SYSLIB:

```
.LIBRARY/EXTRACT SYSLIB VHANDL(RET)
Global? $OVRHV(RET)
Global? (RET)
.
```

3. Combine the files you extract in a new library (which can have any name, including SYSLIB). You *must* combine the files in the order indicated. In the command, you need both the REMOVE and CREATE options. CREATE creates the new library, and REMOVE removes a duplicate global, $OVRH from the library directory. This global appears in both OHANDL and VHANDL because VHANDL includes the program code found in OHANDL. Remember, VHANDL processes both unmapped and virtual overlays.

```
.LIBRARY/REMOVE/CREATE newlib VHANDL,OHANDL(RET)
Global? $OVRH(RET)
Global? (RET)
.
```

You can also put either handler in a library by itself. OHANDL handles only unmapped overlays. VHANDL handles both unmapped and extended memory overlays, but it is larger than OHANDL.

In the following command, which creates a library from one of the overlay handlers, x is O for OHANDL or V for VHANDL.

```
.LIBRARY/CREATE newlib xHANDL(RET)
.
```

To add the overlay handlers to another library or module (for example, FORLIB or the DIBOL library), you can combine the distributed SYSLIB.OBJ with the library or module. Use the LIBR utility, but remove the global $OVRH from the new library. You must remove this global if you use as input to LIBR any library that includes both overlay handlers (as does the distribured SYSLIB). In the following command to combine libraries, the /REMOVE option removes $OVRH. Note that newfile is a single module or a library and gbl is any global that must be removed from newfile. If you combine FORLIB with SYSLIB, you must also remove the FORTRAN globals (refer to the FORTRAN library generation procedures).

```
.LIBRARY/REMOVE/INCLUDE SYSLIB newfile(RET)
Global?   $OVRH(RET)
Global?   gbl(RET)
Global?   (RET)
.
```

2.8.13 Installing Other Devices

You may need to install device handlers that are available but are not installed in the standard monitors. Installing a device handler adds information to the monitor device tables so that you can use the device. Many devices are available in the standard monitors (see the *RT–11 System User's Guide* for a list of available devices). You can perform the system generation process to create monitors and handlers that support nonstandard devices. The *RT–11 Software Support Manual* describes how to write your own device handler.

When you bootstrap RT–11, the bootstrap routine locates the system device handler and installs it. Then the bootstrap looks at the rest of the device handler files on the system device and tries to install the corresponding handler for each hardware device it finds in the configuration. It does not try to install any handlers for which there is no hardware. If there are more handlers than device slots, the bootstrap uses a certain priority scheme to establish which handlers to install. See the *RT–11 Software Support Manual* for a description of these priorities.

To ascertain which device handlers have been installed, use the keyboard monitor SHOW command, which shows you which devices are installed and whether any empty device slots are available. If the bootstrap did not install a device when you booted the system, it did not have enough device slots when it encountered the handler, the hardware was not present, or the device handler was not present. To install a device, ensure that the correct handler is on the system device and that the hardware is present. If there are no free slots, use the REMOVE command to remove an unneeded device, and the INSTALL command to install the device you need.

```
.SHOW RET
TT    (Resident)
RK    (Resident)
 RK0    =  SY
DX
DY
DD
DT
MT
CT
LP
.REMOVE LP:RET
.INSTALL LS:RET
```

The standard (distributed) SJ and FB monitors provide a total of 16 device slots. If your application requires more device slots, you can perform the system generation process to create your own monitor and device handlers. However, if you do not need to use more than 16 devices simultaneously, you need not generate your own monitor.

To control which handlers the bootstrap installs, simply place on the system device for your working system only the handlers for the devices you will be using. Do not include a handler in your working system if you will not be using that device.

You can keep a handler from being installed at boot time by giving it a name that does not correspond to the naming conventions for the monitor being booted. A device handler is named yy.SYS (where yy is the device name) for SJ and FB monitors and yyX.SYS for XM monitors. Use the RENAME command to rename a handler.

Certain other device support cannot be installed by the bootstrap. Thus, different procedures are required. You can change the control status register and vector addresses for the parallel line printer, serial line printer, and DECtape II. You can install hardware magtape support, set magtape parity and density, and "change" RP02 support to RP03.

2.8.13.1 The CSR and Vector Addresses for Line Printers and DECtape II — Three devices — parallel line printer, serial line printer, and DECtape II — allow you to change the control status register (CSR) and vector addresses with the SET command. You need to change the addresses if the controller is installed at nonstandard addresses. When using the SET command, enter one of the following commands, where aaaaaa is the CSR address and bbb is the vector address.

```
.SET LP: CSR=aaaaaa(RET)
.SET LP: VECTOR=bbb(RET)

.SET LS: CSR=aaaaaa(RET)
.SET LS: VECTOR=bbb(RET)

.SET DD: CSR=aaaaaa(RET)
.SET DD: VECTOR=bbb(RET)

.SET DD: CSR2=aaaaaa(RET)
.SET DD: VEC2=bbb(RET)
```

These commands permanently alter the handler .SYS file. If you want to change the addresses again, you can use the SET command again.

2.8.13.2 Hardware Magtape Support — To install hardware magtape support, you must switch from the file-structured handler to the hardware handler. The files MT.SYS, MM.SYS, and MS.SYS are file-structured TM11, TJU16, and TS11 magtape handlers. If you want to use a hardware magtape handler instead of a file-structured handler, you must rename the hardware handler (included in the software kit as MTHD.SYS, MMHD.SYS, and MSHD.SYS) to MT.SYS, MM.SYS, or MS.SYS. Note however, that all the features of the hardware handlers are included in the file-structured handlers. The hardware handlers offer fewer features but smaller size.

If you do need to use the hardware handler, be sure to rename the file-structured handler to something else if you want to save the file-structured handler. Then, rename the hardware handler MTHD.SYS to MT.SYS (for TM11), MMHD.SYS to MM.SYS (for TJU16), or MSHD.SYS to MS.SYS (for TS11). Use the INSTALL command to install the handler or reboot the system to let the bootstrap install the handler.

For example, to install TM11 hardware support in place of TM11 file-structured support, first make sure that MTHD.SYS is on the system volume and then do the following:

1. Rename the file-structured handler to save it.

```
.RENAME/SYSTEM MT.SYS MT.FIL(RET)
.REMOVE MT:(RET)
.
```

You must remove the old handler before you can install the new one.

2. Rename the hardware handler.

```
.RENAME/SYSTEM MTHD.SYS MT.SYS(RET)
.INSTALL MT:(RET)
.
```

You can switch between the file-structured handler and the hardware handler by first renaming the handlers, then either using the INSTALL and REMOVE commands or rebooting. The INSTALL and REMOVE commands are unnecessary if you reboot the system. The handler you have named MT.SYS will be installed in the device table when the system is booted.

Likewise, you can install the TJU16 hardware handler in place of the TJU16 file-structured handler by first making sure that MMHD.SYS is on the system volume and then renaming it MM.SYS. Or you can install the TS11 hardware handler in place of the TS11 file-structured handler by first making sure that MSHD.SYS is on the system volume and then renaming it MS.SYS.

2.8.13.3 Magtape Parity and Density — If you need to operate TM11 or TJU16 magtape at parity or density settings different from standard support, you can use the monitor SET command as described in the *RT-11 System User's Guide* to set them. The distributed monitors support seven- and nine-track TM11 and TJU16 magtape at 800 BPI and odd parity. You cannot set the density of TS11 magtape, since it is 1600 bpi odd parity only.

2.8.13.4 "Change" RP02 Support to RP03 — RT-11 can accommodate the 40000 (decimal) block RP02 cartridge as a single logical unit but not the RP03, because the RT-11 file structure can accommodate only a maximum of 65536 (decimal) blocks. Therefore, to use RP03, think of each RP03 drive as two logical units of 40000 blocks each (a single RP03 looks like two RP02 drives to the system). Access the cartridge on physical unit n as logical DPn+4; thus, refer to drive 0 as DP0: and DP4:, drive 1 as DP1: and DP5:, and so on. Each logical unit has its own directory and data space. The system can support up to four RP03s. You can mix RP02s and RP03s as long as the total number of units (physical drives) on the system does not exceed four. The system can support as many as eight RP02s.

2.8.14 BATCH

To minimize space demands on your system device, you can make patches that cause BATCH to access certain system programs on DK: rather than on SY:. These patches allow you to store certain system programs on DK: rather than on SY: and let BATCH access them there.

You can make any or all of the patches. Simply patch BATCH for the system programs you need to remove from the system device. Copy the programs for which you install patches to the device on which you want them to reside, then delete them from the system device (SY:). Finally, use the ASSIGN command to assign the logical name DK: to the device to which you copied the system programs. Then, run BATCH as usual.

The following patch to BATCH makes DK: the default storage volume for one of the specified programs. In the patch, use the value of nnnnnn (from the table) that corresponds to the program you want to affect.

Program	Value of nnnnnn
DIR	1361
MACRO	1766
FORTRAN	1777
LINK	2016
PIP	2031
BASIC	2050

```
.RUN SIPP<RET>
*BATCH.SAV<RET>
Base?     26612<RET>
Offset?   nnnnnn<RET>
  Base        Offset        Old    New
   026612     nnnnnn      xxxxxx   \<RET>

  Base        Offset        Old    New
   026612     nnnnnn        040    125<RET>
   026612     nnnnnn+2      040    <CTRL/Y> <RET>
*<CTRL/C>
.
```

Once you assign DK: to a device other than SY:, the new device becomes the default input and output storage device for most system programs. You may need to modify BATCH jobs to reference certain files on SY: explicitly, since that is no longer the same device as DK:. You can keep .BAT and .CTL files on SY: by invoking BATCH as follows:

```
.RUN BATCH<RET>
*SY:myjob=SY:myjob<RET>
```

2.8.15 The Default SYSLIB Device

You can patch the linker to make it look for the default system library (SYSLIB.OBJ) on the device you choose instead of on the system device (SY:). This change may be useful if you have space problems on your system device, because you can then place SYSLIB on the device you have specified to LINK.

To change the device on which SYSLIB.OBJ resides, make the following patch to LINK.SAV. In the patch, dev is the name of the device on which you want to place SYSLIB.

```
.RUN SIPP<RET>
*LINK.SAV<RET>
Segment? 1<RET>
Base?    10070<RET>
Offset?  10<RET>

Segment    Base     Offset        Old   New?
000001     10070    000010     075250   ;Rdev<RET>
000001     10070    000012     075273   <CTRL/Y> <RET>
*<CTRL/C>
.
```

2.8.16 The Help Text

To change the help text that prints when you use the HELP command, you must create your own help text macro library, process that library with LIBR, and copy the library and the file HELP.EXE to the same volume.

The distribution kit includes three help files: HELP.SAV, HELP.MLB, and HELP.EXE. HELP.EXE is the help program. To run, the help program requires a help text macro library file. DIGITAL supplies HELP.MLB, which is such a library. HELP.SAV consists of HELP.EXE and HELP.MLB (after the librarian, LIBR processes it) merged into a ready-to-use utility. HELP.SAV is the only file you need if you do not want to change the help text.

However, if you do want to change the text that prints when you use the HELP command, you must perform the following procedure.

First, edit the file HELP.MLB in your working system. Make sure that this file (as well as the rest of the distribution) is safely backed up and the actual distribution media are stored away. Add any explanations your application requires and delete any explanations that do not apply to your application.

When you edit HELP.MLB, you must follow a specific format, as follows:

1. Give each topic in the file an alphabetic name.
2. The name you give must be unique within the first six characters.
3. Place each topic on a page, delimited by form feeds (see following example).
4. Place topics in alphabetical order within the file.
5. Leave the dummy topic 999999 at the end of the file.

The following is an example of a topic, properly formatted, on a page.

```
<FF>
.MACRO<TAB>TOPICNAME<RET><LF>
TOPICNAME<TAB>ONE LINE DESCRIPTION OF THE TOPIC<RET><LF>
<RET><LF>
<SP><SP>SUBTOPICNAME<RET><LF>
<TAB>TEXT ABOUT THE SUBTOPIC<RET><LF>
<TAB>MORE TEXT (ANY NUMBER OF LINES) <RET><LF>
<SP><SP><SP>SUBTOPICITEMNAME<RET><LF>
<TAB>DESCRIPTION OF SUBTOPIC ITEM (USUALLY ONLY FOR<RET><LF>
<TAB>THE SUBTOPIC 'OPTION'), ANY NUMBER OF LINES<RET><LF>
[ANOTHER SUBTOPIC, STARTING FROM THE BLANK LINE ABOVE]
.ENDM<RET><LF>
<FF>
```

Now, create the help text library file by processing HELP.MLB with the librarian. Use the following command:

```
.LIBR/MACRO HELP.TXT HELP.MLB<RET>
.
```

You can leave HELP.TXT as a separate file or merge it with HELP.EXE.

To run HELP with a separate program and text library file or files, copy HELP.EXE to the system volume. When you copy HELP.EXE, name it HELP.SAV.

```
.COPY xxn:HELP.EXE SY:HELP.SAV<RET>
.
```

Make sure that the file HELP.TXT that you create is also on the system volume. You can back up HELP.EXE and HELP.MLB on another volume and delete them from the system volume. Then, to invoke HELP, type:

```
.HELP<RET>
```

If you leave HELP.TXT separate, you can alter it more easily in the future. However, if HELP.TXT is a separate file, the program runs more slowly. This problem is significant on systems that use a small device (for example, diskette or DECtape II) as the system device.

DIGITAL recommends that you merge the two files, since the program runs faster that way. Use the following command to merge the program and text library file or files.

```
.COPY HELP.(EXE+TXT) HELP.SAV<RET>
.
```

In this case too, you can back up HELP.EXE and HELP.MLB on another volume and delete them from the system volume. You can also delete the file HELP.TXT since you can recreate it from the file HELP.MLB.

2.8.17 Listing Page Length in MACRO and CREF

If you do not use line printer paper of a standard size (10.5 inches long) or if your configuration does not include a line printer, you may need to modify the listing page length in MACRO and CREF. RT–11 MACRO and CREF set the number of lines printed on each listing page at 60. This line count is generally satisfactory only for applications with line printers that use paper 10.5 inches long. Therefore, in either case, you may require a listing of a different length. In each patch, nnn is the desired listing page length (in lines).

The patch to modify MACRO is as follows:

```
.RUN SIPP<RET>
*MACRO.SAV<RET>
Segment?  0<RET>
Base?     0<RET>
Offset?   4156<RET>

Segment       Base       Offset       Old   New?
000000       000000       004156    000074   nnn<RET>
000000       000000       004160    000367   <CTRL/Y><RET>
*<CTRL/C>

.
```

The patch to modify CREF is as follows:

```
.RUN SIPP(RET)
*CREF.SAV(RET)
Base?     0(RET)
Offset?  4436(RET)

   Base          Offset       Old   New?
    000000        004436   000074   nnn(RET)
    000000        004440   016700   (CTRL/Y)(RET)
*(CTRL/C)

.
```

2.8.18 Preventing a Hard Reset

Normally, the monitor performs a hard reset when a fatal system error occurs. The reset stops I/O transfers, minimizing the possiblity that the error will corrupt media. In some cases, the cause of software errors might still be in memory, and the reset preserves the data, making it possible to analyze the error.

However, in rare cases, the reset may prevent diagnosis of hardware errors. If you prefer to suppress the reset, you can install the following patch in the monitor, although doing so increases the risk of corrupting media. DIGITAL does not recommend using a monitor with this patch installed except for diagnostic purposes. Do not use such a monitor for normal operations.

In the patch, monitr.SYS is the name of the monitor file you want to modify, and FATAL is the value of that symbol from the monitor link map. Find the monitor link map for the monitor you want to alter (see Appendix H), and use the value of FATAL in the patch.

```
.RUN SIPP(RET)
*monitr.sys(RET)
Base?     0(RET)
Offset?  FATAL(RET)

   Base          Offset       Old   New?
    000000       FATAL     000005   240(RET)
    000000       FATAL+2   010127   (CTRL/Y)(RET)
*(CTRL/C)

.
```

2.8.19 Running RT–11 in Less Memory Than Is Available

If your application requires that RT–11 run in less memory than is available, you can make a patch that allows you to bootstrap the system to run in, for example, the lower 12K words or 8K words of a 16K word machine. Both the SJ and the FB monitors have bootstraps that allow the system to run in less memory than is available. The distributed monitors automatically make use of all available memory, since most applications require that RT–11 do so. However, if your configuration includes a hardware switch register and your application does require less memory, you can make the following patch. In the patch, monitr.SYS is the name of the monitor file that you want to modify.

If this is the hardware bootable monitor, you must write a new system bootstrap with the COPY/BOOT command after you install this patch.

```
.RUN SIPP<RET>
*monitr.SYS<RET>
Base?     1000<RET>
Offset?   30<RET>

   Base          Offset          Old          New?
    001000        000030       000407        0<RET>
    001000        000032       013704        <CTRL/Y> <RET>
*<CTRL/C>
.
```

Once you make the patch to your monitor, a halt occurs whenever you boot that monitor file. While the system is halted, set the switch register to one of the values from the following table and press the CONTINUE switch; the bootstrap operation completes for the memory size you specify.

If your configuration does not include a hardware switch register or if you want the system to always boot in a specified amount of memory without halting, you can make the following patch. In the patch, monitr.SYS is the name of the monitor file that you want to modify and nnnnnn is a value from the table.

```
.RUN SIPP<RET>
*monitr.SYS<RET>
Base?     1000<RET>
Offset?   30<RET>

   Base          Offset          Old          New?
    001000        000030       000407        240<RET>
    001000        000032       013704        12704<RET>
    001000        000034       177570        nnnnnn<RET>
    001000        000036       042704        <CTRL/Y> <RET>
*<CTRL/C>
.
```

If this is the hardware bootable monitor, write a new system bootstrap with the COPY/BOOT command. Once you make this patch to your monitor, the system will boot in the memory you specify whenever you boot the patched monitor file.

Value	Size in Words
40000	8K
44000	9K
50000	10K
54000	11K
60000	12K
64000	13K
70000	14K
74000	15K
100000	16K
104000	17K

Value	Size in Words
110000	18K
114000	19K
120000	20K
124000	21K
130000	22K
134000	23K
140000	24K
144000	25K
150000	26K
154000	27K

To run RT–11 in exactly 28K words, use the customization patch in Section 2.8.24.

2.8.20 Setting Upper Limit on File Size

If your application requires an upper limit on the size of a file, you can install a patch that changes the maximum size RT–11 allocates in a general .ENTER request. On distributed monitors, the .ENTER programmed request allocates space in such a way that the maximum size of a file is either half the largest space available or the entire second largest space available, whichever is larger. For most applications, this scheme is satisfactory and should be left unchanged. However, if yours is an application that requires an upper limit, you should make the following patch.

In the patch, monitr.SYS is the name of the monitor file that you want to modify, $RMON is the value of that symbol from the monitor link map, and nnnnnn is the octal number of blocks that is to be the maximum file size for a general .ENTER. Find the monitor link map for the monitor you want to alter (see Appendix H), and use the value of $RMON in the patch.

```
.RUN SIPP(RET)
*monitr.SYS(RET)
Base?     $RMON(RET)
Offset?   314(RET)

      Base      Offset      Old      New?
      $RMON     000314    177777     nnnnnn(RET)
      $RMON     000316    xxxxxx     (CTRL/Y) (RET)
*(CTRL/C)
.
```

2.8.21 50-Cycle Instead of 60-Cycle Clock Rate

You can modify the monitor so that it causes the TIME command to base calculations on a 50-cycle clock rate rather than a 60-cycle rate, which is the standard version of RT–11.The 50-cycle clock has specialized uses and is the common frequency in Europe. To alter the rate, you must modify the monitor so that bit 5 is set in the monitor configuration word.

In the patch, monitr.SYS is the name of the monitor file that you want to modify and $RMON is the value of that symbol from the monitor link map. To make this change, you need to modify the monitor so that bit 5 is set in the monitor configuration word. Therefore, the new value that you enter must be the sum of the old value displayed by SIPP plus 40 (octal). Find the monitor link map for the monitor you want to alter (see Appendix H), and use the value of $RMON in the patch.

```
.RUN SIPP(RET)
*monitr.SYS(RET)
Base?     $RMON(RET)
Offset?   300(RET)

  Base        Offset      Old     New?
   $RMON      000300    nnnnnn    nnnnnn+40(RET)  (You must add 40)
   $RMON      000302    000000    (CTRL/Y)(RET)
*(CTRL/C)

.
```

2.8.22 Number of RF11 Platters

RT–11 no longer supports RF11 disks. However, the RF handler is included in the software kit, so that if your configuration includes an RF11 drive, you can still use it. RT–11 is distributed with RF11 fixed-head disk initialized for one platter, but you can make a customization that allows RT–11 to make use of more than one. To use more than one platter, modify the device size entry in the RF handler. In the patch, nnnnn is the value from the table.

```
.RUN SIPP(RET)
*RF.SYS(RET)
Base?     0(RET)
Offset?   54(RET)

  Base        Offset      Old     New?
   000000     000054    004000    nnnnnn(RET)
   000000     000056    100012    (CTRL/Y)(RET)
*(CTRL/C)

.
```

Number of Platters	New Value of Table
1	2000
2	4000
3	6000
4	10000

2.8.23 Loading Files with CAPS–11

If you use CAPS–11 to load RT–11 files, you must modify the cassette handler. CAPS–11 cassette file headers differ from RT–11 cassette file headers, so that problems occur when you transfer files to cassette. When you load files with the CAPS–11 CABLDR or with the CTLOAD bootstrap, CABLDR and CTLOAD interpret the level byte in the file header as a header continuation byte. If the byte in the cassette header is non-zero, CAPS–11 ignores at least the first data record of the file, since it assumes that the record is an auxiliary header record. To avoid losing any data records, you must patch CT.SYS so that all header records contain a level byte of 0.

Patch CT.SYS as follows:

```
.RUN SIPP(RET)
*CT.SYS(RET)
Base?     1000(RET)
Offset?   3463(RET)

   Base        Offset       Old   New?
    001000      003463      001   0(RET)
    001000      003464      040   (CTRL/Y)(RET)
*(CTRL/C)

.
```

If you need to use the system generation process to build your own system, you can edit CT.MAC. Use any editor to create a file CT.SLP as follows. Use SLP to edit CT.MAC.

```
-/LEVEL:/,.
LEVEL:(TAB).BYTE(TAB)0
/
```

2.8.24 Setting Upper Limit on Memory Size

If your PDP–11 does not generate a bus timeout trap, when the running program accesses location 160000, the RT–11 bootstrap may assume that you have an LSI–11 with the MSV11–DD memory option. The bootsrtap assumes that there are 30K words available for the operating system. If this is not the case, RT–11 will not load into memory properly. If you install the following patch in your monitor, the bootstrap will never look for more than 28K words of memory. You cannot install this patch in an XM monitor. In the patch, monitr.SYS is the name of the monitor file that you want to modify.

```
.RUN SIPP(RET)
*monitr.SYS(RET)
Base?     1000(RET)
Offset?   60(RET)

   Base        Offset       Old     New?
    001000      000060   170000   160000(RET)
    001000      000062   001402   (CTRL/Y)(RET)
* (CTRL/C)

.
```

If the monitor you patch is the hardware bootable monitor, write a new system bootstrap with the COPY/BOOT command.

2.8.25 Suppressing the Bootstrap Message

If you want to prevent the monitor identification message from printing when you bootstrap a monitor, you can patch that monitor. In the patch, monitr.SYS is the name of the monitor file that you want to modify and ..SLNT is the value of that symbol from the monitor link map. Find the monitor link map for the monitor you want to alter (see Appendix H), and use the value of ..SLNT in the patch.

```
.RUN SIPP(RET)
*monitr.SYS(RET)
Base?     0(RET)
Offset?   ..SLNT(RET)

  Base        Offset        Old     New?
   000000     ..SLNT      000000    1(RET)
   000000     ..SLNT+2    001003    (CTRL/Y)(RET)
*(CTRL/C)

.
```

If the monitor you patch is the hardware bootable monitor, write a new system bootstrap with the COPY/BOOT command.

2.8.26 Suppressing the Startup Indirect Command File

If you want to prevent the startup indirect command file from executing when you bootstrap a monitor, you can patch that monitor. The standard monitors include startup indirect command file support although you need not select it if you perform the system generation process to create special monitors. In the patch, monitr.SYS is the name of the monitor file that you want to modify and ..NIND is the value of that symbol from the monitor link map. Find the monitor link map for the monitor you want to alter (see Appendix H), and use the value of ..NIND in the patch.

```
.RUN SIPP(RET)
*monitr.SYS(RET)
Base?     0(RET)
Offset?   ..NIND(RET)

  Base         Offset       Old     New?
   000000      ..NIND     004000    0(RET)
   000000      ..NIND+2   000044    (CTRL/Y)(RET)
*(CTRL/C)

.
```

If the monitor you patch is the hardware bootable monitor, write a new system bootstrap with the COPY/BOOT command.

2.8.27 Startup Indirect Command File Echo

If you want the startup indirect command file to execute when you bootstrap a monitor but you do not want the command lines in the file to echo on the terminal (appear on the terminal), you can patch the monitor. This patch causes the monitor to use the SET TT QUIET mode of operation. In the patch, monitr.SYS is the name of the monitor file that you want to modify and ..TTQU is the value of that symbol from the monitor link map. Find the monitor link map for the monitor you want to alter (see Appendix H), and use the value of ..TTQU in the patch.

```
.RUN SIPP<RET>
*monitr.SYS<RET>
Base?     0<RET>
Offset?   ..TTQU<RET>

  Base        Offset       Old   New?
   000000     ..TTQU     000000  1<RET>
   000000     ..TTQU+2   001403  <CTRL/Y><RET>
*<CTRL/C>

.
```

If the monitor you patch is the hardware bootable monitor, write a new system bootstrap with the COPY/BOOT command.

2.8.28 Changing Bootstrap Message

If you want to change the monitor identification message that appears when you bootstrap a monitor, you can patch that monitor. Run SIPP to patch the monitor file. Place the string (the message) in the monitor image starting at location 4740. End the string with a null byte. The string must not extend beyond location 4760. If the monitor you patch is the hardware bootable monitor, write a new system bootstrap with the COPY/BOOT command.

2.8.29 Default Device for Indirect Command Files

If you want to change the default device for indirect command files, you can patch the monitor. Normally, when you invoke an indirect command file (by typing @filnam), the default device where the monitor looks for the command file is DK:. If you have a special application, you can change this default to any three-character device name.

In the patch, monitr.SYS is the name of the monitor file that you want to modify, ..ATDK is the value of that symbol from the monitor link map, and nnn is the new default device name. Find the monitor link map for the monitor you want to alter (see Appendix H), and use the value of ..ATDK in the patch.

```
.RUN SIPP<RET>
*monitr.SYS<RET>
Base?     0<RET>
Offset?   ..ATDK<RET>

  Base        Offset       Old   New?
   000000     ..ATDK     015270  ;R<RET>
   000000     ..ATDK      <DK >  ;Rnnn<RET>
   000000     ..ATDK+2    <AN1>  <CTRL/Y><RET>
*<CTRL/C>

.
```

2.8.30 Default File Type for Indirect Command Files

If you want to change the default file type for indirect command files, you can patch the monitor. Normally, indirect command files have the default file type .COM. When you invoke an indirect command file (by typing @filnam), the monitor looks for the file filnam.COM. If you have a special application, you can change this default to any three-character file type.

In the patch, monitr.SYS is the name of the monitor file that you want to modify, ..ATFX is the value of that symbol from the monitor link map, and nnn is the new default file type. Find the monitor link map for the monitor you want to alter (see Appendix H), and use the value of ..ATFX in the patch.

```
.RUN SIPP<RET>
*monitr.SYS<RET>
Base?     0<RET>
Offset?   ..ATFX<RET>

  Base        Offset        Old   New?
   000000      ..ATFX     012445  ;R<RET>
   000000      ..ATFX      <COM>  ;Rnnn<RET>
   000000      ..ATFX+2    <xxx>  <CTRL/Y><RET>
*<CTRL/C>

.
```

2.8.31 Default Device for the FRUN Command

If you want to change the default device for the FRUN command, you can patch the monitor. Normally, when you start a foreground program under the FB or XM monitor (by typing FRUN filnam), the default device where the monitor looks for the program file is DK:. If you have a special application, you can change this default to any three-character device name.

In the patch, monitr.SYS is the name of the monitor file that you want to modify, ..FRDK is the value of that symbol from the monitor link map, and nnn is the new default device name. Find the link map for the monitor you want to alter (see Appendix H), and use the value of ..FRDK in the patch.

```
.RUN SIPP<RET>
*monitr.SYS<RET>
Base?     0<RET>
Offset?   ..FRDK<RET>

  Base        Offset        Old   New?
   000000      ..FRDK     015270  ;R<RET>
   000000      ..FRDK      <DK >  ;Rnnn<RET>
   000000      ..FRDK+2    <xxx>  <CTRL/Y><RET>
*<CTRL/C>

.
```

2.8.32 Default File Type for the FRUN Command

When you start a foreground program under the FB or XM monitor (by typing FRUN filnam), the default file type for the program file is .REL. If you have a special application, you can change this default to any three-character file type.

In the patch, monitr.SYS is the name of the monitor file that you want to modify, ..FRUX is the value of that symbol from the monitor link map, and nnn is the new default file type. Find the monitor link map for the monitor you want to alter (see Appendix H), and use the value of ..FRUX in the patch.

```
.RUN SIPP(RET)
*monitr.SYS(RET)
Base?     0(RET)
Offset?   ..FRUX(RET)

  Base        Offset       Old     New?
   000000     ..FRUX      070524   ;R(RET)
   000000     ..FRUX      <REL>    ;Rnnn(RET)
   000000     ..FRUX+2    <ANG>    (CTRL/Y)(RET)
*(CTRL/C)

.
```

2.8.33 Default Device for the EDIT Command

If you want to change the default device for the EDIT command, you can patch the monitor. Normally, when you invoke an editor by typing the EDIT command, the default device where the monitor looks for EDIT.SAV is DK:. If you have a special application, you can change this default to any three-character device name.

In the patch, monitr.SYS is the name of the monitor file that you want to modify, ..EDDV is the value of that symbol from the monitor link map, and nnn is the new default device name. Find the monitor link map for the monitor you want to alter (see Appendix H), and use the value of ..EDDV in the patch.

```
.RUN SIPP(RET)
*monitr.SYS(RET)
Base?     0(RET)
Offset?   ..EDDV(RET)

  Base        Offset       Old     New?
   000000     ..EDDV      075250   ;R(RET)
   000000     ..EDDV      <SY >    ;Rnnn(RET)
   000000     ..EDDV+2    <CHU>    (CTRL/Y)(RET)
*(CTRL/C)

.
```

2.8.34 Default File Name for the EDIT/EDIT Command

If you want the monitor to run a program other than EDIT.SAV when you type the EDIT command, you can patch the monitor to change the default file name for the EDIT/EDIT command. If you have a special application, you can change this file name to any six-character file name.

In the patch, monitr.SYS is the name of the monitor file that you want to modify, ..EDIF is the value of that symbol from the monitor link map, and mmmnnn is the new editor file name. Find the monitor link map for the monitor you want to alter (see Appendix H), and use the value of ..EDIF in the patch.

```
.RUN SIPP(RET)
*monitr.SYS(RET)
Base?     0(RET)
Offset?   ..EDIF(RET)

  Base        Offset         Old   New?
   000000     ..EDIF      017751   ;R(RET)
   000000     ..EDIF       <EDI>   ;Rmmm(RET)
   000000     ..EDIF+2    <T   >   ;Rnnn(RET)
   000000     ..EDIF+4    <   Y>   (CTRL/Y)(RET)
*(CTRL/C)

.
```

2.8.35 Default File Name for the EDIT/TECO Command

If you want the monitor to run a program other than TECO.SAV when you type the EDIT/TECO command, you can patch the monitor to change the default file name for the EDIT/TECO command. If you have a special application, you can change this file name to any six-character file name.

In the patch, monitr.SYS is the name of the monitor file that you want to modify, ..TECF is the value of that symbol from the monitor link map, and mmmnnn is the new editor file name. Find the monitor link map for the monitor you want to alter (see Appendix H), and use the value of ..TECF in the patch.

```
.RUN SIPP(RET)
*monitr.SYS(RET)
Base?     0(RET)
Offset?   ..TECF(RET)

  Base        Offset         Old   New?
   000000     ..TECF      076713   ;R(RET)
   000000     ..TECF       <TEC>   ;Rmmm(RET)
   000000     ..TECF+2    <O   >   ;Rnnn(RET)
   000000     ..TECF+4    <    >   (CTRL/Y)(RET)
*(CTRL/C)

.
```

2.8.36 Examine and Deposit Above the Background Job

If you want to be able to examine and modify the monitor and the I/O page, you can patch the monitor to remove a restriction on the use of the E (Examine) and the D (Deposit) keyboard commands. Normally, the monitor allows you to examine and modify only locations inside the background job's area. You can remove this restriction, but you must be extremely careful when modifying the monitor or I/O page, since you may inadvertently destroy the resident monitor or corrupt a device.

In the patch, monitr.SYS is the name of the monitor file that you want to modify and ..EMON is the value of that symbol from the monitor link map. Find the link map for the monitor you want to alter (see Appendix H), and use the value of ..EMON in the patch.

```
.RUN SIPP(RET)
*monitr.SYS(RET)
Base?      0(RET)
Offset?    ..EMON(RET)

  Base          Offset       Old    New?
   000000       ..EMON     103041   240(RET)
   000000       ..EMON+2   103007   (CTRL/Y)(RET)
*(CTRL/C)

.
```

2.8.37 Default Device for QUEMAN

You can change the default device for the queue manager (QUEMAN). Normally, when you use the PRINT or DELETE/ENTRY command (in an FB or XM system with QUEUE running as a foreground or system job) QUEMAN sends the file to the device LP: or deletes the entry from the LP: queue. If you frequently queue to another device (such as the serial line printer, LS:), you can change the default device for these commands.

In the patch, monitr.SYS is the name of the monitor file that you want to modify, ..QULP is the value of that symbol from the monitor link map, and dv is the two-character device name that you want as the default. Find the monitor link map for the monitor you want to alter (see Appendix H), and use the value of ..QULP in the patch.

```
.RUN SIPP(RET)
*monitor.SYS(RET)
Base?      0(RET)
Offset?    ..QULP(RET)

  Base          Offset       Old    New?
   000000       ..QULP     050114   ;A(RET)
   000000       ..QULP        <L>   ;Ad(RET)
   000000       ..QULP+1      <P>   ;Av(RET)
   000000       ..QULP+2      <:>   (CTRL/Y)(RET)
*(CTRL/C)

.
```

2.8.38 Indirect File Nesting Depth

You can increase the indirect file nesting depth. Normally, RT–11 allows you to nest indirect files to a depth of three. A nesting depth of three allows your indirect file to invoke another indirect file, which invokes another indirect file. If you have a special application that requires more nesting, you can change the maximum nesting depth by patching the monitor. Note that if you increase the nesting depth, any use of the indirect file feature will use more memory than is usual.

In the patch, monitr.SYS is the name of the monitor file that you want to modify, $RMON is the value of that symbol from the monitor link map, and nnn is the maximum indirect file nesting depth that you want; nnn should be a small integer, and must not be zero. Find the monitor link map for the monitor you want to alter (see Appendix H), and use the value of $RMON in the patch.

```
.RUN SIPP<RET>
*monitr.SYS<RET>
Base?      $RMON<RET>
Offset?    377<RET>

  Base         Offset        Old   New?
   $RMON        000377       003   nnn<RET>
   $RMON        000400       xxx   <CTRL/Y><RET>
*<CTRL/C>

.
```

2.8.39 Resuming Output-Stalled Jobs

You can improve FB or XM system throughput by changing the threshold for resuming output-stalled jobs. In an RT–11 FB or XM system, a job is placed in a stalled state whenever it has terminal output to print but there is no room in its terminal output ring buffer. The system restarts the job when room becomes available in that ring buffer. The system's default mode of operation is to restart the job as soon as a single character of space is available. If more than one job is running, this mode of operation can cause the system to spend much time swapping in the context of a job, simply to have it output a single character and then stall again. You can patch the monitor so that a job that is stalled waiting for room in the terminal output buffer does not resume execution until several characters are available in the ring buffer. If you have a foreground or system job that produces a large amount of terminal output, installing this patch can greatly improve system thoroughput.

In the patch, monitr.SYS is the name of the monitor file that you want to modify, ..TTON is the value of that symbol from the monitor link map, and nnn is the threshold value for resuming a terminal output stalled job. Find the monitor link map for the monitor you want to modify (see Appendix H), and use the value of ..TTON in the patch. The monitor will resume such a job when there are nnn-1 characters left to print in the output ring. The default value, 50 (octal), is the size of the ring; consequently, the monitor resumes a job when 47 characters are left to print (that is, when only one character position is available in the output ring). You can specify any value from 1 to the size of the output ring buffer (which is normally octal 50, but can be changed at system generation time). Note that a value of 1 will cause the job to stay stalled until its ring buffer is empty, and may cause terminal output to appear jerky.

```
.RUN SIPP<RET>
*monitr.SYS<RET>
Base?       0<RET>
Offset?     ..TTON<RET>

  Base         Offset          Old   New?
   000000       ..TTON      000050   nnn<RET>
   000000       ..TTON+2    xxxxxx   <CTRL/Y><RET>
*<CTRL/C>

.
```

2.8.40 Default Directory Segments

You can change the number of directory segments DUP creates when you initialize a volume. Normally, DUP uses the default number of directory segments (see the *RT–11 System User's Guide*), which depends on the size of the volume. In other words, when you use the INITIALIZE command to initialize a volume, DUP ascertains the size of the volume and checks a table (within DUP) to establish the number of directory segments to use. This table consists of two-word entries that give a size and the number of segments. DUP searches the table until it finds a size larger than or equal to the size of the volume being initialized and uses the value in the following word as the number of directory segments.

If the default number of directory segments for a volume is unacceptable for your application, you can use the /SEGMENTS:n option with INITIALIZE to initialize a volume with n directory segments. Refer to the *RT–11 System User's Guide.*

However, if you want DUP to always use a specific number, you can patch DUP to change DUP's directory segment table (duplicated in the following). Use SIPP to change whichever value or values you require.

```
.WORD     1000       ;Volumes with <= 512. blocks
.WORD     1          ;get 1 segment directories
.WORD     4000       ;Volumes with <= 2048. blocks
.WORD     4          ;get 4 segment directories
.WORD     30000      ;Volumes with <= 12288. blocks
.WORD     20         ;get 16. segments
.WORD     177777     ;Volumes with <= 65535. blocks
.WORD     37         ;get 31. segments
.BLKW     10.        ;Expansion space for finer variations
.WORD     0          ;Must be 0.
```

2.8.41 MACRO–11 .LIST/.NLIST Defaults

You can change the defaults for MACRO–11 listing control directives from .LIST (list) to .NLIST (no list) by modifying a word in MACRO.SAV. The word contains a set of flags that, if set, indicate that an item is not to be listed (.NLIST). Refer to the *PDP–11 MACRO–11 Language Reference Manual* for a description of the listing control directives. In the patch, nnnnnn is an octal value from the table. Choose the arguments you want as defaults for the .NLIST directive, add the values together, and enter the sum in nnnnnn.

```
.RUN SIPP(RET)
*MACRO.SAV(RET)
Segment? 0(RET)
Base?    0(RET)
Offset?  4162(RET)

Segment     Base      Offset     Old   New?
000000     000000     004162  103040   nnnnnn(RET)
000000     000000     004164  046456   (CTRL/Y)(RET)
*(CTRL/C)

.
```

Argument	Value
BEX	2
BIN	4
CND	10
COM	20
LD[1]	40
LOC	100
MC	200
MD	400
ME[1]	1000
MEB[1]	2000
SEQ	4000
SRC	10000
SYM	20000
TOC	40000
TTM[1]	100000

2.8.42 MACRO–11 .ENABL/.DSABL Defaults

You can change the defaults for MACRO–11 function control directives from .ENABL (enable) to .DSABL (disable) by modifying a word in MACRO.SAV. The word contains a set of flag bits that, if set, indicate that an argument is disabled (.DSABL). Refer to the *PDP–11 MACRO–11 Language Reference Manual* for a description of the function control directives. In the patch, nnnnnn is an octal value from the table. Choose the arguments you want as defaults for the .DSABL directive, add the values together, and enter the sum in nnnnnn.

```
.RUN SIPP(RET)
*MACRO.SAV(RET)
Segment? 0(RET)
Base?    0(RET)
Offset?  4160(RET)

Segment     Base      Offset     Old    New?
000000      000000     004160  000367  nnn(RET)
000000      000000     004162  103040  (CTRL/Y)(RET)
*(CTRL/C)

.
```

Argument	Value
ABS[1]	1
AMA[1]	2
CDR[1]	4
CRF	10
FPT[1]	20
GBL[1]	40
LC[1]	100
LSB[1]	200
PNC	400
REG	1000

[1] Initial default.

Figure 2–2: Installation Worksheet

SELECTED COMPONENTS **SIZE IN BLOCKS**

Monitors

□ RT11BL.SYS ____________
□ RT11SJ.SYS ____________
□ RT11FB.SYS ____________

System Device Handler

____

Other Device Handlers

____
____
____
____
____

□ SWAP.SYS ____________

Utility Programs

□ PIP.SAV ____________
□ DUP.SAV ____________
□ DIR.SAV ____________
□ FORMAT.SAV ____________
□ RESORC.SAV ____________
□ EDIT.SAV ____________
□ CREF.SAV ____________
□ LINK.SAV ____________
□ LIBR.SAV ____________
□ FILEX.SAV ____________
□ SRCCOM.SAV ____________
□ BINCOM.SAV ____________
□ SLP.SAV ____________
□ SIPP.SAV ____________
□ DUMP.SAV ____________
□ PAT.SAV ____________
□ HELP.SAV ____________
□ QUEUE.REL ____________
□ QUEMAN.SAV ____________

Help Files

□ HELP.MLB ____________
□ HELP.EXE ____________

____
____

Libraries

□ SYSLIB.OBJ ____________
□ SYSMAC.SML ____________

____
____

Assemblers

□ MACRO ____________
□ MAC8K ____________

____

Source Files

____
____
____
____

Other Files

____
____
____

ARRANGEMENT

System Volume

____
____
____
____
____
____
____
____
____
____
____

Volume 2

____
____
____
____
____

Volume 3

____
____
____
____
____

Volume 4

____
____
____
____
____

Volume 5

____
____
____
____
____

Customizations

____________ PAGE ____________
____________ PAGE ____________
____________ PAGE ____________
____________ PAGE ____________
____________ PAGE ____________

Chapter 3

Installing a System Distributed on a Small Device to Run on a Small Device

If RT–11 was distributed to you on single-density diskettes, DECtape II cartridges, or PDT–11 volumes and you intend to build a system to run on single-density diskette, DECtape II, or PDT–11, you should perform the procedures described in this chapter.

These procedures cover minor variations that depend on the specific device you have (diskette or cartridge). The word "volume" in this chapter refers to either diskette or cartridge (PDP–11 or PDT–11), whichever is appropriate.

Notice that each software kit contains two number one (bootable) volumes. The rest of the volumes in the kits can be used on either PDP or PDT hardware. PDT–11 volumes come in two types: PDT–11/130 volumes (which look just like DECtape II cartridges) and PDT–11/150 volumes (which look just like single-density diskettes). In fact, the PDT–11/130 and DECtape II software kits are identical. The PDT–11/150 and single-density diskette software kits are identical. However, the cartridge labeled 1/7 is bootable on PDP–11 hardware, while the cartridge labeled 1-T/7 is bootable on PDT–11/130 Intelligent Terminals. The diskette labeled 1/7 is bootable on PDP–11 hardware, while the diskette labeled 1-T/7 is bootable on PDT–11/150 Intelligent Terminals.

NOTE

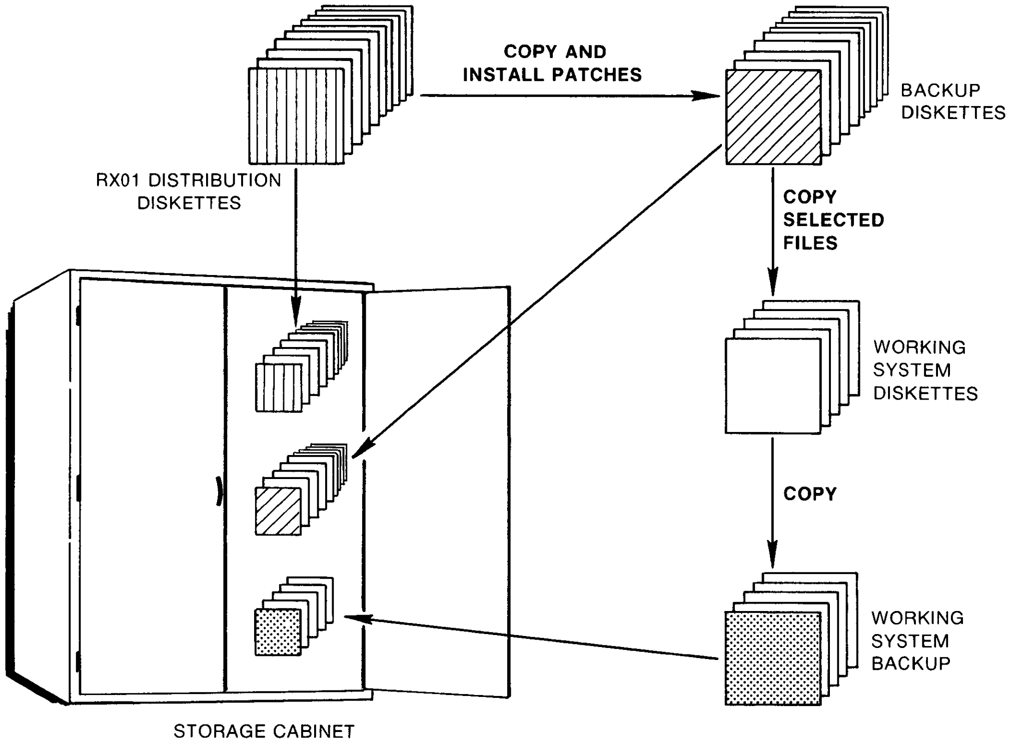
> If your hardware configuration includes a VT100 terminal, be sure to set auto XOFF/XON in SETUP mode before attempting to bootstrap RT–11. Never set TT NOPAGE when you use this terminal. Refer to your hardware manuals for more information about these settings.

To install your system, you will have to perform the steps summarized in the following list. Sections 3.1 through 3.10 describe the procedures for each step. Figure 3–1 shows the various backup volumes you create when you install RT–11.

1. Bootstrap the distribution volume.
2. Preserve the distribution volumes.
3. Install mandatory patches.
4. Create the system from chosen components.
5. Install the bootstrap on volumes that need to be bootable.
6. Customize the system.
7. Compress free space on each volume.
8. Preserve the working system.
9. Test the working system.
10. If appropriate, perform the system generation process.

The following sections correspond to each of these steps and describe in detail the procedures you must perform to complete each step.

Figure 3–1: Sample Backup Volumes



3.1 Bootstrapping the Distribution Volume

The first procedure you perform when installing RT–11 is bootstrapping the distribution volume.

Begin by making sure that the processor is powered up but not running. Insert distribution volume number 1 or 1-T (write protected, if possible) in RX01, DECtape II, or PDT–11 Unit 0. If your device is RX01 diskette, it has the device name DX:. If your device is DECtape II cartridge, it has the device name DD:. If your device is either PDT–11 device, it has the device name PD:.

Use the hardware bootstrap to boot the volume. (If your configuration does not include a hardware bootstrap, see Appendix E for toggle-in software bootstraps.)

NOTE

> If your hardware is a PDT–11 Intelligent Terminal, boot volume 1-T and store volume 1. If your configuration includes a PDP–11 processor, boot volume 1 and store volume 1-T.

RT–11 should respond with the following message if you have successfully bootstrapped the volume:

```
RT-11SJ V04.00
(Followed by any startup file commands.)
.
```

Use the DATE command to set the date (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
.DATE dd-mmm-yy(RET)
.
```

3.2 Preserving the Distribution Volumes

The first operation you perform with the running RT–11 system is to copy all the distribution volumes for backup as a safety measure in case of machine failure or human error.

NOTE

> If your volumes are diskettes, you can ignore instructions to write enable or write protect a volume. Diskettes are always write enabled. DECtape II provides a write protect feature, but single-density diskette does not.

Insert a blank volume (write enabled) in Unit 1. Then use the INITIALIZE command to initialize the blank volume (procedure follows). Use the BAD-

BLOCKS option with INITIALIZE to cover any bad blocks that may be on your volume. If the volume contains bad blocks, the **?DUP-I-Bad block detected nnnnnn** message appears on the terminal.

NOTE

> DIGITAL recommends that you use only volumes that do not have bad blocks when you back up distribution volumes and build a working system. To ascertain whether an already initialized volume has bad blocks, use the command DIRECTORY/BAD xxn:. You can use volumes with bad blocks later, for working or data volumes. However, there is an exception to this rule. If your volumes are cartridges, you can use the procedure in Section 2.4.3 when you build your working system but not when you back up the distribution volumes. The procedure in Section 2.4.3 creates *dummy* bad blocks on cartridges to improve response time.

```
.INITIALIZE/BADBLOCKS xx1:(RET)
xx1:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected xx1:
```

There may be a significant delay (as much as eight minutes) while the system scans the volume for bad blocks and creates a new directory. The monitor dot appears when this process is complete.

```
.
```

Now remove the newly initialized volume and initialize six additional blank volumes, leaving a blank volume that is write enabled and initialized inserted in Unit 1.

The next step in the preservation process is to copy all the files from distribution volume 1 or 1-T to the initialized blank volume, which becomes backup volume 1 or 1-T.

As long as the volume you intend to copy is bootable and contains certain system utility programs, you can boot RT–11 from that volume and copy the files from that volume to another volume in Unit 1. A bootable volume has the appropriate monitor file and system device handler and a bootstrap. The SQUEEZE/OUTPUT:xxn: command transfers all the files from one volume in Unit 0 to another one in Unit 1. At the same time, the command consolidates all the empty space at the end of the volume. Distribution volume 1 or 1-T is bootable, but the other volumes in your kit are not bootable because they lack the necessary components. Thus, these volumes require a different copy procedure. For volume 1 or 1-T, however, the procedure is straightforward. Type the following command, where xx is the physical device name.

```
.SQUEEZE/OUTPUT:xx1: xx0:(RET)
.
```

NOTE

> Both the SQUEEZE command with the /OUTPUT:xxn: option and the COPY command transfer files from one device to another. SQUEEZE/OUTPUT:xxn: consolidates free space on the device at the same time. The procedures in this manual use the command that is most appropriate and most efficient for an individual operation. For a better understanding of all RT–11 keyboard monitor commands, refer to the *RT–11 System User's Guide.*

There is a delay while the file transfer operation takes place. Then copy the bootstrap on the volume.

```
.COPY/BOOT xx1:RT11SJ.SYS xx1:(RET)
.
```

Remove the newly created backup volume from Unit 1, write protect it, and label it "backup RT–11 V04 1/7" or "backup RT–11 V04 1-T/7." (Use a soft-tipped pen when you label diskettes.)

To copy the remaining distribution volumes, which lack certain system components, run RT–11 from distribution volume 1 or 1-T in Unit 0.

Then, type:

```
.SQUEEZE/WAIT/OUTPUT:xx1: xx0:(RET)
Mount output volume in xx1:; Continue?
```

Insert an initialized blank volume (write enabled) in Unit 1.

```
Y(RET)

Mount input volume in xx0:; Continue?
```

Replace the volume in Unit 0 with distribution volume 2 (write protected, if possible).

```
Y(RET)

Mount system volume in xx0:; Continue?
```

Replace the volume in Unit 0 with distribution volume 1 or 1-T (write protected, if possible).

```
Y(RET)
.
```

Remove the volume from Unit 1, label it "backup RT–11 V04 2/7," write protect it, if possible.

Again type the command:

```
.SQUEEZE/WAIT/OUTPUT:xx1: xx0:(RET)
Mount output volume in xx1:; Continue?
```

Insert another initialized blank volume in Unit 1.

```
Y(RET)

Mount input volume in xx0:; Continue?
```

Replace the volume in Unit 0 with distribution volume 3 (write protected, if possible).

```
Y(RET)

Mount system volume in xx0:; Continue?
```

Replace the volume in Unit 0 with distribution volume 1 or 1-T (write protected, if possible).

```
Y(RET)
.
```

Remove the volume from Unit 1, label it "backup RT–11 V04 3/7," write protect it, if possible, and insert another initialized blank volume in Unit 1. Repeat these procedures to copy the rest of the distribution volumes.

Now halt the processor. Replace distribution volume 1 or 1-T with backup 1 or 1-T (write protected) in Unit 0. Write protect the distribution volumes (if possible) and store them. Use the backup copies to build a working system.

Use the hardware bootstrap to boot backup 1 or 1-T.

```
RT-11SJ V04.00
(Followed by any startup file commands.)
.
```

NOTE

> If the backup volume does not boot, repeat the procedure to create the backup volume.

Set the date, using the following command (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
.DATE dd-mmm-yy(RET)
.
```

Next, remove the protection from all the files on the backup volumes. The files on the distribution volumes have been protected to prevent you from accidentally deleting them (see the *RT–11 System User's Guide* for a description of file protection). When you copied the files to the backup volumes, RT–11 also copied the protection (note the P that prints next to the file size in the directory). For the rest of these procedures, you need to remove the protection from the backup volumes. Type the following command to remove it from the files on backup volume 1 or 1-T.

```
.RENAME/SYSTEM/NOPROTECTION *.* *.*(RET)
 Files renamed:
DK:aaaaaa.ttt  to DK:aaaaaa.ttt
DK:bbbbbb.ttt  to DK:bbbbbb.ttt
DK:cccccc.ttt  to DK:cccccc.ttt
DK:dddddd.ttt  to DK:dddddd.ttt
           .
           .
           .
DK:zzzzzz.ttt  to DK:zzzzzz.ttt
.
```

Insert backup volume 2 in Unit 1 and type the following command:

```
.RENAME/SYSTEM/NOPROTECTION xx1:*.* xx1:*.*(RET)
 Files renamed:
xx1:aaaaaa.ttt  to xx1:aaaaaa.ttt
xx1:bbbbbb.ttt  to xx1:bbbbbb.ttt
xx1:cccccc.ttt  to xx1:cccccc.ttt
xx1:dddddd.ttt  to xx1:dddddd.ttt
           .
           .
           .
xx1:zzzzzz.ttt  to xx1:zzzzzz.ttt
.
```

Replace backup volume 2 with backup volume 3 in Unit 1 and repeat this command. In the same way, remove protection from the files on the rest of the backup volumes.

3.3 Installing Mandatory Patches

To make sure that RT-11 operates correctly, install mandatory patches at this point. Mandatory patches are critical to system or component operation. They correct software errors discovered since the software was released. Chapter 2 describes how to identify the appropriate patches.

NOTE

> If you are using Autopatch to install mandatory patches, do not read the remainder of this section. Instead, turn to the *RT-11 Autopatch User's Guide* for assistance.

Most published patches are binary patches that you install with the Save Image Patch Program (SIPP). In the event that monitors or handlers require patches, the patches may be published in both binary and source form. If you later perform the system generation process to create your own monitors and handlers, you need source files with any patches applied. Therefore, you must install all source patches, using the Source Language Patch (SLP) utility program.

To install binary patches, invoke SIPP, as indicated, and enter the values specified in the published patch.

```
.R SIPP<RET>
*filnam.typ/C<RET>
Base?        xxxx<RET>
Offset?      xx<RET>
      Base         Offset      Old      New?
    xxxxxx         xxxxxx    xxxxxx     xxxxxx<RET>
    xxxxxx         xxxxxx    xxxxxx     <CTRL/Y><RET>
Checksum?    xxxxx<RET>
*<CTRL/C>
.
```

The input file, filnam.typ, is the component to be patched. If the input file is not on your system volume, you must identify the device unit on which it resides (xxn:filnam.typ). Respond to SIPP's prompts with the values specified in the published patch. The base and offset describe the locations to be patched. The patch may look different from this example if the component is overlaid. CTRL/Y causes SIPP to ask for the checksum. If the checksum matches, SIPP installs the patch and prompts with an asterisk. If you have typed the patch exactly as published, the checksum will match. If the checksum does not match, issue ;V to display modified locations. Correct any errors. Then type CTRL/Y and RETURN. To patch components that are not on backup volume number 1, insert in Unit 1 the volume that contains the component to be patched. Then specify that unit in the command string to SIPP:

```
*xx1:filnam.typ<RET>
```

Refer to the *RT–11 System User's Guide* for more information about SIPP.

To install source patches, create the file, using any editor, exactly as published. Type the command indicated, but insert the correct unit number if the file being patched is not on the system volume. The following is an example of a source patch.

Use a text editor to create the SLP command file, filnam.SLP, as follows.

```
-563
<TAB>DEC<TAB>R5
-660,661
<TAB>BIT<TAB> SPECL$!FILST$,@R2
<TAB>BEQ<TAB>17$
-682
<TAB>MOV<TAB>R3,R0
-700,701
<TAB>BIT<TAB> SPECL$!FILST$,@R2
<TAB>BEQ<TAB>23$
/
```

Then use the SLP utility to patch filnam.MAC as follows:

```
.R SLP<RET>
*filnam.MAC=filnam.MAC,filnam.SLP<RET>
*<CTRL/C>
.
```

Use these procedures to make all the mandatory binary and source patches. Patch each component separately, following the sequence specified in the *RT–11 Software Dispatch Review.*

NOTE

> DIGITAL strongly recommends that you analyze all future patches as soon as they are published. Install mandatory patches immediately. Ascertain whether optional patches are useful for your application. Often, you can save much time and effort through such analysis, because these patches may solve a problem you have already experienced.

Once you have created your working system (procedure follows), store the patched backup. Then, as additional mandatory patches are published in the *RT–11 Software Dispatch,* you can install them on the backup and copy the patched component to the working system. In this way, you can make sure that you have up-to-date, patched versions of all RT–11 components, even if your working system is destroyed.

3.4 Creating the Working System from Chosen Components

Once you have chosen your system components (Section 2.3) and have planned the best arrangement of them on volumes (Section 2.4), you can create the working system. Create the working system by copying selected components to initialized, blank volumes.

Start by initializing a number of blank volumes. Follow the same procedure that you used in Section 3.2. Insert a write enabled, blank volume in Unit 1 (with the system booted from Unit 0), and type INITIALIZE/BADBLOCKS xx1:. Repeat the process to create as many initialized blank volumes as you need for the system that you have planned.

NOTE

> If you want to create dummy bad blocks on cartridges to avoid excessive rewinds (as described in Section 2.4.3), do so at this point.

Then, use the COPY command with the SYSTEM option (for .SYS files) to copy selected files from backup volume 1 to the volume that becomes your working system volume.

```
.COPY/SYSTEM xx0:filnam.typ xx1:filnam.typ(RET)
.
```

Or you can use the following command to avoid typing numerous file specifications.

```
.COPY/SYSTEM/QUERY xx0: xx1:(RET)
  Files copied:
xx0:aaaaaa.ttt to xx1:aaaaaa.ttt? Y(RET) (to include a specific file)
xx0:bbbbbb.ttt to xx1:bbbbbb.ttt? N(RET) (to exclude a specific file)
```

(and so on)

To copy files from nonbootable volumes, alternate volumes.

Use the SET command to set the USR to NOSWAP.

```
.SET USR NOSWAP(RET)
.
```

Type the following command, where filnam.typ is the name of the file you want to copy. In this case, you cannot use the QUERY option; you must specify individual files.

```
.COPY/SYSTEM/WAIT xx1:filnam.typ xx0:filnam.typ(RET)
Mount input volume in xx1:; Continue?
```

Place the volume containing the file you want to copy in Unit 1.

```
Y(RET)

Mount output volume in xx0:; Continue?
```

Replace the system volume in Unit 0 with the volume to which you want to copy filnam.typ.

```
Y(RET)

Mount system volume in xx0:; Continue?
```

Replace the volume in Unit 0 with backup volume number 1 (write protected, if possible).

```
Y(RET)
.
```

Repeat this procedure to copy all the files for the working system volume. When you have copied all the files you have planned for the working system volume, label it "RT–11 working system V04 1/x" (where x is the number of volumes in your working system). Repeat these procedures to create the other volumes in the working system.

When you have created and labeled all the working system volumes, you can permit the USR to swap again.

```
.SET USR SWAP(RET)
.
```

3.5 Installing the Bootstrap on Any Volumes That Need to Be Bootable

Once you have created your system, you need to install the bootstrap on any volumes that must be bootable (that is, that you can use as the system volume). Generally, any volume that includes a monitor file and system device handler should be bootable (but do not forget that the volume would need SWAP.SYS and, for the SJ monitor, TT.SYS).

Insert in Unit 1 the volume on which you need to install the bootstrap. In the command, yy is BL, SJ, or FB.

```
.COPY/BOOT xx1:RT11yy.SYS xx1:(RET)
.
```

In this command, you need to identify the device on which the monitor that contains the bootstrap information resides, the name of that monitor file, and the device on which you need to install the bootstrap. This command copies bootstrap information from the monitor file to blocks 0 and 2 through 5 of the same volume.

Then, insert working system volume number 1 in Unit 0 and use the hardware bootstrap to boot your working system.

```
RT-11xx V04.00
(Followed by any startup file commands.)
.
```

Set the date, using the following command (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
DATE dd-mmm-yy(RET)
.
```

Store the patched backup volumes for future patching purposes.

3.6 Customizing the System

You may want to make certain customizations (described in Section 2.8) to the distributed RT–11 components. At this point, perform the procedures to implement any of these customizations. Table 1–4 summarizes the available customizations and directs you to the section in Chapter 2 that describes a particular customization and the procedure for implementing it.

NOTE

> Some RT–11 configurations can perform the system generation process to implement additional customizations. See Section 3.10.

3.7 Compressing Each Volume

DIGITAL recommends that you compress each working system volume to make its free space contiguous. Consolidating free space allows you to use space on the volume that would otherwise be too fragmented to be usable. However, if your volumes are cartridges and you create bad blocks to avoid excessive rewinds, the amount of contiguous free space possible is limited (refer to Section 2.4.3).

Continue to run RT–11 from Unit 0, and use the SQUEEZE command to compress free space. (The volume must be write enabled.) The squeeze operation does not move files with the .BAD file type.

```
.SQUEEZE xx0:RET
xx0:/Squeeze; Are you sure? YRET
```

There may be a delay as the system compresses the volume.

```
RT-11xx V04.00x
(Followed by any startup file commands.)
.
```

The system automatically reboots when you compress a system volume.

Then insert the next volume that you need to compress (write enabled) in Unit 1.

```
.SQUEEZE xx1:RET
xx1:/Squeeze; Are you sure? YRET
.
```

Replace the volume in Unit 1 with the next one you need to compress, and repeat this procedure for all the volumes you need to compress.

NOTE

When you compress a volume with system files (.SYS), PIP warns you to reboot. Do reboot as advised. When you compress a system volume, the system automatically reboots.

3.8 Preserving the Working System

Once you build a satisfactory working system, DIGITAL recommends that you protect all the files in the working system and preserve the system on backup volumes.

Use the following command to protect all the files on the system volume.

```
.RENAME/SYSTEM/PROTECTION *.* *.*(RET)
 Files renamed:
DK:aaaaaa.ttt  to DK:aaaaaa.ttt
DK:bbbbbb.ttt  to DK:bbbbbb.ttt
DK:cccccc.ttt  to DK:cccccc.ttt
DK:dddddd.ttt  to DK:dddddd.ttt
              .
              .
              .
DK:zzzzzz.ttt  to DK:zzzzzz.ttt
.
```

To protect files on other volumes in the working system, insert each volume in Unit 1 and use the following command:

```
.RENAME/SYSTEM/PROTECTION xx1:*.* xx1:*.*(RET)
 Files renamed:
xx1:aaaaaa.ttt  to xx1:aaaaaa.ttt
xx1:bbbbbb.ttt  to xx1:bbbbbb.ttt
xx1:cccccc.ttt  to xx1:cccccc.ttt
xx1:dddddd.ttt  to xx1:dddddd.ttt
              .
              .
              .
xx1:zzzzzz.ttt  to xx1:zzzzzz.ttt
.
```

Next, copy the working system to backup volumes. Insert a blank volume (write enabled) in Unit 1 with RT–11 still booted from Unit 0). Use the INITIALIZE/BADBLOCKS command to initialize the blank volume. Then repeat the process to initialize the appropriate number of volumes.

Copy all the files in your working system. You can use the SQUEEZE/OUTPUT:xxn: command to copy any bootable volumes. Remember that you must use SQUEEZE/WAIT/OUTPUT:xxn: and change volumes to copy the volumes that are not bootable (see Section 3.2). Also remember to copy the bootstrap to any volumes that need to be bootable.

Write protect (if possible) the backup volumes, and store them. If you ever need to restore the working system, you can make copies of the backup.

3.9 Testing the Working System

Once you have built and preserved the working system, you can execute the following demonstration to test the system. This demonstration does not serve as a comprehensive system exercise; however, because it uses several major system components, it does serve as a minimal integrity check. Moreover, DIGITAL considers your system officially installed if the demonstration runs without error.

To execute this demonstration, your working system must include at least the following components. To perform the foreground/background monitor exercise, you need a terminal with a bell.

SWAP.SYS
RT11SJ.SYS
RT11FB.SYS (if you do the foreground/background exercise)
xx.SYS (system device handler)
TT.SYS (for SJ monitor)
LP.SYS (if appropriate)
EDIT.SAV
MACRO.SAV
SYSMAC.SML
LINK.SAV
PIP.SAV
DUP.SAV
DIR.SAV
DEMOBG.MAC
DEMOFG.MAC

You can do only the single-job monitor exercise to test the single-job monitor, or you can do both exercises to test both the single-job monitor and the foreground/background monitor.

3.9.1 Single-Job Monitor Exercise

For the single-job monitor exercise, you edit, assemble, link, and execute a small program (DEMOBG.MAC). When you execute DEMOBG, it displays a message on the terminal.

3.9.1.1 Bootstrapping the SJ Monitor and Getting Started — Write enable the system volume. Then, bootstrap the system and set the date, using the following command (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
RT-11SJ V04.00x
(Followed by any startup file commands.)
.DATE dd-mmm-yy(RET)
.
```

NOTE

> If your configuration includes a VT11 or VS60 display processor and scope, shift system output to the display scope. Type GT ON. Verify that the scope is on by turning the BRIGHTNESS knob to an adequate level. You can still enter commands at the keyboard, but the echo is on the screen.

Insert a write enabled, blank volume in Unit 1.

```
.ASSIGN xx1: DK:<RET>

.INITIALIZE/BADBLOCKS xx1:<RET>
xx1:/Initialize; Are you sure? Y<RET>
?DUP-I-No bad blocks detected xx1:
.
```

Display the directory of the system volume on the terminal. The directory varies according to your particular working system. As long as a directory prints, you need not worry if it does not match the one in the following example.

```
.DIRECTORY/BRIEF/COLUMNS:1 SY:<RET>
 dd-mmm-yy
RT11SJ.SYS
SWAP  .SYS
LP    .SYS
TT    .SYS
DD    .SYS
SIPP  .SAV
EDIT  .SAV
MACRO .SAV
SYSMAC.SML
LINK  .SAV
ODT   .OBJ
PIP   .SAV
     .
     .
     .
xxx Files, bbb Blocks
fff Free blocks
.
```

NOTE

> If you have shifted output to a scope and the directory scrolls by too quickly to read, type CTRL/S to stop the display and CTRL/Q to restart it.

3.9.1.2 Editing the Demonstration Program — Next, use the text editor to modify the demonstration program, DEMOBG.MAC. If DEMOBG.MAC is a protected file, remove the protection before making the edits (RENAME/NOPROTECTION SY:DEMOBG.MAC SY:DEMOBG.MAC).

```
.EDIT SY:DEMOBG.MAC<RET>
*F;<TAB>.ASCII<ESC><ESC>
*0AD<ESC><ESC>
*EX<ESC><ESC>
.
```

3.9.1.3 Assembling the Demonstration Program — To assemble DEMOBG.MAC and obtain a listing, make sure that your configuration has a line printer, that is on-line and ready.

```
.ASSIGN LP: LST:<RET>
.
```

NOTE

If your configuration does not include a line printer, use the console terminal.

```
.ASSIGN TT: LST:(RET)
.
```

Assemble DEMOBG.MAC as follows:

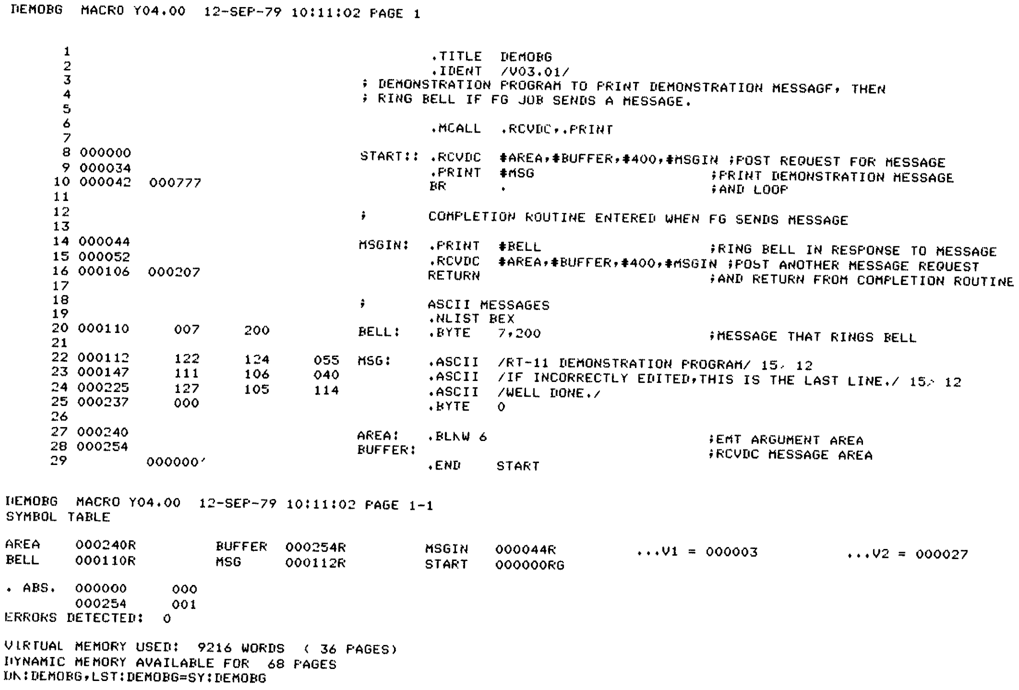
```
.MACRO/LIST:LST: SY:DEMOBG(RET)
```

(see Figure 3–2)

If any errors occur when you assemble DEMOBG.MAC, you have incorrectly edited the file and should repeat the edits. Use the backup demonstration program.

```
.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC(RET)
.
```

Figure 3–2: DEMOBG Assembly Listing



```
DEMOBG  MACRO Y04.00  12-SEP-79 10:11:02 PAGE 1

                                           .TITLE  DEMOBG
                                           .IDENT  /V03.01/
                                   ; DEMONSTRATION PROGRAM TO PRINT DEMONSTRATION MESSAGF, THEN
                                   ; RING BELL IF FG JOB SENDS A MESSAGE.

                                           .MCALL  .RCVDC,.PRINT

   000000                          START:: .RCVDC  #AREA,#BUFFER,#400,#MSGIN ;POST REQUEST FOR MESSAGE
   000034                                  .PRINT  #MSG                      ;PRINT DEMONSTRATION MESSAGE
   000042  000777                          BR      .                         ;AND LOOP

                                   ;       COMPLETION ROUTINE ENTERED WHEN FG SENDS MESSAGE

   000044                          MSGIN:  .PRINT  #BELL                     ;RING BELL IN RESPONSE TO MESSAGE
   000052                                  .RCVDC  #AREA,#BUFFER,#400,#MSGIN ;POST ANOTHER MESSAGE REQUEST
   000106  000207                          RETURN                            ;AND RETURN FROM COMPLETION ROUTINE

                                   ;       ASCII MESSAGES
                                           .NLIST BEX
   000110     007     200          BELL:   .BYTE   7,200                     ;MESSAGE THAT RINGS BELL

   000112     122     124     055  MSG:    .ASCII  /RT-11 DEMONSTRATION PROGRAM/ 15, 12
   000147     111     106     040          .ASCII  /IF INCORRECTLY EDITED,THIS IS THE LAST LINE./ 15, 12
   000225     127     105     114          .ASCII  /WELL DONE./
   000237     000                          .BYTE   0

   000240                          AREA:   .BLKW 6                           ;EMT ARGUMENT AREA
   000254                          BUFFER:                                   ;RCVDC MESSAGE AREA
           000000'                         .END    START

DEMOBG  MACRO Y04.00  12-SEP-79 10:11:02 PAGE 1-1
SYMBOL TABLE

AREA    000240R      BUFFER  000254R      MSGIN   000044R      ...V1 = 000003      ...V2 = 000027
BELL    000110R      MSG     000112R      START   000000RG

. ABS.  000000     000
        000254     001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  9216 WORDS  ( 36 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  68 PAGES
DK:DEMOBG,LST:DEMOBG=SY:DEMOBG
```

3.9.1.4 Linking the Demonstration Program — Now, link the program DEMOBG.

```
.LINK DEMOBG(RET)
.
```

3.9.1.5 Running the Demonstration Program — Run the program DEMOBG to check the results of the first exercise.

```
.RUN DEMOBG<RET>
RT-11 DEMONSTRATION PROGRAM
IF INCORRECTLY EDITED, THIS IS THE LAST LINE.
WELL DONE.
<CTRL/C>
<CTRL/C>
.
```

NOTE

> If your configuration includes a graphics display, turn it off at this point. Type GT OFF.

If you incorrectly edited the file, you can repeat this exercise, although you can continue without correcting the file. If, however, you want to repeat the exercise, begin by using the backup demonstration program.

```
.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC<RET>
.
```

Then, repeat the editing procedure.

3.9.2 Foreground/Background Monitor Exercise

For the foreground/background monitor exercise, assemble a second program (DEMOFG.MAC), link it for the foreground, and execute it in conjunction with DEMOBG (procedures follow). DEMOFG is a small foreground program that sends a message every two seconds to DEMOBG (running in the background), telling it to ring the terminal bell. Besides printing the terminal message used in the single-job exercise, DEMOBG recognizes these messages and rings the bell once for each message sent.

Although DEMOFG is always active, sending messages to the background every two seconds, this exercise can execute other programs in the background besides DEMOBG. The circuit is complete and messages are successfully received and honored only when DEMOBG is active. During those periods when DEMOBG is not running, DEMOFG enters the messages in the monitor message queue. Once you restart DEMOBG in the background, the system immediately releases all the messages queued since the last forced exit, resulting in many successive bell rings. When the queue is empty, the normal send/receive cycle resumes and the bell rings every two seconds as each current message is sent and honored.

3.9.2.1 Bootstrapping the FB Monitor and Getting Started — Make sure that the RT–11 single-job monitor is running. Then boot the foreground/background monitor.

```
.BOOT RT11FB.SYS<RET>
RT-11FB V04.00x
(Followed by any startup file commands.)
.
```

NOTE

> If your configuration includes a VT11 or VS60 display processor and scope, shift system output to the display scope. Type GT ON. Verify that the scope is on by turning the BRIGHTNESS knob to an adequate level. You can still enter commands at the keyboard, but the echo is on the screen.

```
.ASSIGN xx1: DK:(RET)
.
```

Now, enter the time of day, using the following command. In the command, hh:mm:ss is the hour, minutes, and seconds in the form 13:12:00 (1:12 p.m.).

```
.TIME hh:mm:ss(RET)
.
```

3.9.2.2 Assembling the Demonstration Program — Assemble the foreground demonstration program, DEMOFG.MAC.

```
.MACRO SY:DEMOFG(RET)
ERRORS DETECTED:  0
.
```

3.9.2.3 Linking the Demonstration Program — Link DEMOFG for the foreground.

```
.LINK/FOREGROUND DEMOFG(RET)
.
```

3.9.2.4 Running the Demonstration Programs — Start DEMOFG as the foreground job.

```
.FRUN DEMOFG(RET)
F>
FOREGROUND DEMONSTRATION PROGRAM
SENDS A MESSAGE TO THE BACKGROUND PROGRAM DEMOBG
EVERY 2 SECONDS, TELLING IT TO RING THE BELL.
(CTRL/B)
B>
```

DEMOFG is running and queuing the message for DEMOBG every two seconds. Now execute DEMOBG in the background and receive the messages.

```
RUN DEMOBG(RET)
```

(The bell rings quickly several times, then once every two seconds.)

```
RT-11 DEMONSTRATION PROGRAM
IF INCORRECTLY EDITED, THIS IS THE LAST LINE.
WELL DONE.
```

Execute a DIRECTORY command in the background to obtain a directory listing.

```
[CTRL/C]
[CTRL/C]
```

(The bell stops.)

```
.DIRECTORY[RET]
dd-mmm-yy
```

(The directory of the device DK prints on the terminal.)

```
.
```

Rerun DEMOBG to collect all the foreground messages queued while the directory was printing.

```
.RUN DEMOBG[RET]
```

(The bell rings several times in rapid succession, then rings once every two seconds.)

```
RT-11 DEMONSTRATION PROGRAM
IF INCORRECTLY EDITED, THIS IS THE LAST LINE.
WELL DONE.
[CTRL/C]
[CTRL/C]
.
```

(The bell stops.)

Now, stop the foreground program and remove it from memory.

```
[CTRL/F]
F>
[CTRL/C]
[CTRL/C]
B>
UNLOAD F[RET]
.
```

NOTE

> If your configuration includes a graphics display, turn it off at this point. Type GT OFF.

If you completed these exercises without error, your system has passed this minimal test and you can consider it successfully installed.

3.10 Performing the System Generation Process

If you have decided that you need RT–11 features that are available only if you generate your own monitor(s) and handlers, you can perform the system generation process at this point. You should have thoroughly studied Chapter 1 to make this decision and to establish that you can perform the system generation process on your particular hardware configuration. Turn to Part III (Chapters 8, 9, and 10) for guidance in planning and performing system generation.

Chapter 4
Installing a System Distributed on a Small Device to Run on a Hard Disk

If RT–11 was distributed to you on single-density diskette or DECtape II cartridge and you intend to build a system to run on hard disk, perform the procedures described in this chapter. These procedures cover minor variations that depend on the specific device you have (diskette or cartridge) and assume that your configuration includes two hard disk drives.

NOTE

> If your hardware configuration includes a VT100 terminal, be sure to set auto XOFF/XON in SETUP mode before attempting to bootstrap RT–11. Never set TT NOPAGE when you use this terminal. Refer to your hardware manuals for more information about these settings.

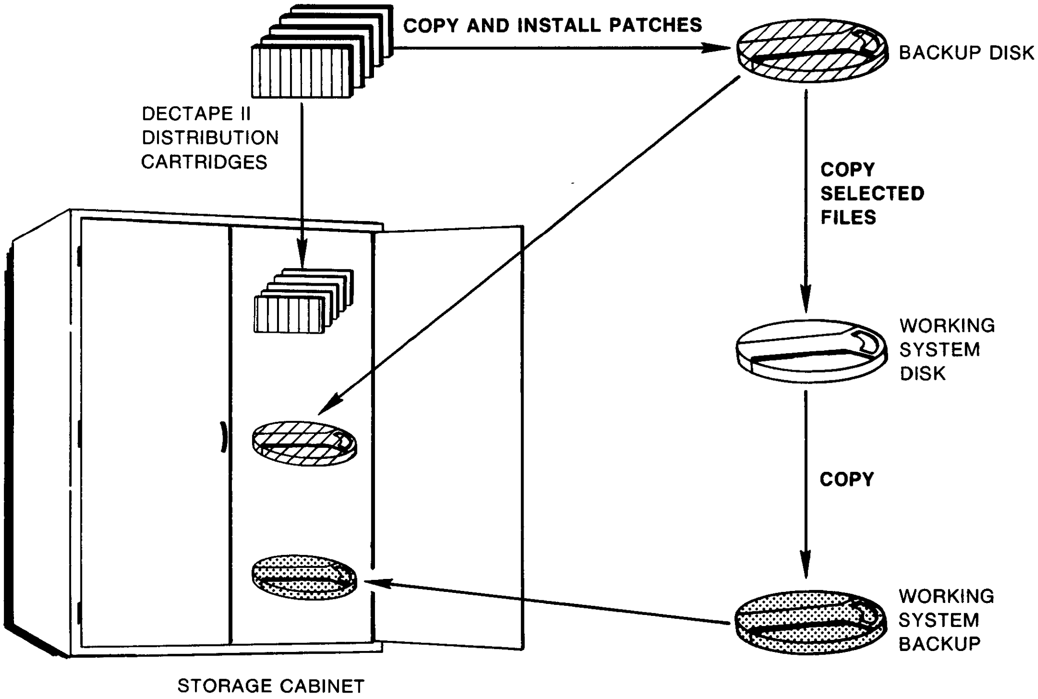
To install your system, perform the steps summarized in the following list. Sections 4.1 through 4.11 describe the procedures involved in each step. Figure 4–1 shows the various backup volumes you create when you install RT–11.

1. Bootstrap the distribution volume.
2. Copy the distribution volumes to the disk.
3. Preserve the distribution volumes.
4. Install mandatory patches.
5. Create the system from chosen components.
6. Install the bootstrap on the disk.
7. Customize the system.
8. Compress the disk.

9. Preserve the working system.
10. Test the working system.
11. If appropriate, perform the system generation process.

The following sections correspond to each of these steps and describe in detail the procedures you must perform to complete each step.

Figure 4–1: Sample Backup Disks



4.1 Bootstrapping the Distribution Volume

The first procedure you perform when installing RT–11 is bootstrapping the distribution volume.

Begin by making sure that the processor is powered up but not running. Insert the distribution volume labeled 1/7 (write protected, if possible) in RX01 diskette or TU58 DECtape II cartridge Unit 0. If your device is RX01 diskette, it has the device name DX:. If your device is DECtape II cartridge, it has the device name DD:.

NOTE

> Store the volume labeled 1-T/7. This volume is bootable only on a PDT–11 Intelligent terminal (device name PD:).

Use the hardware bootstrap to boot the volume. (If your configuration does not include a hardware bootstrap, see Appendix E for toggle-in software bootstraps.)

RT–11 should respond with the following message if you have successfully bootstrapped the volume:

```
RT-11SJ V04.00
(Followed by any startup file commands.)
.
```

Set the date, using the following command (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
.DATE dd-mmm-yy(RET)
.
```

4.2 Copying the Distribution Volumes to the Disk

With the RT–11 system running, copy the distribution volumes to the disk that will serve as your working system disk.

If the disk is an RK05 disk, format the disk, then initialize it and cover any bad blocks on it.

If the disk is another type, initialize the disk and replace or cover bad blocks. You have a choice of replacing or covering bad blocks if your disk is RK06, RK07, or RL01. If your disk is another type, you should cover bad blocks. To replace bad blocks, use the INITIALIZE/REPLACE command, and to cover bad blocks, use the INITIALIZE/BADBLOCKS command (procedures follow). In the commands, xx is the permanent device name for your disk. If the disk contains bad blocks, the **?DUP-I-Bad block detected nnnnnn** message appears on the terminal, but you can use the disk.

RK05 disk

```
.FORMAT RK0:(RET)
RK0:/FORMAT-Are you sure?Y(RET)
?FORMAT-I-Formatting complete

.INITIALIZE/BADBLOCKS RK0:(RET)
RK0:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected RK0:
```

Other type disk

```
.INITIALIZE/BADBLOCKS xx0:(RET)
xx0:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected xx0:
```

or

```
.INITIALIZE/REPLACE xx0:(RET)
xx0:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected xx0:
```

There is a delay as the system scans the disk for bad blocks and creates a new directory. The monitor dot appears when this process is complete.

```
.
```

The next step in the preservation process is to copy all the files from distribution volume number 1 to the initialized disk. The following command transfers files from the distribution volume to the disk and consolidates all the empty space at the end of the disk. In the command, xx is the permanent device name for your disk and yy is the device name for the distribution device.

```
.SQUEEZE/OUTPUT:xx0: yy0:(RET)
```

NOTE

> Both the SQUEEZE command with the /OUTPUT:xxn: option and the COPY command transfer files from one device to another. SQUEEZE/OUTPUT:xxn: consolidates free space on the device at the same time. The procedures in this manual use the command that is most appropriate and most efficient for an individual operation. For a better understanding of all RT–11 keyboard monitor commands, refer to the *RT–11 System User's Guide.*

There may be a delay while the system performs this operation.

You must also copy the rest of the distribution volumes to the disk. Insert distribution volume number 2 (write protected, if possible) in RX01 or TU58 Unit 1. In the following command, xx is the permanent device name for your disk and yy is the name for your distribution device.

```
.COPY/SYSTEM yy1: xx0:(RET)
 Files copied:
yy1:aaaaaa.ttt  to xx0:aaaaaa.ttt
yy1:bbbbbb.ttt  to xx0:bbbbbb.ttt
               .
               .
               .
yy1:zzzzzz.ttt  to xx0:zzzzzz.ttt
.
```

Insert distribution volume number 3 (write protected, if possible) in RX01 or TU58 Unit 1, and repeat this command. Copy all the distribution volumes to your disk in this way. Write protect each volume, if possible, before you insert it.

NOTE

> For optimal system performance, copy the distribution volumes in order.

Next, copy the bootstrap to the disk and compress the disk. Use the following commands, where xx is the permanent device name for your disk.

```
.COPY/BOOT xx0:RT11SJ.SYS xx0:(RET)

.SQUEEZE xx0:(RET)
xx0:/Squeeze; Are you sure? Y(RET)
.
```

4.3 Preserving the Distribution Volumes

Halt the processor, remove the distribution volumes, and write protect them (if possible). Store them, as a safety measure in case of machine failure or human error. Use the disk to build a working system. Leave the write enabled disk in Unit 0. Use the hardware bootstrap to boot RT-11 from your disk.

```
RT-11xx V04.00
(Followed by any startup file commands.)
.
```

NOTE

If the disk does not boot, repeat the procedures to this point.

Set the date, using the following command (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
.DATE dd-mmm-yy(RET)
.
```

Next, remove the protection from the files on the backup disk. The files on the distribution volumes have been protected to prevent you from accidentally deleting them (see the *RT-11 System User's Guide* for a description of file protection). When you used the SQUEEZE command to copy the files to the backup disk, RT-11 also copied the protection (note the P that prints next to the file size in the directory). For the rest of these procedures, you need to remove the protection from the backup disk. Use the following command.

```
.RENAME/SYSTEM/NOPROTECTION *.* *.*(RET)
 Files renamed:
DK:aaaaaa.ttt  to DK:aaaaaa.ttt
DK:bbbbbb.ttt  to DK:bbbbbb.ttt
DK:cccccc.ttt  to DK:cccccc.ttt
DK:dddddd.ttt  to DK:dddddd.ttt
          .
          .
          .
DK:zzzzzz.ttt  to DK:zzzzzz.ttt
.
```

4.4 Installing Mandatory Patches

To make sure that RT-11 operates correctly, install mandatory patches at this point. Mandatory patches are critical to system or component operation. They correct software errors discovered since the software was released. Chapter 2 describes how to identify the appropriate patches.

NOTE

If you are using Autopatch to install mandatory patches, do not read the remainder of this section. Instead, turn to the *RT-11 Autopatch User's Guide* for assistance.

Most published patches are binary patches that you install with the Save Image Patch Program (SIPP). In the event that monitors or handlers require patches, the patches may be published in both binary and source form. If you later perform the system generation process to create your own monitors and handlers, you need source files with any patches applied. Therefore, you must install all source patches, using the Source Language Patch (SLP) utility program. To install binary patches, invoke SIPP, as indicated, and enter the values specified in the published patch.

```
.R SIPP(RET)
*filnam.typ/C(RET)
Base?          xxxx(RET)
Offset?        xx(RET)
     Base          Offset       Old      New?
   xxxxxx          xxxxxx    xxxxxx      xxxxxx(RET)
   xxxxxx          xxxxxx    xxxxxx      (CTRL/Y)(RET)
Checksum?      xxxxx(RET)
*(CTRL/C)
```

The input file, filnam.typ, is the component to be patched. Respond to SIPP's prompts with the values specified in the published patch. The base and offset describe the locations to be patched. The patch may look different from this example if the component is overlaid. CTRL/Y causes SIPP to ask for the checksum. If the checksum matches, SIPP installs the patch and prompts with an asterisk. If you have typed the patch exactly as published, the checksum will match. If the checksum does not match, issue ;V to display modified locations. Correct any errors. Then type CTRL/Y and RETURN.

Refer to the *RT–11 System User's Guide* for more information about SIPP.

To install source patches, create the file, using any editor, exactly as published. Type the command indicated, but insert the correct unit number if the file being patched is not on the system volume. The following is an example of a source patch.

Use a text editor to create the SLP command file, filnam.SLP, as follows.

```
-563
(TAB)DEC(TAB)R5
-660,661
(TAB)BIT(TAB) SPECL$!FILST$,@R2
(TAB)BEQ(TAB)17$
-682
(TAB)MOV(TAB)R3,R0
-700,701
(TAB)BIT(TAB) SPECL$!FILST$,@R2
(TAB)BEQ(TAB)23$
/
```

Then use the SLP utility to patch filnam.MAC as follows:

```
.R SLP(RET)
*filnam.MAC=filnam.MAC,filnam.SLP(RET)
*(CTRL/C)
```

Use these procedures to make all the mandatory binary and source patches. Patch each component separately, following the sequence specified in the *RT–11 Software Dispatch Review.*

NOTE

> DIGITAL strongly recommends that you analyze all future patches as soon as they are published. Install mandatory patches immediately. Ascertain whether optional patches are useful for your application. Often, you can save much time and effort through such analysis, because these patches may solve a problem you have already experienced.

Once you have created your working system (procedure follows), label and store the patched backup disk. Then, as additional mandatory patches are published in the *RT–11 Software Dispatch*, you can install them on the backup and copy the patched component to the working system. In this way, you will be sure that you have up-to-date, patched versions of all RT–11 components, even if your working system is destroyed.

4.5 Creating the Working System from Chosen Components

Once you have chosen your system components (Section 2.3) and have planned the best arrangement of them on volumes (Section 2.4), you can create the working system. You create the working system by copying selected components to another disk or disks.

Mount a blank disk in disk Unit 1 and initialize it. If the disk is an RK05 disk, do not forget to format it before you initialize it.

RK05 disk

```
.FORMAT RK1:(RET)
RK1:/FORMAT-Are you sure?Y(RET)
?FORMAT-I-Formatting complete

.INITIALIZE/BADBLOCKS RK1:(RET)
RK1:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected RK1:
```

Other type disk

```
.INITIALIZE/BADBLOCKS xx1:(RET)
xx1:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected xx1:
```

or

```
.INITIALIZE/REPLACE xx1:(RET)
xx1:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected xx1:
```

Copy the files you have selected from the disk in Unit 0 to the disk, which will become your working system disk, in Unit 1. If you use the following command, RT–11 queries you about all the files on the disk in Unit 0, and you choose the files it copies.

```
.COPY/SYSTEM/QUERY xx0: xx1:(RET)
  Files copied:
xx0:aaaaaa.ttt to xx1:aaaaaa.ttt? Y(RET) (to include a specific file)
xx0:bbbbbb.ttt to xx1:bbbbbb.ttt? N(RET) (to exclude a specific file)
(and so on)
```

4.6 Installing the Bootstrap on the Disk

Once you have created your system, you need to install the bootstrap on the system disk. In the following command, yy is BL, SJ, or FB.

```
.COPY/BOOT xx1:RT11yy.SYS xx1:(RET)
.
```

In this command, identify the device on which the monitor that contains the bootstrap information resides (your disk), the name of that monitor file (RT11BL, RT11SJ, RT11FB), and the device on which you need to install the bootstrap (your disk). This command copies bootstrap information from the monitor file to blocks 0 and 2 through 5 of the same volume.

Remove the disk from Unit 0 and store it for future patching. Mount the new working system disk in Unit 0, and use the hardware bootstrap to boot the working system disk.

```
RT-11xx V04.00
(Followed by any startup file commands.)
.
```

Set the date using the following command (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
.DATE dd-mmm-yy(RET)
.
```

4.7 Customizing the System

You may want to make certain customizations (described in Section 2.8) to the distributed RT–11 components. At this point, perform the procedures to implement any of these customizations. Table 1–4 summarizes the available customizations and directs you to the section in Chapter 2 that describes a particular customization and the procedure for implementing it.

NOTE

You can perform the system generation process to implement additional customizations. See Section 4.11.

4.8 Compressing the Disk

DIGITAL recommends that you compress the working system disk to make its free space contiguous. Consolidating free space allows you to use space on the disk that would otherwise be too fragmented to be usable.

Use the SQUEEZE command to compress free space. (The volume must be write enabled.) The squeeze operation does not move files with the .BAD file type.

```
.SQUEEZE xx0:(RET)
xx0:/Squeeze; Are you sure? Y(RET)
RT-11xx V04.00x
(Followed by any startup file commands.)
.
```

The system automatically reboots when you compress a system volume.

4.9 Preserving the Working System

Once you build a satisfactory working system, DIGITAL recommends that you protect all the files in the working system and back it up.

Use the following command to protect all the files on the system disk.

```
.RENAME/SYSTEM/PROTECTION *.* *.*(RET)
 Files renamed:
DK:aaaaaa.ttt   to DK:aaaaaa.ttt
DK:bbbbbb.ttt   to DK:bbbbbb.ttt
DK:cccccc.ttt   to DK:cccccc.ttt
DK:dddddd.ttt   to DK:dddddd.ttt
           .
           .
           .
DK:zzzzzz.ttt   to DK:zzzzzz.ttt
.
```

Now preserve the working system on the backup medium of your choice.

4.9.1 Procedure to Back Up the System on RX01 or TU58

Insert a blank RX01 or TU58 volume (write enabled) in Unit 1 (with RT–11 still booted from disk Unit 0). Use the INITIALIZE/BADBLOCKS command to initialize the blank volume. Then repeat the process to initialize a number of volumes.

Copy all the files in your working system, using the COPY/SYSTEM/QUERY command. Replace the diskette or cartridge in Unit 1 and repeat the command when you run out of space on each diskette or cartridge. In the command, xx is the permanent device name for your backup device and yy is the name for your disk.

```
.COPY/SYSTEM/QUERY yy0: xx1:(RET)
  Files copied:
yy0:aaaaaa.ttt to xx1:aaaaaa.ttt? Y(RET) (to include a specific file)
yy0:bbbbbb.ttt to xx1:bbbbbb.ttt? N(RET) (to exclude a specific file)
(and so on)
```

Then copy the bootstrap to the backup volume that is to be bootable. Remember that a bootable volume needs a monitor file, a system device handler, the file SWAP.SYS, and, for the SJ monitor, TT.SYS.

```
.COPY/BOOT xx1:RT11yy.SYS xx1:(RET)
.
```

Write protect (if possible) the backup volumes and store them. If you ever need to restore the working system, you can copy the backup volumes to the disk.

4.9.2 Procedure to Back Up the System on Another Disk

To back up your system on another disk, initialize a blank disk and copy all the files from the working system disk to the backup disk. In the following commands, xx is the permanent device name for your disk.

RK05 disk

```
.FORMAT RK1:(RET)
RK1:/FORMAT-Are you sure?Y(RET)
?FORMAT-I-Formatting complete

.INITIALIZE/BADBLOCKS RK1:(RET)
RK1:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected RK1:
```

Other type disk

```
.INITIALIZE/BADBLOCKS xx1:(RET)
xx1:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected xx1:
.
```

or

```
.INITIALIZE/REPLACE xx1:(RET)
xx1:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected xx1:
.
```

Copy all the files in your working system.

```
.SQUEEZE/OUTPUT:xx1 xx0:RET
.
```

Copy the bootstrap to the disk. In the following command, yy is BL, SJ, or FB.

```
.COPY/BOOT xx1:RT11yy.SYS xx1:RET
.
```

Store the backup disk. If you ever need to restore the working system, you can make a copy of the backup.

4.10 Testing the Working System

Once you have built and preserved the working system, you can execute the following demonstration to test the system. This demonstration does not serve as a comprehensive system exercise; however, because it uses several major system components, it does serve as a minimal integrity check. Moreover, DIGITAL considers your system officially installed if the demonstration runs without error.

To execute this demonstration, you should have at least 20 free blocks on your working system device and the working system must include at least the following components. To perform the foreground/background monitor exercise, you need a terminal with a bell.

SWAP.SYS
RT11SJ.SYS
RT11FB.SYS (if you do the foreground/background exercise)
xx.SYS (system device handler)
TT.SYS (for SJ monitor)
LP.SYS (if appropriate)
EDIT.SAV
MACRO.SAV
SYSMAC.SML
LINK.SAV
PIP.SAV
DUP.SAV
DIR.SAV
DEMOBG.MAC
DEMOFG.MAC

You can do only the single-job monitor exercise to test the single-job monitor or you can do both exercises to test both the single-job monitor and the foreground/background monitor.

4.10.1 Single-Job Monitor Exercise

For the single-job monitor exercise, you edit, assemble, link, and execute a small program (DEMOBG.MAC). When you execute DEMOBG, it displays a message on the terminal.

4.10.1.1 Bootstrapping the SJ Monitor and Getting Started — Bootstrap the system and set the date, using the following command (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
RT-11SJ V04.00x
(Followed by any startup file commands.)
.DATE dd-mmm-yy(RET)
.
```

NOTE

> If your configuration includes a VT11 or VS60 display processor and scope, shift system output to the display scope. Type GT ON. Verify that the scope is on by turning the BRIGHTNESS knob to an adequate level. You can still enter commands at the keyboard, but the echo is on the screen.

Display the directory of the system diskette on the terminal. The directory varies according to your particular working system. As long as a directory prints, you need not worry if it does not match the one in the following example.

```
.DIRECTORY/BRIEF/COLUMNS:1 SY:(RET)
  dd-mmm-yy
SWAP  .SYS
RT11SJ.SYS
LP    .SYS
TT    .SYS
DY    .SYS
SIPP  .SAV
EDIT  .SAV
MACRO .SAV
SYSMAC.SML
LINK  .SAV
ODT   .OBJ
PIP   .SAV
    .
    .
    .
xxx Files, bbb Blocks
fff Free blocks
.
```

NOTE

> If you have shifted output to a scope and the directory scrolls by too quickly to read, type CTRL/S to stop the display and CTRL/Q to restart it.

4.10.1.2 Editing the Demonstration Program — Use the text editor to modify the demonstration program, DEMOBG.MAC. If DEMOBG.MAC is a protected file, remove the protection before making the edits (RENAME/NOPROTECTION SY:DEMOBG.MAC SY:DEMOBG.MAC).

```
.EDIT SY:DEMOBG.MAC(RET)
*F;(TAB).ASCII(ESC)(ESC)
*OAD(ESC)(ESC)
*EX(ESC)(ESC)
.
```

4.10.1.3 Assembling the Demonstration Program — To assemble DEMOBG.MAC and obtain a listing, make sure that your configuration has a line printer that is on-line and ready.

```
.ASSIGN LP: LST:(RET)
.
```

NOTE

> If your configuration does not include a line printer, use the console terminal.
>
> ```
> .ASSIGN TT: LST:(RET)
> .
> ```

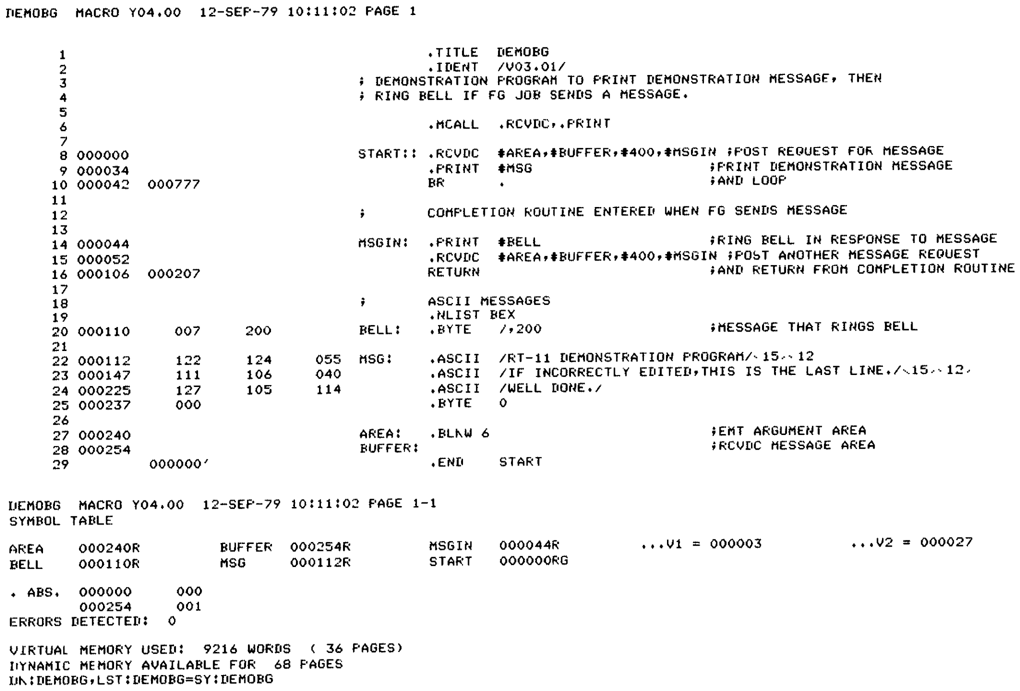
Assemble DEMOBG.MAC as follows:

```
.MACRO/LIST:LST: SY:DEMOBG(RET)
```
(see Figure 4–2)

If any errors occur when you assemble DEMOBG.MAC, you have incorrectly edited the file and should repeat the edits. Use the backup demonstration program.

```
.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC(RET)
.
```

Figure 4–2: DEMOBG Assembly Listing



```
DEMOBG  MACRO Y04.00  12-SEP-79 10:11:02 PAGE 1

                                          .TITLE  DEMOBG
                                          .IDENT  /V03.01/
                                  ; DEMONSTRATION PROGRAM TO PRINT DEMONSTRATION MESSAGE, THEN
                                  ; RING BELL IF FG JOB SENDS A MESSAGE.

                                          .MCALL  .RCVDC,.PRINT

     000000                       START:: .RCVDC  #AREA,#BUFFER,#400,#MSGIN ;POST REQUEST FOR MESSAGE
     000034                               .PRINT  #MSG                      ;PRINT DEMONSTRATION MESSAGE
     000042  000777                       BR      .                         ;AND LOOP

                                  ;       COMPLETION ROUTINE ENTERED WHEN FG SENDS MESSAGE

     000044                       MSGIN:  .PRINT  #BELL                     ;RING BELL IN RESPONSE TO MESSAGE
     000052                               .RCVDC  #AREA,#BUFFER,#400,#MSGIN ;POST ANOTHER MESSAGE REQUEST
     000106  000207                       RETURN                            ;AND RETURN FROM COMPLETION ROUTINE

                                  ;       ASCII MESSAGES
                                          .NLIST BEX
     000110     007      200      BELL:   .BYTE   /,200                     ;MESSAGE THAT RINGS BELL

     000112     122      124   055 MSG:   .ASCII  /RT-11 DEMONSTRATION PROGRAM/<15><12>
     000147     111      106   040        .ASCII  /IF INCORRECTLY EDITED,THIS IS THE LAST LINE./<15><12>
     000225     127      105   114        .ASCII  /WELL DONE./
     000237     000                       .BYTE   0

     000240                       AREA:   .BLKW 6                           ;EMT ARGUMENT AREA
     000254                       BUFFER:                                   ;RCVDC MESSAGE AREA
             000000'                      .END    START

DEMOBG  MACRO Y04.00  12-SEP-79 10:11:02 PAGE 1-1
SYMBOL TABLE

AREA    000240R       BUFFER  000254R       MSGIN   000044R       ...V1 = 000003        ...V2 = 000027
BELL    000110R       MSG     000112R       START   000000RG

. ABS.  000000      000
        000254      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  9216 WORDS  ( 36 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  68 PAGES
DK:DEMOBG,LST:DEMOBG=SY:DEMOBG
```

4.10.1.4 Linking the Demonstration Program — Now, link the program DEMOBG.

```
.LINK DEMOBG<RET>
.
```

4.10.1.5 Running the Demonstration Program — Run the program DEMOBG to check the results of the first exercise.

```
.RUN DEMOBG<RET>
RT-11 DEMONSTRATION PROGRAM
IF INCORRECTLY EDITED, THIS IS THE LAST LINE.
WELL DONE.
<CTRL/C>
<CTRL/C>
.
```

NOTE

> If your configuration includes a graphics display, turn it off at this point. Type GT OFF.

If you incorrectly edited the file, you can repeat this exercise, although you can continue without correcting the file. If, however, you want to repeat the exercise, begin by using the backup demonstration program.

```
.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC<RET>
.
```

Then, repeat the editing procedure.

4.10.2 Foreground/Background Monitor Exercise

For the foreground/background monitor exercise, assemble a second program (DEMOFG.MAC), link it for the foreground, and execute it in conjunction with DEMOBG (procedures follow). DEMOFG is a small foreground program that sends a message every two seconds to DEMOBG (running in the background), telling it to ring the terminal bell. Besides printing the terminal message used in the single-job exercise, DEMOBG recognizes these messages and rings the bell once for each message sent.

Although DEMOFG is always active, sending messages to the background every two seconds, this exercise can execute other programs in the background besides DEMOBG. The circuit is complete and messages are successfully received and honored only when DEMOBG is active. During those periods when DEMOBG is not running, DEMOFG enters the messages in the monitor message queue. Once you restart DEMOBG in the background, the system immediately releases all the messages queued since the last forced exit, resulting in many successive bell rings. When the queue is empty, the normal send/receive cycle resumes and the bell rings every two seconds as each current message is sent and honored.

4.10.2.1 Bootstrapping the FB Monitor and Getting Started — Make sure that the RT–11 single-job monitor is running. Then boot the foreground/background monitor.

```
.BOOT RT11FB.SYS(RET)
RT-11FB V04.00x
(Followed by any startup file commands.)
.
```

NOTE

> If your configuration includes a VT11 or VS60 display processor and scope, shift system output to the display scope. Type GT ON. Verify that the scope is on by turning the BRIGHTNESS knob to an adequate level. You can still enter commands at the keyboard, but the echo is on the screen.

Now, enter the time of day, using the following command. In the command, hh:mm:ss is the hour, minutes, and seconds in the form 13:12:00 (1:12 p.m.).

```
.TIME hh:mm:ss(RET)
.
```

4.10.2.2 Assembling the Demonstration Program — Assemble the foreground demonstration program, DEMOFG.MAC.

```
.MACRO SY:DEMOFG(RET)
ERRORS DETECTED: 0
.
```

4.10.2.3 Linking the Demonstration Program — Link DEMOFG for the foreground.

```
.LINK/FOREGROUND DEMOFG(RET)
.
```

4.10.2.4 Running the Demonstration Programs — Start DEMOFG as the foreground job.

```
.FRUN DEMOFG(RET)
F>
FOREGROUND DEMONSTRATION PROGRAM
SENDS A MESSAGE TO THE BACKGROUND PROGRAM DEMOBG
EVERY 2 SECONDS, TELLING IT TO RING THE BELL.
B>
```

DEMOFG is running and queuing the message for DEMOBG every two seconds. Now execute DEMOBG in the background and receive the messages.

```
RUN DEMOBG(RET)
```

(The bell rings quickly several times, then once every two seconds.)

```
RT-11 DEMONSTRATION PROGRAM
IF INCORRECTLY EDITED, THIS IS THE LAST LINE.
WELL DONE.
```

Execute a DIRECTORY command in the background to obtain a directory listing.

```
(CTRL/C)
(CTRL/C)
```

(The bell stops.)

```
.DIRECTORY(RET)
 dd-mmm-yy
```

(The directory of the device DK prints on the terminal.)

```
.
```

Rerun DEMOBG to collect all the foreground messages queued while the directory was printing.

```
.RUN DEMOBG(RET)
```

(The bell rings several times in rapid succession, then rings once every two seconds.)

```
RT-11 DEMONSTRATION PROGRAM
IF INCORRECTLY EDITED, THIS IS THE LAST LINE.
WELL DONE.
(CTRL/C)
(CTRL/C)
.
```

(The bell stops.)

Now, stop the foreground program and remove it from memory.

```
(CTRL/F)
F>
(CTRL/C)
(CTRL/C)
B>
UNLOAD F(RET)
.
```

NOTE

> If your configuration includes a graphics display, turn it off at this point. Type GT OFF.

If you completed these exercises without error, your system has passed this minimal test and you can consider it successfully installed.

4.11 Performing the System Generation Process

If you have decided that you need RT–11 features that are available only if you generate your own monitor(s) and handlers, perform the system generation process at this point. You should have thoroughly studied Chapter 1 to make this decision and to establish that you can perform the system generation process on your particular hardware configuration. Turn to Part III (Chapters 8, 9, and 10) for guidance in planning and performing system generation.

Chapter 5
Installing a System Distributed on Hard Disk to Run on Hard Disk

If RT–11 was distributed to you on hard disk (RL01/RL02 or RK05) and you intend to build a system to run on hard disk, perform the procedures described in this chapter. These procedures assume your configuration includes two disk drives (not necessarily of the same type).

NOTE

> If your hardware configuration includes a VT100 terminal, be sure to set auto XOFF/XON in SETUP mode before attempting to bootstrap RT–11. Never set TT NOPAGE when you use this terminal. Refer to your hardware manuals for more information about these settings.

To install your system, perform the steps summarized in the following list. Sections 5.1 through 5.10 describe the procedures involved in each step. Figure 5–1 shows the various backup disks you create when you install RT–11.

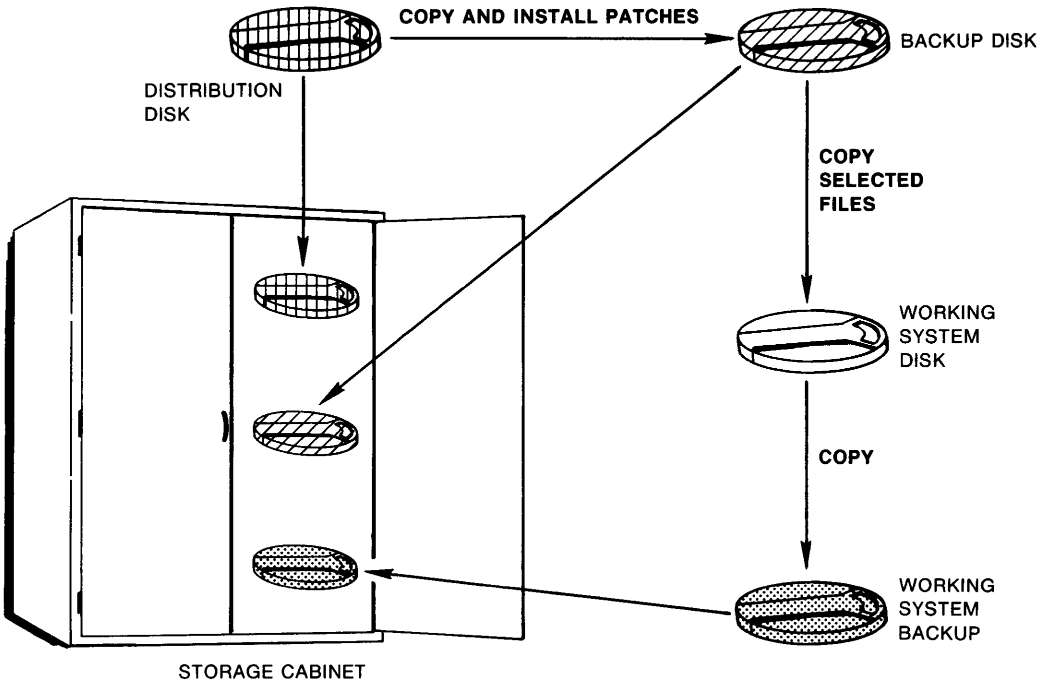
1. Bootstrap the distribution disk.
2. Preserve the distribution disk.
3. Install mandatory patches.
4. Create the system from chosen components.
5. Install the bootstrap on the disk.
6. Customize the system.
7. Compress the disk.
8. Preserve the working system.

9. Test the working system.

10. If appropriate, perform the system generation process.

The following sections correspond to each of these steps and describe in detail the procedures you must perform to complete each step.

Figure 5–1: Sample Backup Disks



5.1 Bootstrapping the Distribution Disk

The first procedure you perform when installing RT–11 is bootstrapping the distribution disk.

Begin by making sure that the processor is powered up but not running. Mount the distribution disk (write protected) in disk Unit 0 (the device name is RK0: or DL0:, depending on the type of disk device). Use the hardware bootstrap to boot the volume. (If your configuration does not include a hardware bootstrap, refer to Appendix E for the toggle-in software bootstrap.)

RT–11 should respond with the following message if you have successfully bootstrapped the volume:

```
RT-11SJ V04.00
(Followed by any startup file commands.)
.
```

Set the date, using the following command (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
.DATE dd-mmm-yy(RET)
.
```

5.2 Preserving the Distribution Disk

The first operation you perform with the running RT–11 system is to copy the distribution disk for backup as a safety measure in case of machine failure or human error. You can then use the backup copy of the distribution to build your working system.

Mount a blank disk in the other disk drive. If the disk is an RK05 disk, format it, then initialize it and cover any bad blocks on it.

If the disk is another type, initialize the disk and replace (using the command INITIALIZE/REPLACE) or cover (using the command INITIALIZE/BADBLOCKS) bad blocks. You have a choice of replacing or covering bad blocks if your disk is RK06, RK07, or RL01 (see the *RT–11 System User's Guide*). If your disk is another type, you should use INITIALIZE/BADBLOCKS to cover bad blocks. If the disk contains bad blocks, the **?DUP-I-Bad block detected nnnnnn** message appears on the terminal, but you can use the disk. In the commands, xxn is the permanent device name and unit number for your disk.

RK05 disk

```
.FORMAT RKn:(RET)
RKn:/FORMAT-Are you sure?Y(RET)
?FORMAT-I-Formatting complete

.INITIALIZE/BADBLOCKS RKn:(RET)
RKn:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected RKn:
```

Other type disk

```
.INITIALIZE/BADBLOCKS xxn:(RET)
xxn:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected xxn:
```

or

```
.INITIALIZE/REPLACE xxn:(RET)
xxn:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected xxn:
```

The system scans the disk for bad blocks and creates a new directory.

```
.
```

The next step in the preservation process is to copy all the files from the distribution disk to the initialized blank disk.

The following command transfers files from the distribution to the backup disk and consolidates all the empty space at the end of the disk. In the command, yyn is the device name and unit number of your distribution disk and xxn is the device and unit of your backup disk.

```
.SQUEEZE/OUTPUT:xxn: yyn:(RET)
.
```

NOTE

> Both the SQUEEZE command with the /OUTPUT:xxn: option and the COPY command transfer files from one device to another. SQUEEZE/OUTPUT:xxn: consolidates free space on the device at the same time. The procedures in this manual use the command that is most appropriate and most efficient for an individual operation. For a better understanding of all RT–11 keyboard monitor commands, refer to the *RT–11 System User's Guide.*

Next, copy the bootstrap to the backup disk.

```
.COPY/BOOT xxn:RT11SJ.SYS xxn:(RET)
.
```

Now halt the processor. Remove the distribution disk and store it. Mount the backup disk and use the hardware bootstrap to boot the backup disk.

```
RT-11xx V04.00
(Followed by any startup file commands.)
.
```

NOTE

> If the backup disk does not boot, repeat the procedures to this point.

Set the date, using the following command (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
.DATE dd-mmm-yy(RET)
.
```

Next, remove the protection from all the files on the backup disk. The files on the distribution disk have been protected to prevent you from accidentally deleting them (see the *RT–11 System User's Guide* for a description of file protection). When you copied the files to the backup disk, RT–11 also copied the protection (note the P that prints next to the file size in the directory). For the rest of these procedures, you need to remove the protection from the backup disk. Use the following command.

```
.RENAME/SYSTEM/NOPROTECTION *.* *.*<RET>
 Files renamed:
DK:aaaaaa.ttt  to DK:aaaaaa.ttt
DK:bbbbbb.ttt  to DK:bbbbbb.ttt
DK:cccccc.ttt  to DK:cccccc.ttt
DK:dddddd.ttt  to DK:dddddd.ttt
            .
            .
            .
DK:zzzzzz.ttt  to DK:zzzzzz.ttt
.
```

5.3 Installing Mandatory Patches

To make sure that RT–11 operates correctly, install mandatory patches at this point. Mandatory patches are critical to system or component operation. They correct software errors discovered since the software was released. Chapter 2 describes how to identify the appropriate patches.

NOTE

> If you are using Autopatch to install mandatory patches, do not read the remainder of this section. Instead, turn to the *RT–11 Autopatch User's Guide* for assistance.

Most published patches are binary patches that you install with the Save Image Patch Program (SIPP). In the event that monitors or handlers require patches, the patches may be published in both binary and source form. If you later perform the system generation process to create your own monitors and handlers, you need source files with any patches applied. Therefore, you must install all source patches, using the Source Language Patch (SLP) utility program.

To install binary patches, invoke SIPP, as indicated, and enter the values specified in the published patch.

```
.R SIPP<RET>
*filnam.typ/C<RET>
Base?       xxxx<RET>
Offset?     xx<RET>
    Base       Offset     Old    New?
  xxxxxx       xxxxxx   xxxxxx   xxxxxx<RET>
  xxxxxx       xxxxxx   xxxxxx   <CTRL/Y> <RET>
Checksum?   xxxxx<RET>
*<CTRL/C>
.
```

The input file, filnam.typ, is the component to be patched. Respond to SIPP's prompts with the values specified in the published patch. The base and offset describe the locations to be patched. The patch may look different from this example if the component is overlaid. CTRL/Y causes SIPP to ask for the checksum. If the checksum matches, SIPP installs the patch and prompts with an asterisk. If you have typed the patch exactly as published, the checksum will match. If the checksum does not match, issue ;V to display modified locations. Correct any errors. Then type CTRL/Y and RETURN.

Refer to the *RT–11 System User's Guide* for more information about SIPP.

To install source patches, create the file, using any editor, exactly as published. Type the command indicated, but insert the correct unit number if the file being patched is not on the system volume. The following is an example of a source patch.

Use a text editor to create the SLP command file, filnam.SLP, as follows.

```
-563
[TAB]DEC[TAB]R5
-660,661
[TAB]BIT[TAB] SPECL$!FILST$,@R2
[TAB]BEQ[TAB]17$
-682
[TAB]MOV[TAB]R3,R0
-700,701
[TAB]BIT[TAB] SPECL$!FILST$,@R2
[TAB]BEQ[TAB]23$
/
```

Then use the SLP utility to patch filnam.MAC as follows:

```
.R SLP[RET]
*filnam.MAC=filnam.MAC,filnam.SLP[RET]
*[CTRL C]
.
```

Use these procedures to make all the mandatory binary and source patches. Patch each component separately, following the sequence specified in the *RT–11 Software Dispatch Review.*

NOTE

> DIGITAL strongly recommends that you analyze all future patches as soon as they are published. Install mandatory patches immediately. Ascertain whether optional patches are useful for your application. Often, you can save much time and effort through such analysis, because these patches may solve a problem you have already experienced.

Once you have created your working system (procedure follows), label and store the patched backup disk. Then, as additional mandatory patches are published in the *RT–11 Software Dispatch*, you can install them on the backup and copy the patched component to the working system. In this way, you will be sure that you have up-to-date, patched versions of all RT–11 components, even if your working system is destroyed.

5.4 Creating the Working System from Chosen Components

Once you have chosen your system components (Section 2.3) and have planned the best arrangement of them on disks (Section 2.4), you can create the working system by copying selected components to another disk or disks.

Mount a blank disk in another disk unit and initialize it. If your disk is an RK05, do not forget to format the disk before you initialize it.

RK05 disk

```
.FORMAT RKn:(RET)
RKn:/FORMAT-Are you sure?Y(RET)
?FORMAT-I-Formatting complete

.INITIALIZE/BADBLOCKS RKn:(RET)
RKn:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected RKn:
```

Other type disk

```
.INITIALIZE/BADBLOCKS xxn:(RET)
xxn:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected xxn:
.
```

or

```
.INITIALIZE/REPLACE xxn:(RET)
xxn:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected xxn:
.
```

Copy the files you have selected from the backup disk to the disk that will become your working system disk. If you use the following command, RT–11 queries you about all the files on the backup disk, and you choose the files it copies. In the command, yyn is the device name and unit number of your backup disk and xxn is the device and unit of the disk to be your working system disk.

```
.COPY/SYSTEM/QUERY yyn: xxn:(RET)
  Files copied:
yyn:aaaaaa.ttt to xxn:aaaaaa.ttt? Y(RET) (to include a specific file)
yyn:bbbbbb.ttt to xxn:bbbbbb.ttt? N(RET) (to exclude a specific file)
(and so on)
```

5.5 Installing the Bootstrap on the Disk

Once you have created your system, you need to install the bootstrap on the working system disk. In the following command, aa is BL, SJ, or FB.

```
.COPY/BOOT xxn:RT11aa.SYS xxn:(RET)
```

In this command, identify the device on which the monitor that contains the bootstrap information resides, the name of that monitor file (RT11BL, RT11SJ, RT11FB), and the device on which you need to install the bootstrap. This command copies bootstrap information from the monitor file to blocks 0 and 2 through 5 of the same disk.

Use the hardware bootstrap to boot the new working system disk (mount the disk in Unit 0 if it is not already there).

```
RT-11xx V04.00
(Followed by any startup file commands.)
.
```

Set the date using the following command (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
.DATE dd-mmm-yy(RET)
.
```

5.6 Customizing the System

You may want to make certain customizations (described in Section 2.8) to the distributed RT–11 components. At this point, perform the procedures to implement any of these customizations. Table 1–4 summarizes the available customizations and directs you to the section in Chapter 2 that describes a particular customization and the procedure for implementing it.

NOTE

You can perform the system generation process to implement additional customizations. See Section 5.10.

5.7 Compressing the Disk

DIGITAL recommends that you compress the working system disk to make its free space contiguous. Consolidating free space allows you to use space on the disk that would otherwise be too fragmented to be usable.

Continue to run RT–11 from Unit 0 and use the SQUEEZE command to compress free space. (The volume must be write enabled.) The squeeze operation does not move files with the .BAD file type.

```
.SQUEEZE xxn:(RET)
xxn:/Squeeze; Are you sure? Y(RET)
RT-11xx V04.00x
(Followed by any startup file commands.)
.
```

The system automatically reboots when you compress a system volume.

5.8 Preserving the Working System

Once you build a satisfactory working system, DIGITAL recommends that you preserve it on the backup medium of your choice. The following sections describe backing up on disk. If you want to back up the system on another device, the procedure may appear elsewhere in this manual; refer to the table of contents.

Insert a blank disk (write enabled) in another disk unit and initialize the blank disk. In the following commands, xxn is the device name and unit number for the working system backup disk.

RK05 disk

```
.FORMAT RKn:(RET)
RKn:/FORMAT-Are you sure?Y(RET)
?FORMAT-I-Formatting complete

.INITIALIZE/BADBLOCKS RKn:(RET)
RKn:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected RKn:
```

Other type disk

```
.INITIALIZE/BADBLOCKS xxn:(RET)
xxn:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected xxn:
.
```

or

```
.INITIALIZE/REPLACE xxn:(RET)
xxn:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected xxn:
.
```

Copy all the files in your working system. In the following command, yyn is the device name and unit number for your working system disk and xxn is the device and unit for the backup working system disk.

```
.SQUEEZE/OUTPUT:xxn yyn:(RET)
.
```

Copy the bootstrap to the backup working system disk. In the following command, aa is BL, SJ, or FB.

```
.COPY/BOOT xxn:RT11aa.SYS xxn:(RET)
.
```

Store the backup disk. If you ever need to restore the working system, you can make a copy of the backup.

5.9 Testing the Working System

Once you have built and preserved the working system, you can execute the following demonstration to test the system. This demonstration does not serve as a comprehensive system exercise; however, because it uses several major system components, it does serve as a minimal integrity check. Moreover, DIGITAL considers your system officially installed if the demonstration runs without error.

To execute this demonstration, you should have at least 20 free blocks on your working system device and the working system must include at least the following components. To perform the foreground/background monitor exercise, you need a terminal with a bell.

SWAP.SYS
RT11SJ.SYS
RT11FB.SYS (if you do the foreground/background exercise)
xx.SYS (system device handler)
TT.SYS (for SJ monitor)
LP.SYS (if appropriate)
EDIT.SAV
MACRO.SAV
SYSMAC.SML
LINK.SAV
PIP.SAV
DUP.SAV
DIR.SAV
DEMOBG.MAC
DEMOFG.MAC

You can do only the single-job monitor exercise to test the single-job monitor or you can do both exercises to test both the single-job monitor and the foreground/background monitor.

5.9.1 Single-Job Monitor Exercise

For the single-job monitor exercise, you edit, assemble, link, and execute a small program (DEMOBG.MAC). When you execute DEMOBG, it displays a message on the terminal.

5.9.1.1 Bootstrapping the SJ Monitor and Getting Started — Write enable the system disk, and then bootstrap the system.

```
RT-11SJ V04.00x
(Followed by any startup file commands.)
.
```

Set the date, using the following command (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
.DATE dd-mmm-yy(RET)
.
```

NOTE

> If your configuration includes a VT11 or VS60 display processor and scope, shift system output to the display scope. Type GT ON. Verify that the scope is on by turning the BRIGHTNESS knob to an adequate level. You can still enter commands at the keyboard, but the echo is on the screen.

Display the directory of the system disk on the terminal. The directory varies according to your particular working system. As long as a directory prints, you need not worry if it does not match the one in the following example.

```
.DIRECTORY/BRIEF/COLUMNS:1 SY:<RET>
  dd-mmm-yy
SWAP  .SYS
RT11SJ.SYS
LP    .SYS
RK    .SYS
TT    .SYS
SIPP  .SAV
EDIT  .SAV
MACRO .SAV
SYSMAC.SML
LINK  .SAV
ODT   .OBJ
PIP   .SAV
    .
    .
    .
xxx Files, bbb Blocks
fff Free blocks
.
```

NOTE

> If you have shifted output to a scope and the directory scrolls by too quickly to read, type CTRL/S to stop the display and CTRL/Q to restart it.

5.9.1.2 Editing the Demonstration Program — Next, use the text editor to modify the demonstration program, DEMOBG.MAC. If DEMOBG.MAC is a protected file, remove the protection before making the edits (RENAME/NOPROTECTION SY:DEMOBG.MAC SY:DEMOBG.MAC).

```
.EDIT SY:DEMOBG.MAC<RET>
*F;<TAB>.ASCII<ESC><ESC>
*0AD<ESC><ESC>
*EX<ESC><ESC>
.
```

5.9.1.3 Assembling the Demonstration Program — To assemble DEMOBG.MAC and obtain a listing, make sure that your configuration has a line printer that is on-line and ready.

```
.ASSIGN LP: LST:<RET>
.
```

NOTE

If your configuration does not include a line printer, use the console terminal.

```
.ASSIGN TT: LST:(RET)
.
```

Assemble DEMOBG.MAC as follows:

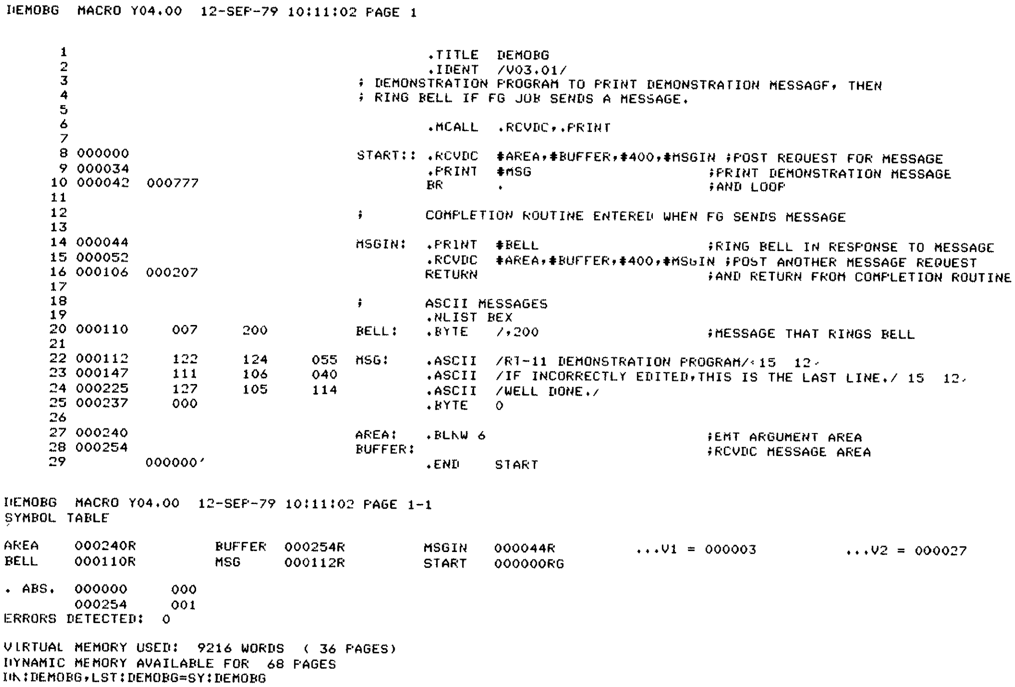
```
.MACRO/LIST:LST: SY:DEMOBG(RET)
```

(see Figure 5–2)

If any errors occur when you assemble DEMOBG.MAC, you have incorrectly edited the file and should repeat the edits. Use the backup demonstration program.

```
.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC(RET)
.
```

Figure 5–2: DEMOBG Assembly Listing



```
DEMOBG  MACRO Y04.00  12-SEP-79 10:11:02 PAGE 1

      1                                         .TITLE  DEMOBG
      2                                         .IDENT  /V03.01/
      3                                 ; DEMONSTRATION PROGRAM TO PRINT DEMONSTRATION MESSAGF, THEN
      4                                 ; RING BELL IF FG JOB SENDS A MESSAGE.
      5
      6                                         .MCALL  .RCVDC,.PRINT
      7
      8 000000                          START:: .RCVDC  #AREA,#BUFFER,#400,#MSGIN ;POST REQUEST FOR MESSAGE
      9 000034                                  .PRINT  #MSG                      ;PRINT DEMONSTRATION MESSAGE
     10 000042  000777                          BR      .                         ;AND LOOP
     11
     12                                 ;       COMPLETION ROUTINE ENTERED WHEN FG SENDS MESSAGE
     13
     14 000044                          MSGIN:  .PRINT  #BELL                     ;RING BELL IN RESPONSE TO MESSAGE
     15 000052                                  .RCVDC  #AREA,#BUFFER,#400,#MSGIN ;POST ANOTHER MESSAGE REQUEST
     16 000106  000207                          RETURN                            ;AND RETURN FROM COMPLETION ROUTINE
     17
     18                                 ;       ASCII MESSAGES
     19                                         .NLIST BEX
     20 000110     007     200          BELL:   .BYTE   /,200                     ;MESSAGE THAT RINGS BELL
     21
     22 000112     122     124     055  MSG:    .ASCII  /RT-11 DEMONSTRATION PROGRAM/<15  12>
     23 000147     111     106     040          .ASCII  /IF INCORRECTLY EDITED,THIS IS THE LAST LINE./ 15  12>
     24 000225     127     105     114          .ASCII  /WELL DONE./
     25 000237     000                          .BYTE   0
     26
     27 000240                          AREA:   .BLKW 6                           ;EMT ARGUMENT AREA
     28 000254                          BUFFER:                                   ;RCVDC MESSAGE AREA
     29         000000'                         .END    START

DEMOBG  MACRO Y04.00  12-SEP-79 10:11:02 PAGE 1-1
SYMBOL TABLE

AREA    000240R        BUFFER  000254R        MSGIN   000044R        ...V1 = 000003        ...V2 = 000027
BELL    000110R        MSG     000112R        START   000000RG

. ABS.  000000     000
        000254     001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  9216 WORDS  ( 36 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  68 PAGES
DK:DEMOBG,LST:DEMOBG=SY:DEMOBG
```

5.9.1.4 Linking the Demonstration Program — Now, link the program DEMOBG.

```
.LINK DEMOBG(RET)
.
```

5.9.1.5 Running the Demonstration Program — Run the program DEMOBG to check the results of the first exercise.

```
.RUN DEMOBG(RET)
RT-11 DEMONSTRATION PROGRAM
IF INCORRECTLY EDITED, THIS IS THE LAST LINE.
WELL DONE.
(CTRL/C)
(CTRL/C)
.
```

NOTE

> If your configuration includes a graphics display, turn it off at this point. Type GT OFF.

If you incorrectly edited the file, you can repeat this exercise, although you can continue without correcting the file. If, however, you want to repeat the exercise, begin by using the backup demonstration program.

```
.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC(RET)
.
```

Then, repeat the editing procedure.

5.9.2 Foreground/Background Monitor Exercise

For the foreground/background monitor exercise, assemble a second program (DEMOFG.MAC), link it for the foreground, and execute it in conjunction with DEMOBG (procedures follow). DEMOFG is a small foreground program that sends a message every two seconds to DEMOBG (running in the background), telling it to ring the terminal bell. Besides printing the terminal message used in the single-job exercise, DEMOBG recognizes these messages and rings the bell once for each message sent.

Although DEMOFG is always active, sending messages to the background every two seconds, this exercise can execute other programs in the background besides DEMOBG. The circuit is complete and messages are successfully received and honored only when DEMOBG is active. During those periods when DEMOBG is not running, DEMOFG enters the messages in the monitor message queue. Once you restart DEMOBG in the background, the system immediately releases all the messages queued since the last forced exit, resulting in many successive bell rings. When the queue is empty, the normal send/receive cycle resumes and the bell rings every two seconds as each current message is sent and honored.

5.9.2.1 Bootstrapping the FB Monitor and Getting Started — Make sure that the RT–11 single-job monitor is running. Then boot the foreground/background monitor.

```
.BOOT RT11FB.SYS(RET)
RT-11FB V04.00x
(Followed by any startup file commands.)
.
```

NOTE

> If your configuration includes a VT11 or VS60 display processor and scope, shift system output to the display scope. Type GT ON. Verify that the scope is on by turning the BRIGHTNESS knob to an adequate level. You can still enter commands at the keyboard, but the echo is on the screen.

Now, enter the time of day using the following command. In the command, hh:mm:ss is the hour, minutes, and seconds in the form 13:12:00 (1:12 p.m.).

```
.TIME hh:mm:ss(RET)
.
```

5.9.2.2 Assembling the Demonstration Program — Assemble the foreground demonstration program, DEMOFG.MAC.

```
.MACRO SY:DEMOFG(RET)
ERRORS DETECTED: 0
.
```

5.9.2.3 Linking the Demonstration Program — Link DEMOFG for the foreground.

```
.LINK/FOREGROUND DEMOFG(RET)
.
```

5.9.2.4 Running the Demonstration Programs — Start DEMOFG as the foreground job.

```
.FRUN DEMOFG(RET)
F>
FOREGROUND DEMONSTRATION PROGRAM
SENDS A MESSAGE TO THE BACKGROUND PROGRAM DEMOBG
EVERY 2 SECONDS, TELLING IT TO RING THE BELL.
B>
```

DEMOFG is running and queuing the message for DEMOBG every two seconds. Now execute DEMOBG in the background and receive the messages.

```
RUN DEMOBG(RET)
```

(The bell rings quickly several times, then once every two seconds.)

```
RT-11 DEMONSTRATION PROGRAM
IF INCORRECTLY EDITED, THIS IS THE LAST LINE.
WELL DONE.
```

Execute a DIRECTORY command in the background to obtain a directory listing.

```
CTRL/C
CTRL/C
```

(The bell stops.)

```
.DIRECTORYRET
 dd-mmm-yy
```

(The directory of the device DK prints on the terminal.)

```
.
```

Rerun DEMOBG to collect all the foreground messages queued while the directory was printing.

```
.RUN DEMOBGRET
```

(The bell rings several times in rapid succession, then rings once every two seconds.)

```
RT-11 DEMONSTRATION PROGRAM
IF INCORRECTLY EDITED, THIS IS THE LAST LINE.
WELL DONE.
CTRL/C
CTRL/C
.
```

(The bell stops.)

Now, stop the foreground program and remove it from memory.

```
CTRL/F
F>
CTRL/C
CTRL/C
B>
UNLOAD FRET
.
```

NOTE

> If your configuration includes a graphics display, turn it off at this point. Type GT OFF.

If you completed these exercises without error, your system has passed this minimal test and you can consider it successfully installed.

5.10 Performing the System Generation Process

If you have decided that you need RT–11 features that are available only if you generate your own monitor(s) and handlers, perform the system generation process at this point. You should have thoroughly studied Chapter 1 to make this decision and to establish that you can perform the system generation process on your particular hardware configuration. Turn to Part III (Chapters 8, 9, and 10) for guidance in planning and performing system generation.

Chapter 6
Installing a System Distributed on RX02 to Run on RX02

If RT–11 was distributed to you on RX02 double-density diskette and you intend to build a system to run on double-density diskette, perform the procedures described in this chapter.

NOTE

> If your hardware configuration includes a VT100 terminal, be sure to set auto XOFF/XON in SETUP mode before attempting to bootstrap RT–11. Never set TT NOPAGE when you use this terminal. Refer to your hardware manuals for more information about these settings.

To install your system, perform the steps summarized in the following list. Sections 6.1 through 6.10 describe the procedures involved in each step. Figure 6–1 shows the various backup diskettes you create when you install RT–11.

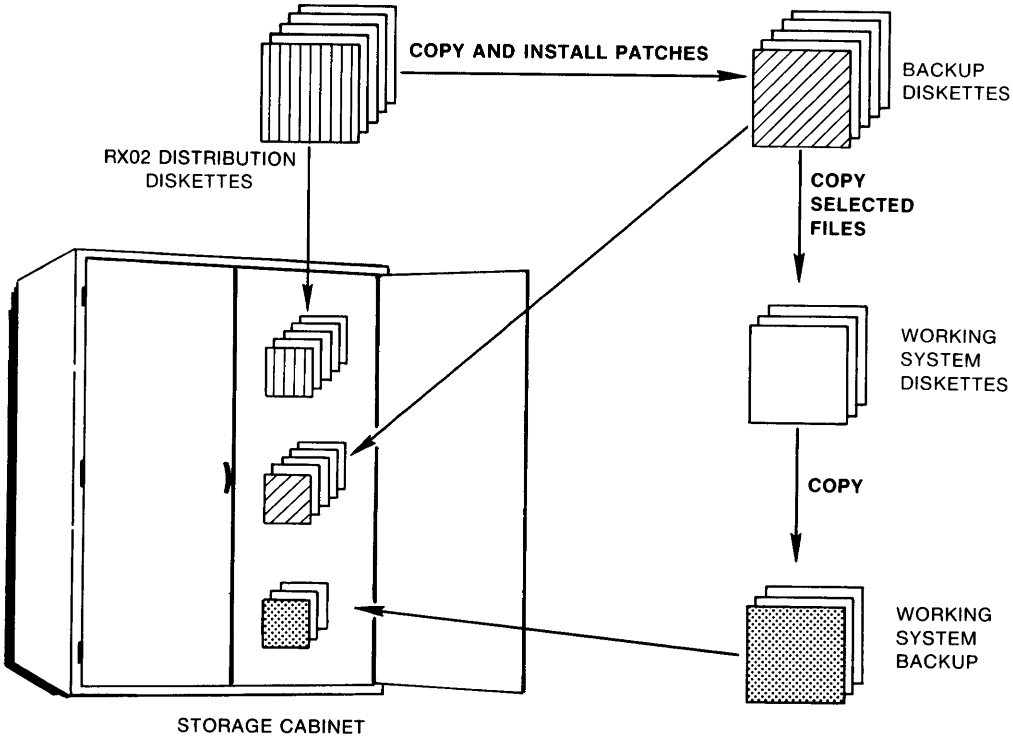
1. Bootstrap the distribution diskette.
2. Preserve the distribution diskette.
3. Install mandatory patches.
4. Create the system from chosen components.
5. Install the bootstrap on diskettes that need to be bootable.
6. Customize the system.
7. Compress free space on each diskette.
8. Preserve the working system.

9. Test the working system.

10. If appropriate, perform the system generation process.

The following sections correspond to each of these steps and describe in detail the procedures you must perform to complete each step.

Figure 6–1: Sample Backup Diskettes



6.1 Bootstrapping the Distribution Diskette

The first procedure you perform when installing RT–11 is bootstrapping the distribution diskette.

Begin by making sure that the processor is powered up but not running. Insert the distribution diskette labeled 1/4 in Unit 0 (which has the physical device name DY0:). Use the hardware bootstrap to boot the diskette. (If your configuration does not include a hardware bootstrap, see Appendix E for toggle-in software bootstraps.)

RT–11 should respond with the following message if you have successfully bootstrapped the volume:

```
RT-11SJ V04.00
(Followed by any startup file commands.)
.
```

Use the DATE command to set the date (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
.DATE dd-mmm-yy(RET)
.
```

6.2 Preserving the Distribution Diskettes

The first operation you perform with the running RT–11 system is to copy all the distribution diskettes for backup as a safety measure in case of machine failure or human error.

Insert a blank diskette in Unit 1 so you can format and initialize it. Diskettes are available in single-density but not double-density format. Therefore, you must reformat all your blank diskettes as double-density diskettes. You can use the FORMAT utility program, which is included on distribution diskette number 1, to format the blank diskette and the INITIALIZE command to initialize it. Use the BADBLOCKS option with INITIALIZE to cover any bad blocks that may be on your diskette. If the diskette contains bad blocks, the **?DUP-I-Bad block detected nnnnnn** message appears on the terminal.

NOTE

> DIGITAL recommends that you use only diskettes that do not have bad blocks when you back up distribution diskettes and build a working system. To ascertain whether an already initialized diskette has bad blocks, use the command DIRECTORY/BAD DYn:. You can use diskettes with bad blocks later, for working or data volumes.

```
.FORMAT DY1:(RET)
DY1:/FORMAT-Are you sure?Y(RET)
?FORMAT-I-Formatting complete

.INITIALIZE/BADBLOCKS DY1:(RET)
DY1:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected DY1:
```

The system scans the diskette for bad blocks and creates a new directory. The monitor dot appears when this process is complete.

```
.
```

Now remove the newly formatted and initialized diskette. Format and initialize three more blank diskettes, leaving one write enabled, initialized, blank diskette inserted in Unit 1.

The next step in the preservation process is to copy all the files from distribution diskette number 1 to an initialized blank diskette, which becomes backup diskette number 1.

As long as the diskette you intend to copy is bootable (that is, it has the appropriate monitor file, the system device handler, and a bootstrap) and contains certain system utility programs, you can boot RT–11 from that diskette and copy the diskette. In this case, the SQUEEZE command, with the OUTPUT option transfers all the files from the diskette in Unit 0 to the one in Unit 1 and consolidates all the empty space at the end of the diskette. Distribution diskette number 1 is bootable, but the rest of the diskettes in your kit are not bootable because they lack the necessary components. Thus, these diskettes require a different copy procedure. For diskette number 1, however, the procedure is straightforward.

```
.SQUEEZE/OUTPUT:DY1: DY0:(RET)
.
```

NOTE

> Both the SQUEEZE command with the /OUTPUT:xxn: option and the COPY command transfer files from one device to another. SQUEEZE/OUTPUT:xxn: consolidates free space on the device at the same time. The procedures in this manual use the command that is most appropriate and most efficient for an individual operation. For a better understanding of all RT–11 keyboard monitor commands, refer to the *RT–11 System User's Guide.*

Then copy the bootstrap to the diskette.

```
.COPY/BOOT DY1:RT11SJ.SYS DY1:(RET)
.
```

Remove the newly created backup diskette from Unit 1 and label it "backup RT–11 V04 1/4." (Use a soft-tipped pen when you label diskettes.)

Because the remaining distribution diskettes lack necessary system components, you cannot boot them. Therefore, to copy the remaining diskettes, run RT–11 from distribution diskette number 1 in Unit 0.

Type:

```
.SQUEEZE/WAIT/OUTPUT:DY1: DY0:(RET)
Mount output volume in DY1:; Continue?
```

Insert a formatted and initialized blank diskette in Unit 1.

```
Y(RET)

Mount input volume in DY0:; Continue?
```

Replace the diskette in Unit 0 with distribution diskette number 2.

```
Y(RET)

Mount system volume in DY0:; Continue?
```

Replace the diskette in Unit 0 with distribution diskette number 1.

```
Y(RET)
.
```

Remove the diskette from Unit 1, label it "backup RT–11 V04 2/4."

Type the command:

```
.SQUEEZE/WAIT/OUTPUT:DY1: DY0:(RET)
Mount output volume in DY1:; Continue?
```

Insert another initialized blank diskette in Unit 1.

```
Y(RET)
Mount input volume in DY0:; Continue?
```

Replace the diskette in Unit 0 with distribution diskette number 3.

```
Y(RET)
Mount system volume in DY0:; Continue?
```

Replace the diskette in Unit 0 with distribution diskette number 1.

```
Y(RET)
.
```

Remove the diskette from Unit 1, label it "backup RT–11 V04 3/4," and insert another initialized blank diskette in Unit 1. Repeat these procedures to copy the fourth distribution diskette.

Now halt the processor, and replace distribution diskette number 1 with backup number 1 in Unit 0. Store the distribution diskettes and use the backup copies to build a working system.

Use the hardware bootstrap to boot backup number 1.

```
RT-11SJ V04.00
(Followed by any startup file commands.)
.
```

NOTE

> If the backup diskette does not boot, repeat the procedure to create the backup diskettes.

Set the date, using the following command (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
.DATE dd-mmm-yy(RET)
.
```

Next, remove the protection from all the files on the backup diskettes. The files on the distribution diskettes have been protected to prevent you from accidentally deleting them (see the *RT-11 System User's Guide* for a description of file protection). When you copied the files to the backup diskettes, RT-11 also copied the protection (note the P that prints next to the file size in the directory). For the rest of these procedures, you need to remove the protection from the backup diskettes. Type the following command to remove it from the files on backup diskette number 1.

```
.RENAME/SYSTEM/NOPROTECTION *.* *.*(RET)
 Files renamed:
DK:aaaaaa.ttt  to DK:aaaaaa.ttt
DK:bbbbbb.ttt  to DK:bbbbbb.ttt
DK:cccccc.ttt  to DK:cccccc.ttt
DK:dddddd.ttt  to DK:dddddd.ttt
              .
              .
              .
DK:zzzzzz.ttt  to DK:zzzzzz.ttt
.
```

Insert backup diskette 2 in Unit 1 and type the following command:

```
.RENAME/NOPROTECTION DY1:*.* DY1:*.*(RET)
 Files renamed:
DY1:aaaaaa.ttt  to DY1:aaaaaa.ttt
DY1:bbbbbb.ttt  to DY1:bbbbbb.ttt
DY1:cccccc.ttt  to DY1:cccccc.ttt
DY1:dddddd.ttt  to DY1:dddddd.ttt
              .
              .
              .
DY1:zzzzzz.ttt  to DY1:zzzzzz.ttt
.
```

Replace backup diskette 2 with backup diskette 3 in Unit 1 and repeat this command. In the same way, remove protection from the files on the fourth backup diskette.

6.3 Installing Mandatory Patches

To make sure that RT-11 operates correctly, install mandatory patches at this point. Mandatory patches are critical to system or component operation. They correct software errors discovered since the software was released. Chapter 2 describes how to identify the appropriate patches.

NOTE

> If you are using Autopatch to install mandatory patches, do not read the remainder of this section. Instead, turn to the *RT-11 Autopatch User's Guide* for assistance.

Most published patches are binary patches that you install with the Save Image Patch Program (SIPP). In the event that monitors or handlers require patches, the patches may be published in both binary and source form. If you later perform the system generation process to create your own monitors and handlers, you need source files with any patches applied. Therefore, you must install all source patches, using the Source Language Patch (SLP) utility program.

To install binary patches, invoke SIPP, as indicated, and enter the values specified in the published patch.

```
.R SIPP(RET)
*filnam.typ/C(RET)
Base?        xxxx(RET)
Offset?      xx(RET)
      Base         Offset      Old      New?
    xxxxxx         xxxxxx    xxxxxx     xxxxxx(RET)
    xxxxxx         xxxxxx    xxxxxx     (CTRL/Y)(RET)
Checksum?    xxxxx(RET)
*(CTRL/C)
.
```

The input file, filnam.typ, is the component to be patched. If the input file is not on your system diskette, identify the device unit on which it resides (xxn:filnam.typ). Respond to SIPP's prompts with the values specified in the published patch. The base and offset describe the locations to be patched. The patch may look different from this example if the component is overlaid. CTRL/Y causes SIPP to ask for the checksum. If the checksum matches, SIPP installs the patch and prompts with an asterisk. If you have typed the patch exactly as published, the checksum will match. If the checksum does not match, issue ;V to display modified locations. Correct any errors. Then type CTRL/Y and RETURN. To patch components that are not on backup diskette number 1, insert in Unit 1 the diskette that contains the component to be patched. Then specify that unit in the command string to SIPP:

```
*DY1:filnam.typ(RET)
```

Refer to the *RT–11 System User's Guide* for more information about SIPP.

To install source patches, create the file, using any editor, exactly as published. Type the command indicated, but insert the correct unit number if the file being patched is not on the system volume. The following is an example of a source patch.

Use a text editor to create the SLP command file, filnam.SLP, as follows.

```
-563
(TAB)DEC(TAB)R5
-660,661
(TAB)BIT(TAB) SPECL$!FILST$,@R2
(TAB)BEQ(TAB)17$
-682
(TAB)MOV(TAB)R3,R0
-700,701
(TAB)BIT(TAB) SPECL$!FILST$,@R2
(TAB)BEQ(TAB)23$
/
```

Then use the SLP utility to patch filnam.MAC as follows:

```
.R SLP(RET)
*filnam.MAC=filnam.MAC,filnam.SLP(RET)
*(CTRL/C)
.
```

Use these procedures to make all the mandatory binary and source patches. Patch each component separately, following the sequence specified in the *RT–11 Software Dispatch Review.*

NOTE

> DIGITAL strongly recommends that you analyze all future patches as soon as they are published. Install mandatory patches immediately. Ascertain whether optional patches are useful for your application. Often, you can save much time and effort through such analysis, because these patches may solve a problem you have already experienced.

Once you have created your working system (procedure follows), store the patched backup diskettes. Then, as additional mandatory patches are published in the *RT–11 Software Dispatch*, you can install them on the backup and copy the patched component to the working system. In this way, you will be sure that you have up-to-date, patched versions of all RT–11 components, even if your working system is destroyed.

6.4 Creating the Working System from Chosen Components

Once you have chosen your system components (Section 2.3) and have planned the best arrangement of them on diskettes (Section 2.4), you can create the working system. You create the working system by copying selected components to formatted and initialized blank diskettes.

Start by formatting and initializing a number of blank diskettes, following the same procedure that you used in Section 6.2. Repeat the process to create as many initialized blank diskettes as you need for the system that you have planned.

Then, use the COPY command with the SYSTEM option to copy selected files from backup diskette 1 to the diskette that becomes your working system diskette.

```
.COPY/SYSTEM DY0:filnam.typ DY1:filnam.typ(RET)
.
```

Or you can use the following command to avoid typing numerous file specifications. RT–11 queries you about all the files on the diskette, and you choose the files it copies.

```
.COPY/SYSTEM/QUERY DY0: DY1:(RET)
  Files copied:
DY0:aaaaaa.ttt to DY1:aaaaaa.ttt? Y(RET) (to include a specific file)
DY0:bbbbbb.ttt to DY1:bbbbbb.ttt? N(RET) (to exclude a specific file)
(and so on)
```

To copy files from nonbootable diskettes, you have to alternate diskettes.

Use the SET command to set the USR to NOSWAP.

```
.SET USR NOSWAP(RET)
.
```

Type the following command, where filnam.typ is the name of the file you want to copy. In this case, you cannot use the QUERY option; you must specify individual files.

```
.COPY/WAIT DY1:filnam.typ DY0:filnam.typ(RET)

Mount input volume in DY1:; Continue?
```

Place the diskette containing the file you want to copy in Unit 1.

```
Y(RET)

Mount output volume in DY0:; Continue?
```

Replace the system diskette in Unit 0 with the diskette to which you want to copy filnam.typ.

```
Y(RET)

Mount system volume in DY0:; Continue?
```

Replace the diskette in Unit 0 with backup diskette number 1.

```
Y(RET)
.
```

Repeat this procedure to copy all the files you planned for the working system diskette. When you have copied all the files, label the diskette "RT–11 working system V04 1/x" (where x is the number of diskettes in your working system). Repeat these procedures to create the other diskettes in the working system.

When you have created and labeled all the working system diskettes, you can permit the USR to swap again.

```
.SET USR SWAP(RET)
.
```

6.5 Installing the Bootstrap on Any Diskettes That Need to Be Bootable

Once you have created your system, you need to install the bootstrap on any diskettes that must be bootable (that is, that you can use as the system diskette). Generally, any diskette that includes a monitor file and system device handler should be bootable (but do not forget that the diskette would need SWAP.SYS and, for the SJ monitor, TT.SYS).

Insert in Unit 1 the diskette on which you need to install the bootstrap. In the command, yy is BL, SJ, or FB.

```
.COPY/BOOT DY1:RT11yy.SYS DY1:(RET)
.
```

In this command, identify the device on which the monitor that contains the bootstrap information resides, the name of that monitor file, and the device on which you need to install the bootstrap. This command copies bootstrap information from the monitor file to blocks 0 and 2 through 5 of the same diskette.

Then, insert working system diskette number 1 in Unit 0, and use the hardware bootstrap to boot your working system.

```
RT-11xx V04.00x
(Followed by any startup file commands.)
.
```

Set the date, using the following command (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
.DATE dd-mmm-yy(RET)
.
```

Store the patched backup diskettes for future patching purposes.

6.6 Customizing the System

You may want to make certain customizations (described in Section 2.8) to the distributed RT–11 components. At this point, perform the procedures to implement any of these customizations. Table 1–4 summarizes the available customizations and directs you to the section in Chapter 2 that describes a particular customization and the procedure for implementing it.

NOTE

Some RT–11 configurations can perform the system generation process to implement additional customizations. See Section 6.10.

6.7 Compressing Each Diskette

DIGITAL recommends that you compress each working system diskette to make its free space contiguous. Consolidating free space allows you to use space on the diskette that would otherwise be too fragmented to be usable.

Continue to run RT–11 from Unit 0 and use the SQUEEZE command to compress free space. The squeeze operation does not move files with the .BAD file type.

```
.SQUEEZE DY0:(RET)
DY0:/Squeeze; Are you sure? Y(RET)
```

There is a delay while the squeeze operation takes place.

```
RT-11xx V04.00x
(Followed by any startup file commands.)
.
```

The system automatically reboots when you compress a system diskette.

Then insert the next diskette that you need to compress in Unit 1.

```
.SQUEEZE DY1:(RET)
DY1:/Squeeze; Are you sure? Y(RET)
.
```

Replace the diskette in Unit 1 with the next one you need to compress, and repeat this procedure for all the diskettes.

NOTE

> When you compress a diskette with system files (.SYS), PIP warns you to reboot. Do reboot as advised. When you compress a system diskette, the system automatically reboots.

6.8 Preserving the Working System

Once you build a satisfactory working system, DIGITAL recommends that you protect all the files and preserve the system on backup diskettes.

Use the following command to protect all the files on the system diskette.

```
.RENAME/SYSTEM/PROTECTION *.* *.*(RET)
 Files renamed:
DK:aaaaaa.ttt   to DK:aaaaaa.ttt
DK:bbbbbb.ttt   to DK:bbbbbb.ttt
DK:cccccc.ttt   to DK:cccccc.ttt
DK:dddddd.ttt   to DK:dddddd.ttt
              .
              .
              .
DK:zzzzzz.ttt   to DK:zzzzzz.ttt
.
```

To protect files on other diskettes in the working system, insert the diskette in Unit 1 and use the following command:

```
.RENAME/SYSTEM/PROTECTION DY1:*.* DY1:*.*(RET)
 Files renamed:
DY1:aaaaaa.ttt   to DY1:aaaaaa.ttt
DY1:bbbbbb.ttt   to DY1:bbbbbb.ttt
DY1:cccccc.ttt   to DY1:cccccc.ttt
DY1:dddddd.ttt   to DY1:dddddd.ttt
              .
              .
              .
DY1:zzzzzz.ttt   to DY1:zzzzzz.ttt
.
```

Next, copy the working system to backup diskettes. Insert a blank diskette in Unit 1 with RT–11 still booted from Unit 0. Format and initialize the appropriate number of diskettes.

Copy all the files in your working system. You can use the SQUEEZE/OUTPUT:DY1: DY0: command to copy any bootable diskettes. Remember that you must use SQUEEZE/WAIT/OUTPUT:DY1: DY0: and change diskettes to copy the diskettes that are not bootable (see Section 6.2). Also remember to copy the bootstrap to any diskettes that need to be bootable.

Store the backup diskettes. If you ever need to restore the working system, you can make copies of the backup.

6.9 Testing the Working System

Once you have built and preserved the working system, you can execute the following demonstration to test the system. This demonstration does not serve as a comprehensive system exercise; however, because it uses several major system components, it does serve as a minimal integrity check. Moreover, DIGITAL considers your system officially installed if the demonstration runs without error.

To execute this demonstration, your working system must include at least the following components. To perform the foreground/background monitor exercise, you need a terminal with a bell.

SWAP.SYS
RT11SJ.SYS
RT11FB.SYS (if you do the foreground/background exercise)
DY.SYS (system device handler)
TT.SYS (for SJ monitor)

LP.SYS (if appropriate)
EDIT.SAV
MACRO.SAV
SYSMAC.SML
LINK.SAV
PIP.SAV
DUP.SAV
DIR.SAV
FORMAT.SAV
DEMOBG.MAC
DEMOFG.MAC

You can do only the single-job monitor exercise to test the single-job monitor or you can do both exercises to test both the single-job monitor and the foreground/background monitor.

6.9.1 Single-Job Monitor Exercise

For the single-job monitor exercise, you edit, assemble, link, and execute a small program (DEMOBG.MAC). When you execute DEMOBG, it displays a message on the terminal.

6.9.1.1 Bootstrapping the SJ Monitor and Getting Started — Bootstrap the system and set the date, using the following command (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
RT-11SJ V04.00x
(Followed by any startup file commands.)
.DATE dd-mmm-yy(RET)
.
```

NOTE

> If your configuration includes a VT11 or VS60 display processor and scope, shift system output to the display scope. Type GT ON. Verify that the scope is on by turning the BRIGHTNESS knob to an adequate level. You can still enter commands at the keyboard, but the echo is on the screen.

Insert a blank diskette in Unit 1.

```
.FORMAT DY1:(RET)
DY1:/FORMAT-Are you sure? Y(RET)
?FORMAT-I-Formatting complete

.INITIALIZE/BADBLOCKS DY1:(RET)
DY1:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected DY1:

.ASSIGN DY1: DK:(RET)
```

Display the directory of the system diskette on the terminal. The directory varies according to your particular working system. As long as a directory prints, you need not worry if it does not match the one in the following example.

```
.DIRECTORY/BRIEF/COLUMNS:1 SY:(RET)
  dd-mmm-yy
SWAP  .SYS
RT11SJ.SYS
LP    .SYS
TT    .SYS
DY    .SYS
SIPP  .SAV
EDIT  .SAV
MACRO .SAV
SYSMAC.SML
LINK  .SAV
ODT   .OBJ
PIP   .SAV
     .
     .
     .
xxx Files, bbb Blocks
fff Free blocks
.
```

NOTE

> If you have shifted output to a scope and the directory scrolls by too quickly to read, type CTRL/S to stop the display and CTRL/Q to restart it.

6.9.1.2 Editing the Demonstration Program — Use the text editor to modify the demonstration program, DEMOBG.MAC. If DEMOBG.MAC is a protected file, remove the protection before making the edits (RENAME/NOPROTECTION SY:DEMOBG.MAC SY:DEMOBG.MAC).

```
.EDIT SY:DEMOBG.MAC(RET)
*F;(TAB).ASCII(ESC)(ESC)
*OAD(ESC)(ESC)
*EX(ESC)(ESC)
.
```

6.9.1.3 Assembling the Demonstration Program — To assemble DEMOBG.MAC and obtain a listing, make sure that your configuration has a line printer that is on-line and ready.

```
.ASSIGN LP: LST:(RET)
.
```

NOTE

> If your configuration does not include a line printer, use the console terminal.
>
> ```
> .ASSIGN TT: LST:(RET)
> .
> ```

Assemble DEMOBG.MAC as follows:

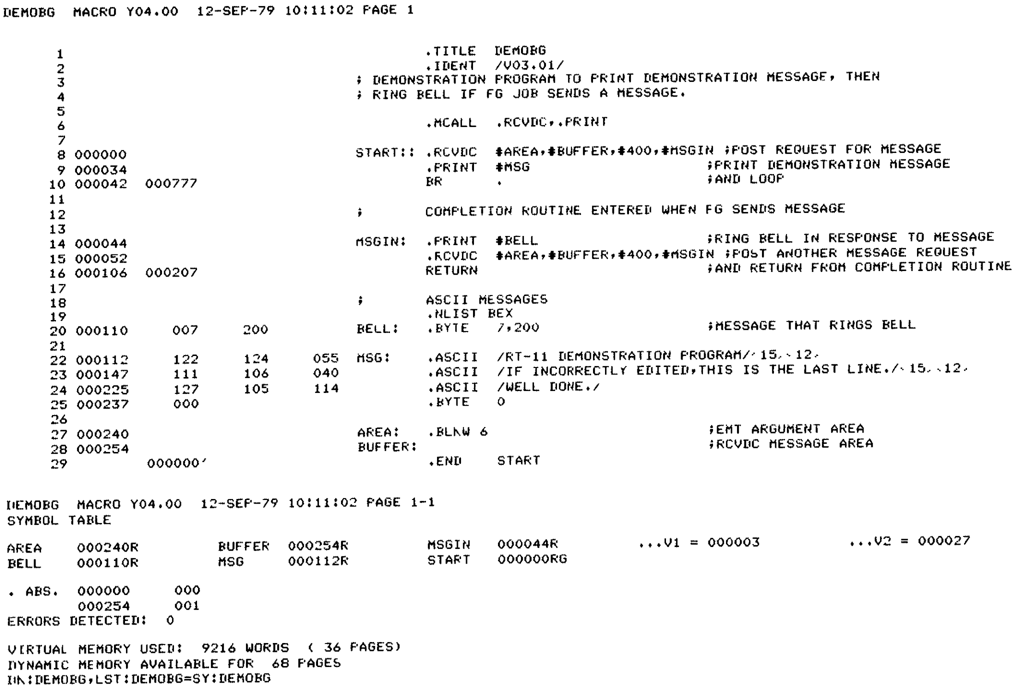
```
.MACRO/LIST:LST: SY:DEMOBG(RET)
```

(see Figure 6–2)

If any errors occur when you assemble DEMOBG.MAC, you have incorrectly edited the file and should repeat the edits. Use the backup demonstration program.

```
.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC(RET)
.
```

Figure 6–2: DEMOBG Assembly Listing



```
DEMOBG  MACRO Y04.00  12-SEP-79 10:11:02 PAGE 1

     1                                        .TITLE  DEMOBG
     2                                        .IDENT  /V03.01/
     3                               ; DEMONSTRATION PROGRAM TO PRINT DEMONSTRATION MESSAGE, THEN
     4                               ; RING BELL IF FG JOB SENDS A MESSAGE.
     5
     6                                        .MCALL  .RCVDC,.PRINT
     7
     8 000000                        START:: .RCVDC  #AREA,#BUFFER,#400,#MSGIN ;POST REQUEST FOR MESSAGE
     9 000034                                .PRINT  #MSG                      ;PRINT DEMONSTRATION MESSAGE
    10 000042  000777                        BR      .                         ;AND LOOP
    11
    12                               ;       COMPLETION ROUTINE ENTERED WHEN FG SENDS MESSAGE
    13
    14 000044                        MSGIN:  .PRINT  #BELL                     ;RING BELL IN RESPONSE TO MESSAGE
    15 000052                                .RCVDC  #AREA,#BUFFER,#400,#MSGIN ;POST ANOTHER MESSAGE REQUEST
    16 000106  000207                        RETURN                            ;AND RETURN FROM COMPLETION ROUTINE
    17
    18                               ;       ASCII MESSAGES
    19                                       .NLIST BEX
    20 000110     007     200        BELL:   .BYTE   7,200                     ;MESSAGE THAT RINGS BELL
    21
    22 000112     122     124     055 MSG:   .ASCII  /RT-11 DEMONSTRATION PROGRAM/<15><12>
    23 000147     111     106     040        .ASCII  /IF INCORRECTLY EDITED,THIS IS THE LAST LINE./<15><12>
    24 000225     127     105     114        .ASCII  /WELL DONE./
    25 000237     000                        .BYTE   0
    26
    27 000240                        AREA:   .BLKW 6                           ;EMT ARGUMENT AREA
    28 000254                        BUFFER:                                   ;RCVDC MESSAGE AREA
    29         000000'                       .END    START

DEMOBG  MACRO Y04.00  12-SEP-79 10:11:02 PAGE 1-1
SYMBOL TABLE

AREA    000240R        BUFFER  000254R       MSGIN   000044R       ...V1 = 000003        ...V2 = 000027
BELL    000110R        MSG     000112R       START   000000RG

. ABS.  000000     000
        000254     001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  9216 WORDS  ( 36 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  68 PAGES
DK:DEMOBG,LST:DEMOBG=SY:DEMOBG
```

6.9.1.4 Linking the Demonstration Program — Now, link the program DEMOBG.

```
.LINK DEMOBG(RET)
.
```

6.9.1.5 Running the Demonstration Program — Run the program DEMOBG to check the results of the first exercise.

```
.RUN DEMOBG(RET)
RT-11 DEMONSTRATION PROGRAM
IF INCORRECTLY EDITED, THIS IS THE LAST LINE.
WELL DONE.
(CTRL/C)
(CTRL/C)
.
```

NOTE

> If your configuration includes a graphics display, turn it off at this point. Type GT OFF.

If you incorrectly edited the file, you can repeat this exercise, although you can continue without correcting the file. If, however, you want to repeat the exercise, begin by using the backup demonstration program.

```
.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC(RET)
.
```

Then, repeat the editing procedure.

6.9.2 Foreground/Background Monitor Exercise

For the foreground/background monitor exercise, assemble a second program (DEMOFG.MAC), link it for the foreground, and execute it in conjunction with DEMOBG (procedures follow). DEMOFG is a small foreground program that sends a message every two seconds to DEMOBG (running in the background), telling it to ring the terminal bell. Besides printing the terminal message used in the single-job exercise, DEMOBG recognizes these messages and rings the bell once for each message sent.

Although DEMOFG is always active, sending messages to the background every two seconds, this exercise can execute other programs in the background besides DEMOBG. The circuit is complete and messages are successfully received and honored only when DEMOBG is active. During those periods when DEMOBG is not running, DEMOFG enters the messages in the monitor message queue. Once you restart DEMOBG in the background, the system immediately releases all the messages queued since the last forced exit, resulting in many successive bell rings. When the queue is empty, the normal send/receive cycle resumes and the bell rings every two seconds as each current message is sent and honored.

6.9.2.1 Bootstrapping the FB Monitor and Getting Started — Make sure that the RT–11 single-job monitor is running. Then boot the foreground/background monitor.

```
.BOOT RT11FB.SYS(RET)
RT-11FB V04.00x
(Followed by any startup file commands.)

.ASSIGN DY1: DK:(RET)
.
```

NOTE

> If your configuration includes a VT11 or VS60 display processor and scope, shift system output to the display scope. Type GT ON. Verify that the scope is on by turning the BRIGHTNESS knob to an adequate level. You can still enter commands at the keyboard, but the echo is on the screen.

Now, enter the time of day, using the following command. In the command, hh:mm:ss is the hour, minutes, and seconds in the form 13:12:00 (1:12 p.m.).

```
.TIME hh:mm:ss(RET)
.
```

6.9.2.2 Assembling the Demonstration Program — Assemble the foreground demonstration program, DEMOFG.MAC.

```
.MACRO SY:DEMOFG(RET)
ERRORS DETECTED:  0
.
```

6.9.2.3 Linking the Demonstration Program — Link DEMOFG for the foreground.

```
.LINK/FOREGROUND DEMOFG(RET)
.
```

6.9.2.4 Running the Demonstration Programs — Start DEMOFG as the foreground job.

```
.FRUN DEMOFG(RET)
F>
FOREGROUND DEMONSTRATION PROGRAM
SENDS A MESSAGE TO THE BACKGROUND PROGRAM DEMOBG
EVERY 2 SECONDS, TELLING IT TO RING THE BELL.
B>
```

DEMOFG is running and queuing the message for DEMOBG every two seconds. Now execute DEMOBG in the background and receive the messages.

```
RUN DEMOBG(RET)
```

(The bell rings quickly several times, then once every two seconds.)

```
RT-11 DEMONSTRATION PROGRAM
IF INCORRECTLY EDITED, THIS IS THE LAST LINE.
WELL DONE.
```

Execute a DIRECTORY command in the background to obtain a directory listing.

```
(CTRL/C)
(CTRL/C)
```

(The bell stops.)

```
.DIRECTORY<RET>
 dd-mmm-yy
```

(The directory of the device DK prints on the terminal.)

```
.
```

Rerun DEMOBG to collect all the foreground messages queued while the directory was printing.

```
.RUN DEMOBG<RET>
```

(The bell rings several times in rapid succession, then rings once every two seconds.)

```
RT-11 DEMONSTRATION PROGRAM
IF INCORRECTLY EDITED, THIS IS THE LAST LINE.
WELL DONE.
<CTRL/C>
<CTRL/C>
.
```

(The bell stops.)

Now, stop the foreground program and remove it from memory.

```
<CTRL/F>
F>
<CTRL/C>
<CTRL/C>
B>
UNLOAD F<RET>
.
```

NOTE

> If your configuration includes a graphics display, turn it off at this point. Type GT OFF.

If you completed these exercises without error, your system has passed this minimal test and you can consider it successfully installed.

6.10 Performing the System Generation Process

If you have decided that you need RT–11 features that are available only if you generate your own monitor(s) and handlers, perform the system generation process at this point. You should have thoroughly studied Chapter 1 to make this decision and to establish that you can perform the system generation process on your particular hardware configuration. Turn to Part III (Chapters 8, 9, and 10) for guidance in planning and performing system generation.

Chapter 7
Installing a System Distributed on Magtape to Run on Hard Disk

If RT–11 was distributed to you on magtape and you intend to build a system to run on hard disk, perform the procedures described in this chapter. These procedures assume that your configuration includes only one disk drive. With two disk drives, you still have to perform the logical steps summarized in the following list; however, you can simplify some procedures by using the additional disk drive. Sections 7.1 through 7.11 describe the procedures involved in each step. Figure 7–1 shows the various backup magtapes you create when you install RT–11.

NOTE

> If your hardware configuration includes a VT100 terminal, be sure to set auto XOFF/XON in SETUP mode before attempting to bootstrap RT–11. Never set TT NOPAGE when you use this terminal. Refer to your hardware manuals for more information about these settings.

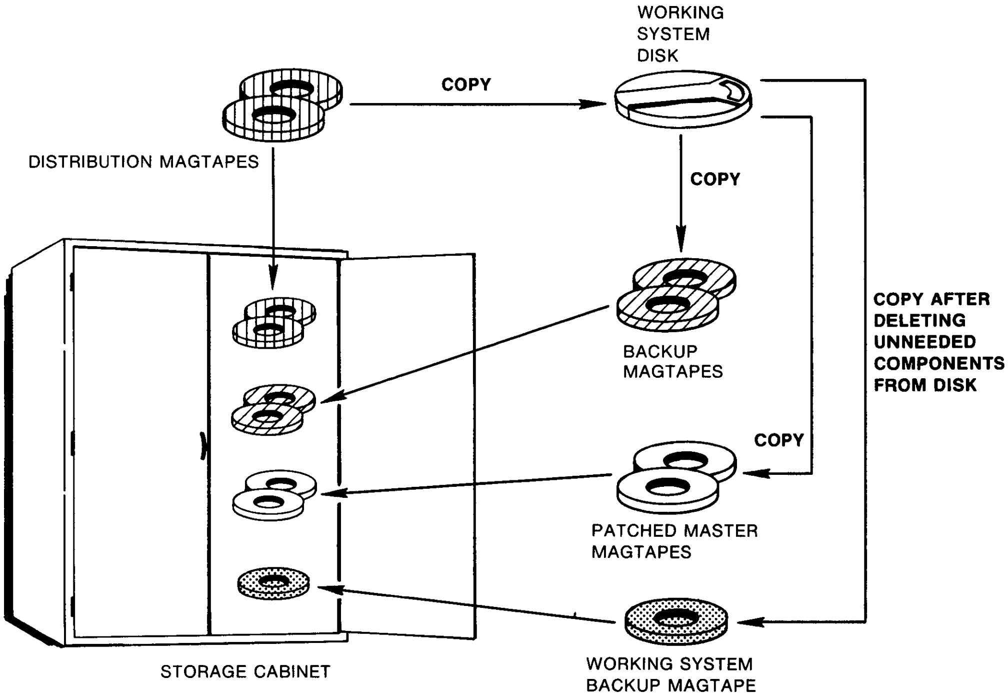
1. Bootstrap the distribution magtape.
2. Preserve the distribution magtapes.
3. Install mandatory patches.
4. Create patched master magtapes.
5. Create the system on the disk from chosen components.
6. Install the bootstrap on the disk.
7. Customize the system.
8. Compress the disk.
9. Preserve the working system.

10. Test the working system.

11. If appropriate, perform the system generation process.

The following sections correspond to each of these steps and describe in detail the procedures you must perform to complete each step.

Figure 7–1: Sample Backup Magtapes



7.1 Bootstrapping the Distribution Magtape

The first procedure you perform when installing RT–11 is bootstrapping the distribution magtape.

Begin by making sure that the processor is powered up but not running. Mount the distribution magtape labeled 1/2 (remove the write ring from the back of the tape reel). Manually position the magtape at the load point if the magtape is not in this position.

If the system has a hardware bootstrap capable of bootstrapping the magtape, boot the tape and proceed. If not, use the toggle-in software bootstap in Appendix E.

The magtape moves as the primary bootstrap loads the secondary bootstrap file MSBOOT.BOT.

MSBOOT responds on the terminal.

```
MSBOOT V0x-yy
*
```

The next step depends on whether you have TM11 as a magtape device, or TJU16. If you have a TM11, use the magtape build program named MDUP.MT. If you have a TJU16, use the magtape build program named MDUP.MM.

```
MDUP.MT(RET)
```

or

```
MDUP.MM(RET)
```

The magtape should move as the specified MDUP program is loaded.

```
MDUP V0x.yy
*
```

The next procedure is to build a minimal system on the disk. Start by initializing the system disk. If your disk is an RK05, be sure to use a formatted disk (see Section 2.7).

You need to initialize the system disk and scan it for bad blocks before you can copy system files to it. You specify these operations to MDUP by entering the device name followed by combinations of the following options.

/Z to initialize the disk
/B to scan it for bad blocks

Mount a formatted disk (write enabled).

Use the following command, where xx is the permanent device name (RK, DL, DM, or DP) for your disk.

```
xx0:/Z/B(RET)
```

The system now scans the disk for bad blocks and creates a directory.

```
*
```

Now you can build a minimal system on the disk. Depending on the type of tape drive you have, use one of the two following commands (where xx is the permanent device name for your disk).

If the magtape is TM11:

```
xx0:A=MT0:(RET)
```

If the magtape is TJU16:

```
xx0:A=MM0:(RET)
```

The tape moves while the system copies a single-job monitor, swap file, system device handler, terminal handler, line printer handler, magtape handler, PIP, DUP, and DIR to the disk. When the files are copied, MDUP boots the minimal system from the disk.

```
RT-11SJ V04.00
?KMON-F-Command file not found
.
```

You are now running from the minimal system on the disk. This minimal system supports enough file maintenance commands to allow you to complete the building process.

NOTE

> MDUP does not support automatic replacement of bad blocks for RK06, RK07, and RL01 disks. If your disk is an RK06, RK07, or RL01 and you want automatic bad block replacement, you must initialize a second disk and copy your files to it at a later time.

Use the DATE command to set the date (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
.DATE dd-mmm-yy(RET)
.
```

Now copy the rest of the files from distribution magtape number 1 to the disk. Use the following command, where xx is MT or MM.

```
.COPY/SYSTEM/NOREPLACE  xx0:*.* DK:(RET)
  Files copied:
xx0:aaaaaa.ttt to DK:aaaaaa.ttt
           .
           .
xx0:zzzzzz.ttt to DK:zzzzzz.ttt
?PIP-W-Reboot
.
```

NOTE

> You must use the NOREPLACE option in this command so that the files you copied to the disk when you built the minimal system will not be copied again. The system prints a message to tell you which files it does not copy (for example: SWAP.SYS not copied).

Reboot as advised.

```
.BOOT SY:(RET)
```

7.2 Preserving the Distribution Magtapes

Copy the distribution magtapes for backup as a safety measure in case of machine failure or human error. To back up the magtapes, complete the steps summarized in the list and described in the following sections.

1. Replace distribution tape number 1 in the tape drive with a blank tape.
2. Copy all the distribution files from the disk to the blank backup tape.
3. Replace the backup tape with distribution tape number 2.
4. Copy everything from distribution tape number 2 to disk.
5. Replace distribution tape number 2 with a blank tape.
6. Copy the files from distribution tape number 2 to backup tape number 2.

Step 1

Remove distribution magtape number 1 and mount a blank magtape on the drive. Leave the write ring in the back of the reel and make sure that the tape is positioned at the load point. Then initialize the blank magtape and write the primary bootstrap on it. Use the following command, where xx is MT or MM.

```
.INITIALIZE/FILE:MBOOT.BOT xx0:(RET)
xx0:/Initialize; Are you sure?Y(RET)
.
```

Step 2

Copy all the files from the disk to the magtape by invoking the indirect command file DISMT1.COM (procedure follows). DISMT1.COM copies a duplicate of distribution magtape 1. Before you invoke this indirect command file, use the ASSIGN command to assign the logical name DIS: to your disk device and the logical name TAP: to your magtape device. In the commands, xx is MT or MM and yy is your disk.

```
.ASSIGN xxn: TAP:(RET)
.ASSIGN yyn: DIS:(RET)
.@DISMT1(RET)
(The commands in the indirect file print on the terminal.)
```

Step 3

Rewind the newly created backup magtape, remove it, and label it "backup RT–11 V04 1/2."

Mount distribution magtape number 2 (remove the write ring from the back of the reel).

Step 4

Copy all the files from distribution magtape number 2 to the disk. Use the following command, where xx is MT or MM.

```
.COPY/SYSTEM/NOREPLACE xx0:*.* DK:(RET)
  Files copied:
xx0:aaaaaa.ttt to DK:aaaaaa.ttt
                  .
                  .
                  .
                  .
xx0:zzzzzz.ttt to DK:zzzzzz.ttt
?PIP-W-Reboot
.
```

Reboot the disk.

Step 5

Rewind the distribution magtape and remove it. Mount another blank magtape, leaving the write ring in the back of the reel. Make sure that the tape is positioned at the load point. Then initialize the blank magtape. Use the following command, where xx is MT or MM. Because the second magtape need not be bootable, you do not need to use the /FILE:MBOOT.BOT option.

```
.INITIALIZE xx0:(RET)
xx0:/Initialize; Are you sure?Y(RET)
.
```

Step 6

Use the indirect command file DISMT2.COM (procedure follows) to duplicate distribution magtape 2 on the newly initialized blank magtape. Use the ASSIGN command to assign the logical name DIS: to your disk device and the logical name TAP: to your magtape device. In the commands, xx is MT or MM and yy is your disk.

```
.ASSIGN xxn: TAP:(RET)
.ASSIGN yyn: DIS:(RET)
.@DISMT2(RET)
(The commands in the indirect file print on the terminal.)
```

Then, rewind the newly created backup magtape, remove it, and label it "backup RT-11 V04 2/2."

Store the distribution magtapes.

7.3 Installing Mandatory Patches

To make sure that RT-11 operates correctly, install mandatory patches at this point. Mandatory patches are critical to system or component operation. They correct software errors discovered since the software was released. Chapter 2 describes how to identify the appropriate patches.

NOTE

If you are using Autopatch to install mandatory patches, do not read the remainder of this section. Instead, turn to the *RT-11 Autopatch User's Guide* for assistance.

Most published patches are binary patches that you install with the Save Image Patch Program (SIPP). In the event that monitors or handlers require patches, the patches may be published in both binary and source form. If you later perform the system generation process to create your own monitors and handlers, you need source files with any patches applied.

Therefore, you must install all source patches, using the Source Language Patch (SLP) utility program.

To install binary patches, invoke SIPP, as indicated, and enter the values specified in the published patch.

```
.R SIPP(RET)
*filnam.typ/C(RET)
Base?         xxxx(RET)
Offset?       xx(RET)
     Base         Offset       Old      New?
   xxxxxx         xxxxxx    xxxxxx      xxxxxx(RET)
   xxxxxx         xxxxxx    xxxxxx      (CTRL/Y)(RET)
Checksum?     xxxxx(RET)
*(CTRL/C)
.
```

The input file, filnam.typ, is the component to be patched. Respond to SIPP's prompts with the values specified in the published patch. The base and offset describe the locations to be patched. The patch may look different from this example if the component is overlaid. CTRL/Y causes SIPP to ask for the checksum. If the checksum matches, SIPP installs the patch and prompts with an asterisk. If you have typed the patch exactly as published, the checksum will match. If the checksum does not match, issue ;V to display modified locations. Correct any errors. Then type CTRL/Y and RETURN.

Refer to the *RT–11 System User's Guide* for more information about SIPP.

To install source patches, create the file, using any editor, exactly as published. The following is an example of a source patch. Use a text editor to create the SLP command file, filnam.SLP, as follows.

```
-563
(TAB)DEC(TAB)R5
-660,661
(TAB)BIT(TAB) SPECL$!FILST$,@R2
(TAB)BEQ(TAB)17$
-682
(TAB)MOV(TAB)R3,R0
-700,701
(TAB)BIT(TAB) SPECL$!FILST$,@R2
(TAB)BEQ(TAB)23$
/
```

Then use the SLP utility to patch filnam.MAC as follows:

```
.R SLP(RET)
*filnam.MAC=filnam.MAC,filnam.SLP(RET)
*(CTRL/C)
.
```

Use these procedures to make all the mandatory binary and source patches. Patch each component separately, following the sequence specified in the *RT–11 Software Dispatch Review*.

NOTE

DIGITAL strongly recommends that you analyze all future patches as soon as they are published. Install mandatory patches immediately. Ascertain whether optional patches are useful for your application. Often, you can save much time and effort through such analysis, because these patches may solve a problem you have already experienced.

7.4 Creating Patched Master Magtapes

Once you have installed patches in the components you copied to the disk, copy all the files to blank magtapes. These magtapes can serve as patched masters. As other mandatory patches are published, you can copy the affected component from the patched master magtape to the disk, install the patch, and copy the file back to the patched master. In this way, you can make sure that you have up-to-date, patched versions of all RT–11 components, even if your working system is destroyed.

Mount another blank magtape, leaving the write ring in the back of the reel. Make sure that the tape is positioned at the load point. Then initialize the blank magtape. Use the following command, where xx is MT or MM.

```
.INITIALIZE/FILE:MBOOT.BOT xx0:(RET)
xx0:/Initialize; Are you sure?Y(RET)
.
```

Use the following procedures to copy the same files to patched master magtape 1 as you copied to backup magtape 1 and the same files to patched master magtape 2 as you copied to backup magtape 2. In the commands, xx is MT or MM and yy is your disk.

```
.ASSIGN xxn: TAP:(RET)
.ASSIGN yyn: DIS:(RET)
.@DISMT1(RET)
(The commands in the indirect file print on the terminal.)
```

Rewind the newly created magtape, remove it, and label it "Patched Master RT–11 V04 1/2." Mount another blank magtape, leaving the write ring in the back of the reel. Make sure that the tape is positioned at the load point. Then initialize the blank magtape. Use the following command, where xx is MT or MM.

```
.INITIALIZE xx0:(RET)
xx0:/Initialize; Are you sure?Y(RET)
.
```

Copy the same files (now patched) to this magtape that you copied to Backup Magtape number 2. In the command xx is MM or MT and yyn is your disk.

```
.ASSIGN xxn: TAP:RET
.ASSIGN yyn: DIS:RET
.@DISMT2RET
(The commands in the indirect file print on the terminal.)
```

Then, rewind the newly created magtape, remove it, and label it "Patched Master RT–11 V04 2/2."

Store the patched master magtapes, and use them when you install any future patches.

7.5 Creating the Working System from Chosen Components

Once you have chosen your system components (Section 2.3), backed up the distribution, and created patched masters, you can create the working system by deleting unneeded components from the disk.

The following command queries you about all the files on the disk. Choose the files you want to delete.

```
.DELETE/SYSTEM *.*RET
 Files deleted:
DK:aaaaaa.ttt? YRET (to delete a specific file)
DK:bbbbbb.ttt? NRET (to include a specific file)
(and so on)
```

7.6 Installing the Bootstrap on the Disk

Once you have created your system, copy the bootstrap from the monitor file of your choice (RT11BL, RT11SJ, or RT11FB) to the disk. In the command, xx is the permanent device name for your disk and yy is BL, SJ, or FB.

```
.COPY/BOOT xx0:RT11yy.SYS xx0:RET
.
```

Then, halt the processor and use the hardware bootstrap to boot the working system disk.

```
RT-11FB V04.00
(Followed by any startup file commands.)
.
```

NOTE

If the disk does not boot, repeat the procedures to this point.

Set the date, using the following command (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
.DATE dd-mmm-yyRET
.
```

7.7 Customizing the System

You may want to make certain customizations (described in Section 2.8) to the distributed RT–11 components. At this point, perform the procedures to implement any of these customizations. Table 1–4 summarizes the available customizations and directs you to the section in Chapter 2 that describes a particular customization and the procedure for implementing it.

NOTE

> You can perform the system generation process to implement additional customizations. See Section 7.11.

7.8 Compressing the Disk

DIGITAL recommends that you compress the working system volume to make its free space contiguous. Consolidating free space allows you to use space on the volume that would otherwise be too fragmented to be usable.

Use the SQUEEZE command for this procedure. (The volume must be write enabled.) The squeeze operation does not move files with the .BAD file type. In the command, xx is the device name for your disk.

```
.SQUEEZE xxn:(RET)
xxn:/Squeeze; Are you sure? Y(RET)
RT-11xx V04.00x
(Followed by any startup file commands.)
.
```

The system automatically reboots when you compress a system volume.

7.9 Preserving the Working System

Once you build a satisfactory working system, DIGITAL recommends that you protect all the files in it and preserve it on the backup medium of your choice. The following procedure describes backing up to magtape.

First, protect all the files on the disk.

```
.RENAME/SYSTEM/PROTECTION *.* *.*(RET)
 Files renamed:
DK:aaaaaa.ttt  to DK:aaaaaa.ttt
DK:bbbbbb.ttt  to DK:bbbbbb.ttt
DK:cccccc.ttt  to DK:cccccc.ttt
DK:dddddd.ttt  to DK:dddddd.ttt
           .
           .
           .
DK:zzzzzz.ttt  to DK:zzzzzz.ttt
.
```

Now use the following procedure to initialize another blank magtape and copy files to it. You may be able to fit the working system on one backup magtape. You can place approximately 3000 blocks on a 600 foot 800 bpi magtape.

Mount a blank magtape, leaving the write ring in the back of the reel. Make sure that the tape is positioned at the load point. Use the following command, where xx is MT or MM.

```
.INITIALIZE/FILE:MBOOT.BOT xx0:(RET)
xx0:/Initialize; Are you sure? (RET)
.
```

Use the following procedure to copy all the files from the disk to the magtape. Start with the following files, in the order shown. This procedure is essentially the procedure you should use whenever you build a bootable magtape. If you do not copy the files in this order, bootstrapping the magtape will be a painfully long process. Keep track of all the files you copy so that when you copy the rest of the files you will know which files you have already copied. If you have deleted some device handlers from the system, the log that prints on the terminal will not include the deleted device handlers. The following example shows a log from a system that includes all the device handlers. In the commands, xx is MT or MM.

```
.COPY/SYSTEM MSBOOT.BOT xx0:MSBOOT.BOT/POSITION:-1(RET)

.COPY/SYSTEM MDUP.* xx0:MDUP.*/POSITION:-1(RET)
 Files copied:
DK:MDUP.MM to xx0:MDUP.MM
DK:MDUP.MT to xx0:MDUP.MT
DK:MDUP.MS to xx0:MDUP.MS

.COPY/SYSTEM SWAP.SYS xx0:SWAP.SYS/POSITION:-1(RET)

.COPY/SYSTEM RT11SJ.SYS xx0:RT11SJ.SYS/POSITION:-1(RET)

.COPY/SYSTEM TT.SYS xx0:TT.SYS/POSITION:-1(RET)

.COPY/SYSTEM D%.SYS xx0:*.SYS/POSITION:-1(RET)
  Files copied:
DK:DT.SYS to xx0:DT.SYS
DK:DP.SYS to xx0:DP.SYS
DK:DX.SYS to xx0:DX.SYS
DK:DY.SYS to xx0:DY.SYS
DK:DL.SYS to xx0:DL.SYS
DK:DM.SYS to xx0:DM.SYS
DK:DS.SYS to xx0:DS.SYS
DK:DD.SYS to xx0:DD.SYS

.COPY/SYSTEM R%.SYS xx0:*.SYS/POSITION:-1(RET)
  Files copied:
DK:RK.SYS to xx0:RK.SYS
DK:RF.SYS to xx0:RF.SYS

.COPY/SYSTEM M%.SYS xx0:*.SYS/POSITION:-1(RET)
  Files copied:
DK:MM.SYS to xx0:MM.SYS
DK:MT.SYS to xx0:MT.SYS
DK:MS.SYS to xx0:MS.SYS
```

```
.COPY/SYSTEM L%.SYS xx0:*.SYS/POSITION:-1(RET)
  Files copied:
DK:LS.SYS to xx0:LS.SYS
DK:LP.SYS to xx0:LP.SYS
.COPY PIP.SAV xx0:PIP.SAV/POSITION:-1(RET)
.COPY DUP.SAV xx0:DUP.SAV/POSITION:-1(RET)
.COPY DIR.SAV xx0:DIR.SAV/POSITION:-1(RET)
.
```

Copy the rest of the files. The following command queries you about all the files on the disk so that you can choose the files it copies.

```
.COPY/SYSTEM/QUERY DK: xx0:/POSITION:-1(RET)
  Files copied:
DK:aaaaaa.ttt to xx0:aaaaaa.ttt? Y(RET) (to include a specific file)
DK:zzzzzz.ttt to xx0:zzzzzz.ttt? N(RET) (to exclude a specific file)
(and so on)
.
```

Rewind the newly created working system backup, remove it, and label it "backup working system RT–11 V04," and store the backup.

7.10 Testing the Working System

Once you have built and preserved the working system, you can execute the following demonstration to test the system. This demonstration does not serve as a comprehensive system exercise; however, because it uses several major system components, it does serve as a minimal integrity check. Moreover, DIGITAL considers your system officially installed if the demonstration runs without error.

To execute this demonstration, you should have at least 20 free blocks on your working system device and the working system must include at least the following components. To perform the foreground/background monitor exercise, you need a terminal with a bell.

SWAP.SYS
RT11SJ.SYS
RT11FB.SYS (if you do the foreground/background exercise)
xx.SYS (system device handler)
TT.SYS (for SJ monitor)
LP.SYS (if appropriate)
EDIT.SAV
MACRO.SAV
SYSMAC.SML
LINK.SAV
PIP.SAV
DUP.SAV
DIR.SAV
DEMOBG.MAC
DEMOFG.MAC

You can do only the single-job monitor exercise to test the single-job monitor or you can do both exercises to test both the single-job monitor and the foreground/background monitor.

7.10.1 Single-Job Monitor Exercise

For the single-job monitor exercise, you edit, assemble, link, and execute a small program (DEMOBG.MAC). When you execute DEMOBG, it displays a message on the terminal.

7.10.1.1 Bootstrapping the SJ Monitor and Getting Started — Write enable the system disk and bootstrap the system.

```
RT-11SJ V04.00x
(Followed by any startup file commands.)
.
```

Set the date, using the following command (where dd-mmm-yy is the day, month, and year in the form 10-JAN-80).

```
.DATE dd-mmm-yy(RET)
.
```

NOTE

> If your configuration includes a VT11 or VS60 display processor and scope, shift system output to the display scope. Type GT ON. Verify that the scope is on by turning the BRIGHTNESS knob to an adequate level. You can still enter commands at the keyboard, but the echo is on the screen.

Display the directory of the system disk on the terminal. The directory varies according to your particular working system. As long as a directory prints, you need not worry if it does not match the one in the following example.

```
.DIRECTORY/BRIEF/COLUMNS:1 SY:(RET)
  dd-mmm-yy
SWAP  .SYS
RT11SJ.SYS
LP    .SYS
DL    .SYS
TT    .SYS
MT    .SYS
SIPP  .SAV
EDIT  .SAV
MACRO .SAV
SYSMAC.SML
LINK  .SAV
ODT   .OBJ
PIP   .SAV
     .
     .
     .
xxx Files, bbb Blocks
fff Free blocks
.
```

NOTE

If you have shifted output to a scope and the directory scrolls by too quickly to read, type CTRL/S to stop the display and CTRL/Q to restart it.

7.10.1.2 Editing the Demonstration Program — Use the text editor to modify the demonstration program, DEMOBG.MAC. If DEMOBG.MAC is a protected file, remove the protection before making the edits (RENAME/NOPROTECTION SY:DEMOBG.MAC SY:DEMOBG.MAC).

```
.EDIT SY:DEMOBG.MAC(RET)
*F;(TAB).ASCII(ESC)(ESC)
*0AD(ESC)(ESC)
*EX(ESC)(ESC)
.
```

7.10.1.3 Assembling the Demonstration Program — To assemble DEMOBG.MAC and obtain a listing, make sure that your configuration has a line printer that is on-line and ready.

```
.ASSIGN LP: LST:(RET)
.
```

NOTE

If your configuration does not include a line printer, use the console terminal.

```
.ASSIGN TT: LST:(RET)
.
```

Assemble DEMOBG.MAC as follows:

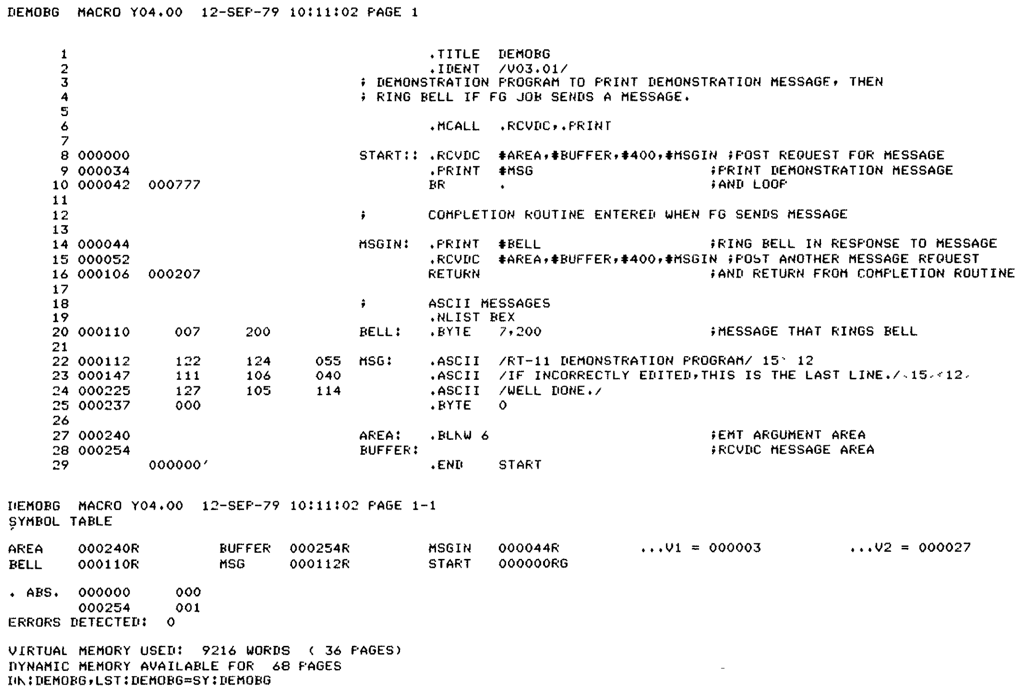
```
.MACRO/LIST:LST: SY:DEMOBG(RET)
```

(see Figure 7–2)

If any errors occur when you assemble DEMOBG.MAC, you have incorrectly edited the file and should repeat the edits. Use the backup demonstration program.

```
.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC(RET)
.
```

Figure 7–2: DEMOBG Assembly Listing



```
DEMOBG  MACRO Y04.00  12-SEP-79 10:11:02 PAGE 1

                                          .TITLE  DEMOBG
                                          .IDENT  /V03.01/
                                  ; DEMONSTRATION PROGRAM TO PRINT DEMONSTRATION MESSAGE, THEN
                                  ; RING BELL IF FG JOB SENDS A MESSAGE.

                                          .MCALL  .RCVDC,.PRINT

   000000                         START:: .RCVDC  #AREA,#BUFFER,#400,#MSGIN ;POST REQUEST FOR MESSAGE
   000034                                 .PRINT  #MSG                      ;PRINT DEMONSTRATION MESSAGE
   000042  000777                         BR      .                         ;AND LOOP

                                  ;       COMPLETION ROUTINE ENTERED WHEN FG SENDS MESSAGE

   000044                         MSGIN:  .PRINT  #BELL                     ;RING BELL IN RESPONSE TO MESSAGE
   000052                                 .RCVDC  #AREA,#BUFFER,#400,#MSGIN ;POST ANOTHER MESSAGE REQUEST
   000106  000207                         RETURN                            ;AND RETURN FROM COMPLETION ROUTINE

                                  ;       ASCII MESSAGES
                                          .NLIST BEX
   000110     007     200         BELL:   .BYTE   7,200                     ;MESSAGE THAT RINGS BELL

   000112     122     124     055 MSG:    .ASCII  /RT-11 DEMONSTRATION PROGRAM/<15><12>
   000147     111     106     040         .ASCII  /IF INCORRECTLY EDITED,THIS IS THE LAST LINE./<15><12>
   000225     127     105     114         .ASCII  /WELL DONE./
   000237     000                         .BYTE   0

   000240                         AREA:   .BLKW 6                           ;EMT ARGUMENT AREA
   000254                         BUFFER:                                   ;RCVDC MESSAGE AREA
           000000'                        .END    START

DEMOBG  MACRO Y04.00  12-SEP-79 10:11:02 PAGE 1-1
SYMBOL TABLE

AREA     000240R        BUFFER  000254R        MSGIN   000044R        ...V1 = 000003         ...V2 = 000027
BELL     000110R        MSG     000112R        START   000000RG

. ABS.   000000     000
         000254     001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  9216 WORDS  ( 36 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  68 PAGES
DK:DEMOBG,LST:DEMOBG=SY:DEMOBG
```

7.10.1.4 Linking the Demonstration Program — Now, link the program DEMOBG.

```
.LINK DEMOBG<RET>
.
```

7.10.1.5 Running the Demonstration Program — Run the program DEMOBG to check the results of the first exercise.

```
.RUN DEMOBG<RET>
RT-11 DEMONSTRATION PROGRAM
IF INCORRECTLY EDITED, THIS IS THE LAST LINE.
WELL DONE.
CTRL/C
CTRL/C
.
```

NOTE

> If your configuration includes a graphics display, turn it off at this point. Type GT OFF.

If you incorrectly edited the file, you can repeat this exercise, although you can continue without correcting the file. If, however, you want to repeat the exercise, begin by using the backup demonstration program.

```
.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC<RET>
.
```

Then, repeat the editing procedure.

7.10.2 Foreground/Background Monitor Exercise

For the foreground/background monitor exercise, assemble a second program (DEMOFG.MAC), link it for the foreground, and execute it in conjunction with DEMOBG (procedures follow). DEMOFG is a small foreground program that sends a message every two seconds to DEMOBG (running in the background), telling it to ring the terminal bell. Besides printing the terminal message used in the single-job exercise, DEMOBG recognizes these messages and rings the bell once for each message sent.

Although DEMOFG is always active, sending messages to the background every two seconds, this exercise can execute other programs in the background besides DEMOBG. The circuit is complete and messages are successfully received and honored only when DEMOBG is active. During those periods when DEMOBG is not running, DEMOFG enters the messages in the monitor message queue. Once you restart DEMOBG in the background, the system immediately releases all the messages queued since the last forced exit, resulting in many successive bell rings. When the queue is empty, the normal send/receive cycle resumes and the bell rings every two seconds as each current message is sent and honored.

7.10.2.1 Bootstrapping the FB Monitor and Getting Started — Make sure that the RT–11 single-job monitor is running. Then boot the foreground/background monitor.

```
.BOOT RT11FB.SYS(RET)
RT-11FB V04.00x
(Followed by any startup file commands.)
.
```

NOTE

> If your configuration includes a VT11 or VS60 display processor and scope, shift system output to the display scope. Type GT ON. Verify that the scope is on by turning the BRIGHTNESS knob to an adequate level. You can still enter commands at the keyboard, but the echo is on the screen.

Now, enter the time of day using the following command. In the command, hh:mm:ss is the hour, minutes, and seconds in the form 13:12:00 (1:12 p.m.).

```
.TIME hh:mm:ss(RET)
.
```

7.10.2.2 Assembling the Demonstration Program — Assemble the foreground demonstration program, DEMOFG.MAC.

```
.MACRO SY:DEMOFG(RET)
ERRORS DETECTED:  0
.
```

7.10.2.3 Linking the Demonstration Program — Link DEMOFG for the foreground.

```
.LINK/FOREGROUND DEMOFG(RET)
.
```

7.10.2.4 Running the Demonstration Programs — Start DEMOFG as the foreground job.

```
.FRUN DEMOFG(RET)
F>
FOREGROUND DEMONSTRATION PROGRAM
SENDS A MESSAGE TO THE BACKGROUND PROGRAM DEMOBG
EVERY 2 SECONDS, TELLING IT TO RING THE BELL.
B>
```

DEMOFG is running and queuing the message for DEMOBG every two seconds. Now execute DEMOBG in the background and receive the messages.

```
RUN DEMOBG(RET)
```

(The bell rings quickly several times, then once every two seconds.)

```
RT-11 DEMONSTRATION PROGRAM
IF INCORRECTLY EDITED, THIS IS THE LAST LINE.
WELL DONE.
```

Execute a DIRECTORY command in the background to obtain a directory listing.

```
(CTRL/C)
(CTRL/C)
```

(The bell stops.)

```
.DIRECTORY(RET)
 dd-mmm-yy
```

(The directory of the device DK prints on the terminal.)

```
.
```

Rerun DEMOBG to collect all the foreground messages queued while the directory was printing.

```
.RUN DEMOBG(RET)
```

(The bell rings several times in rapid succession, then rings once every two seconds.)

```
RT-11 DEMONSTRATION PROGRAM
IF INCORRECTLY EDITED, THIS IS THE LAST LINE.
WELL DONE.
(CTRL/C)
(CTRL/C)
.
```

(The bell stops.)

Now, stop the foreground program and remove it from memory.

```
(CTRL/F)
F>
(CTRL/C)
(CTRL/C)
B>
UNLOAD F(RET)
.
```

NOTE

> If your configuration includes a graphics display, turn it off at this point. Type GT OFF.

If you completed these exercises without error, your system has passed this minimal test and you can consider it successfully installed.

7.11 Performing the System Generation Process

If you have decided that you need RT–11 features that are available only if you generate your own monitor(s) and handlers, perform the system generation process at this point. You should have thoroughly studied Chapter 1 to make this decision and to establish that you can perform the system generation process on your particular hardware configuration. Part III (Chapters 8, 9, and 10) describe planning and performing system generation.

Part III
System Generation

Chapter 8
Preparing for System Generation

You can build a unique RT–11 monitor through the system generation process; however, the procedure requires some preparation. Because the procedure lets you set the parameters for the system you want, you should be both knowledgeable about the process and also prepared with the information and selections you will be called on to provide. System generation is an interactive procedure. You must run the program SYSGEN and answer the questions it asks. To complete the process, you will have to assemble and link one or more monitors and device handlers that will then reflect the parameters you set.

The system generation process requires a significant amount of mass storage space and execution time. DIGITAL recommends that you undertake the system generation process only if none of the standard (distributed) monitors fully meets your needs.

Before you begin the actual system generation process, you should have:

1. Decided that the features you need are available only through system generation (study Chapter 1).
2. Sucessfully installed and tested your working system.
3. Read all the RT–11 documentation, giving special attention to the *RT–11 System User's Guide.*
4. Become comfortable with the keyboard monitor commands and RT–11 operating characteristics.
5. Read the installation guide for any layered products you have. (Some of these products require specific answers to SYSGEN questions.)

In addition, you should not attempt to perform the system generation process unless your hardware configuration meets certain requirements.

DIGITAL *supports* automatic system generation (under license) only on a system with at least 16K words of memory and 2000 free blocks of disk storage. The minimum configuration that DIGITAL *recommends* for system generation is a system with at least two disk drives and 24K words of memory. Chapter 9 describes the procedures for performing system generation on a disk system. DIGITAL also supports system generation on diskette systems (with at least 28K words of memory and a line printer or hardcopy terminal) if you use the manual method described in Chapter 10. However, DIGITAL does not recommend this very lengthy procedure.

To prepare for system generation, you should:

1. Study the system generation process.
2. Gather the information or make the decisions described in Sections 8.2.1 through 8.2.5.
3. Study the SYSGEN dialogue.

You can use the worksheet at the end of this chapter to jot down information, decisions, and choices. When you have completed this preparation, turn to Chapter 9 or 10 to perform the system generation process.

8.1 Survey of System Generation

Once you have gathered some information about your configuration, you can run the program SYSGEN.SAV and answer its questions to set the parameters for your special system. SYSGEN.SAV is a FORTRAN program that drives the system generation process using the input files SYSGEN.CND and SYSTBL.CND.

The questions that SYSGEN.SAV asks you are in the form of a dialogue. SYSGEN.CND contains the dialogue and the monitor conditionals. Your responses to the dialogue establish the conditionals that SYSGEN.SAV writes to the conditional file it creates, SYCND.MAC. SYSGEN.CND also calls SYSTBL.CND. SYSGEN.SAV uses SYSTBL.CND and some of the conditionals generated by SYSGEN.CND to create SYSTBL.MAC.

SYSTBL.MAC is a conditional file that sets up device table entries for each device you specify, defines device handlers, and sets up multi-terminal TCBs (terminal control blocks).

Ultimately, you must assemble and link the two conditional files with system source files (for example, you would need RMONSJ.MAC, the single-job resident monitor source file if your generated system included the single-job monitor).

The source file EDTGBL.MAC is always required, because it contains system-wide definitions.

SYSGEN.SAV creates three other files during the SYSGEN session. SYSBLD.COM, MONBLD.COM, and DEVBLD.COM are indirect command files that contain all the command strings to perform the assemblies and links that build the system you specify when you answer the dialogue questions. MONBLD.COM builds the monitor or monitors you define, DEVBLD.COM builds the device handlers you define, and SYSBLD.COM executes MONBLD and DEVBLD.

Once you have answered the SYSGEN questions, copy the system source files you need to the volume that the command files use as the source input device. Then assemble and link (using SYSBLD or MONBLD and DEVBLD, if you choose) the sources with the conditional files. You can enter the commands individually if you choose.

Figure 8–1 follows the flow of steps involved in the system generation process, and Figure 8–2 shows the files required for system generation.

Figure 8–1: System Generation Process

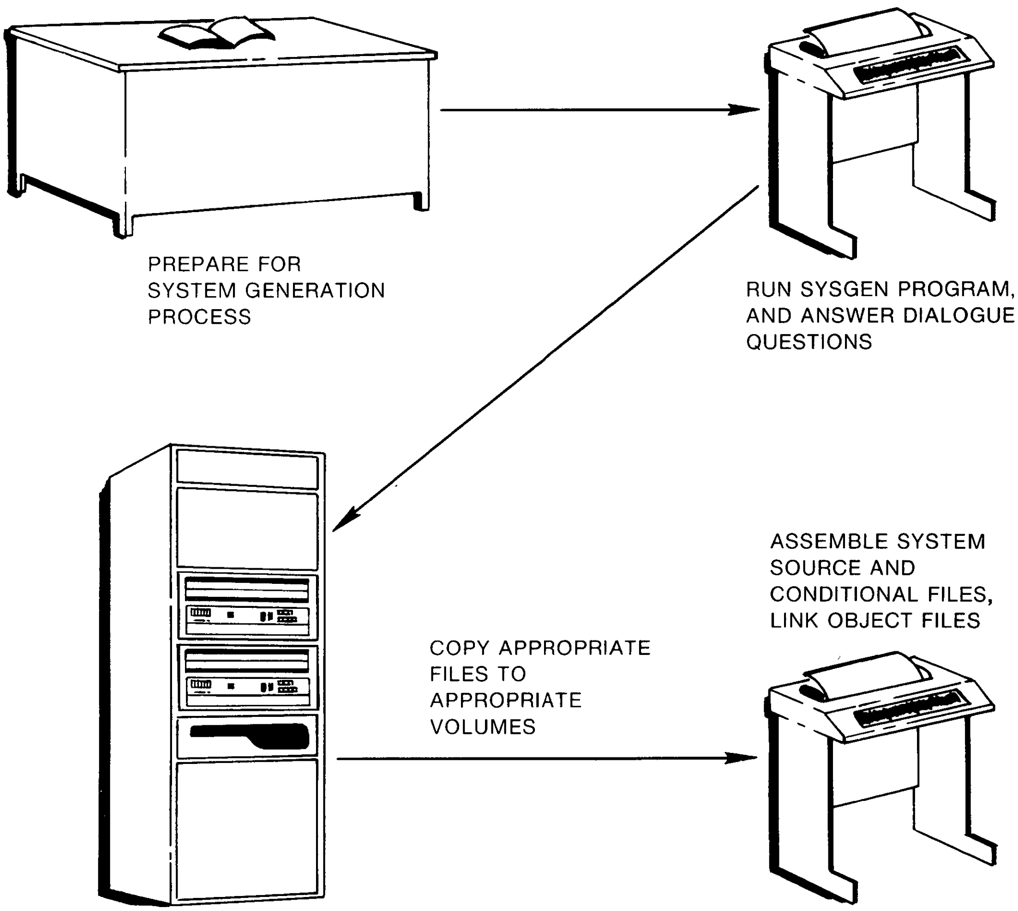
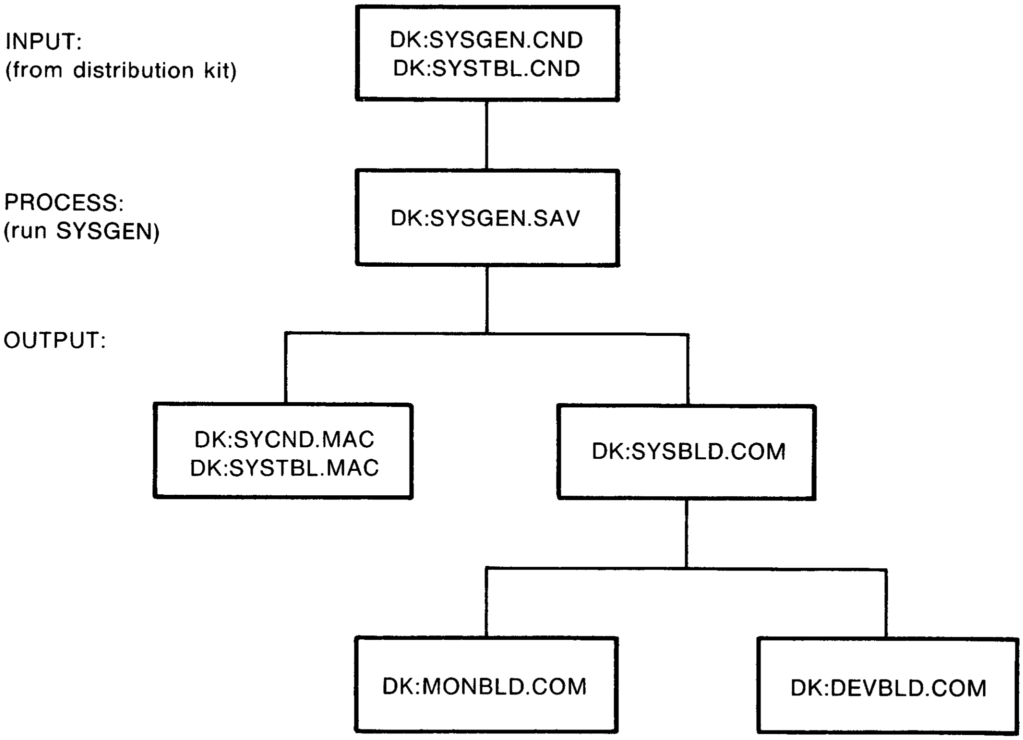


Figure 8–2: SYSGEN Input and Output Files



8.2 Gathering Information

To answer SYSGEN dialogue questions, you must be familiar with both the target system configuration and your current configuration (that is, the one on which you will run SYSGEN and assemble and link the new monitor and handlers). You should also know which monitor services you want in the target system and which vector and control status register (CSR) addresses are involved. When the time comes, you will not be able to perform the system build process unless you know your current configuration and its capabilities, you have planned an optimal file arrangement for system build on your configuration, and you have answered the relevant SYSGEN dialogue questions appropriately. In addition, you may have to edit the conditional files before you build the target system. To make sure that you have all the information you need, identify the following:

1. Peripheral devices to be supported in the special system you are generating (that is, the target system)
2. Interrupt vector and CSR addresses for devices in the target system
3. Monitor services you require for the target system
4. Edits you may need to make to the conditional files
5. Device assignments for the system build procedure

The following sections describe how to identify these items.

8.2.1 Peripheral Devices

Identify all the peripheral devices that your target system configuration will include. List the device mnemonic for each device on the worksheet at the end of this chapter (for example, DX for an RX01 diskette).

NOTE

> You may need to include some empty device slots to make sure that your system has enough device slots for your application's requirements. The number of logical device assignments you will be able to make in the working system is equal to the number of device slots available in the system.

8.2.2 Interrupt Vectors and CSRs

Identify the interrupt vector addresses and CSR addresses at which the field service technician installed each peripheral in your configuration. Although most peripherals have standard interrupt vector and CSR addresses, some have floating addresses. The presence or absence of such floating address devices affects the addresses at which the other floating address devices in the system are installed (floating address devices must be installed in a standard sequence). Even devices with standard vectors may have been installed at nonstandard addresses. When the field service technician installs the hardware, he or she is responsible for leaving a written record of this address information at the customer site, usually on a sticker attached to the processor. List each address on the worksheet.

8.2.3 Monitor Services for Target Application

The standard monitors have been generated through the system generation process; you will probably need to select support for features that standard monitors also support. In other words, many SYSGEN questions offer features you will recognize as standard features in the distributed monitors. Refer to Table 1–5 for a summary of the special features that are available through the system generation process.

Every SYSGEN dialogue question is a SYSGEN option. Some options enable support for monitor services, while other options enable support for peripheral devices and device interfaces. The options that enable monitor services are as follows:

Asynchronous terminal status

BATCH support

Device time-out support

Error logging support

Error messages on system I/O errors

Floating point support

50 Hz clock rather than 60 Hz

Idle loop light pattern

Keyboard monitor command subsets

Memory parity

Month rollover

Multi-terminal support

Multi-terminal time-out

Power failure message

Programmable clock as system clock

Ring buffer sizes

SJ timer support

.SPCPS programmed request

Startup indirect command file

System jobs

Section 8–3 describes each dialogue question in detail. When you study the dialogue, check off on the worksheet any options you want to select when you actually run SYSGEN.

8.2.4 Edits to SYCND.MAC File

Before you assemble your monitor, it is possible to edit SYCND.MAC, the conditional file that SYSGEN produces. SYCND.MAC contains conditionals that, when assembled with system source files, enable options chosen during the SYSGEN session. Editing this file lets you, in effect, change responses to the dialogue without rerunning SYSGEN. However, DIGITAL recommends that, under most circumstances, you rerun SYSGEN to make changes. Editing SYCND.MAC requires great care. For the most part, only the most knowledgeable user should try to manipulate the conditional files.

This rule has several exceptions, which call for minor edits in special situations. Two such circumstances follow. List these edits, if appropriate, on the worksheet.

8.2.4.1 Multi-Terminal Support for United Kingdom — United Kingdom applications require RT–11 to disconnect a remote line immediately when the line hangs up. Normally, RT–11 delays disconnecting remote lines when it loses the carrier. It waits for a specific period of time to avoid dropping lines due to "noise." You can add a conditional to SYCND.MAC that adds support for immediate disconnect of DL11–E interfaces with modem support.

To include immediate disconnect support, insert the following line in the file SYCND.MAC:

```
U.K. = 1
```

8.2.4.2 Nonstandard CSR and Vector Support — You can change the conditionals in SYCND.MAC that define CSR and vector addresses. This procedure lets you include support for devices that are installed at nonstandard CSR and vector addresses without running the SYSGEN program and answering the dialogue questions. If your monitor is a standard monitor, if you did not include support for devices at nonstandard addresses when you answered the SYSGEN dialogue questions, or if you later add a device, insert a line in SYCND.MAC for each nonstandard CSR and vector you want to support. Use the following format:

```
dd$xxx = nnnnnn
```

dd   is the two-letter device mnemonic

xxx   must be replaced by one of the following:

VEC	is first controller's vector
CSR	is first controller's CSR address
VCx	is the vector for controller number x (for example VC2 is second controller's vector, VC3 is third controller's vector, and so on)
CSx	is CSR address for controller number x (for example CS2 is second controller's CSR address, CS3 is third controller's CSR address, and so on)

nnnnnn   is the six-digit octal address

NOTE

TS11 magtape uses VEC and CSR, then VC1 and CS1, then VC2 and CS2, and so on, instead of VEC and CSR, then VC2 and CS2, and so on above.

8.2.5 System Build Procedure

The system build procedure requires that you assemble and link your generated monitors and handlers but not the utility programs or other nonexecutive system components. The SYSGEN program produces indirect command files that issue all the commands to perform these assemblies and links. If your only mass storage device is RX01, RX02, or PDT–11 diskette, you may not have sufficient free storage space to use indirect command files, so that you will need to issue individual commands to perform the same procedures (see Chapter 10). The following sections describe both system build procedures so that you can plan the method you will use and so that you can answer the SYSGEN dialogue questions relative to the indirect command files.

NOTE

> Once you have created the conditional files, SYCND.MAC and SYSTBL.MAC, you can use them over and over without rerunning the SYSGEN program. To add *source* patches to your generated system, simply use the SLP utility to apply the patches. Reassemble those files and relink the system.

8.2.5.1 "Automatic" System Build — Questions 152, 153, and 154 of the SYSGEN dialogue ask you to identify the devices that the indirect command files should use for input, binary output, and map output during the assembly and link (system build) process. When you answer the dialogue questions and SYSGEN completes, SYSGEN has tailored the indirect command files so that they make logical assignments SRC:, BIN:, and MAP: according to your instructions. The command files then expect to find the correct files on each device. Therefore, before you answer questions 152, 153, and 154, you must understand which files you need and how much free space you need for the build procedure. Then you must plan the organization of files so that you can choose the appropriate devices for SRC:, BIN:, and MAP:. Figures 8–3 through 8–5 illustrate examples of such organization.

NOTE

> If you answer "no" to SYSGEN question 155, the indirect command files include deletion commands to *remove* the .OBJ files.

In most cases, you can use the indirect command files to build your system, but you may want to make minor alterations to the command files themselves. For example, you might need to change the devices on which individual source or binary files reside to optimize mass storage usage.

Study the following sections, which categorize requirements by device. Then organize files on volumes according to: 1) the hardware you have available, 2) the free storage you have, and 3) the storage requirements for building each component. If possible, you will want to use the largest and fastest storage devices included in your configuration.

System Device

When you perform the system build procedures, you must run RT–11 from a system device that contains at least the following:

monitor
system device handler

SWAP.SYS
handlers for SRC:, BIN:, and MAP:
MACRO
LINK
PIP
SYSMAC.SML

During the build procedure, the system device needs as many as 128 free blocks to accommodate the work file MACRO uses during system assembly.

Source Input Device

The source input device must contain the files SYCND.MAC and SYSTBL.MAC, which are the conditional files that result from a SYSGEN session. In addition, the source device must include the system source files to build the system you are generating. The system source files are included in the distribution kit, but you probably did not include them in your working system. You will need to copy the appropriate source files from your patched distribution backup volume(s). Establish which source files you need by using Table 8–1, and write the file names on the worksheet.

Table 8–1: Source Files Required for System Build

Source File	To Build Support For
BA.MAC	BATCH
BSTRAP.MAC	All monitors
CR.MAC	Card reader
CT.MAC	TA11 cassette
DD.MAC	DECtape II cartridge
DL.MAC	RL01/2 disk
DM.MAC	RK06/7 disk
DP.MAC	RP11/RPR02/RP03
DS.MAC	RJS03/4 disk
DT.MAC	DECtape
DX.MAC	RX01 diskette
DY.MAC	RX02 diskette
EDTGBL.MAC	All monitors
ELCOPY.MAC	Error logging
ELINIT.MAC	Error logging
ELTASK.MAC	Error logging
ERROUT.MAC	Error logging

(continued on next page)

Table 8–1: Source Files Required for System Build (Cont.)

Source File	To Build Support For
ERRTXT.MAC	Error logging
FB.MAC	FB monitor
FSM.MAC	File-structured magtape
KMON.MAC	All monitors
KMOVLY.MAC	All monitors
LP.MAC	Line printer
LS.MAC	Serial line printer
MTTEMT.MAC	Multi-terminal
MTTINT.MAC	Multi-terminal
NL.MAC	Null handler
PC.MAC	High-speed paper tape
PD.MAC	PDT–11 series
RF.MAC	RF11 disk
RK.MAC	RK05 disk
RMONSJ.MAC	SJ or BL monitor
RMONFB.MAC	FB or XM monitor
SJ.MAC	SJ or BL monitor
SYCND.MAC	All components
SYSTBL.MAC	All monitors
TJ.MAC	TJU16 magtape
TM.MAC	TM11 magtape
TS.MAC	TS11 magtape
TT.MAC	SJ or BL monitor
USR.MAC	All monitors
XM.MAC	XM monitor
XMSUBS.MAC	XM monitor

Binary Output Device

The binary output device receives the object files created by the assembly process. Identify all the components you must build — that is, all the monitors and device handlers for which your SYSGEN answers were intended. Then refer to Table 8–2 to establish how much free storage you will need on the binary output device to build each of these components.

Table 8–3 shows the size of the various devices, but keep in mind that once you initialize a device, some space is used for the boot and directory blocks. You can use the command DIRECTORY/FREE to ascertain how many free blocks remain on the device.

Table 8–2: Free Storage Required to Build Components

Component	Number of Blocks for Each Component
Monitor	300
SJ and FB device handler	10
XM device handler	10
Set of monitor object modules retained	200
Set of handler object modules retained	8

Table 8–3: Device Size

Device	Device Name	Size in Blocks
RX01 Diskette	DX	494
PDT–11/150 Diskette	PD	494
DECtape II Cartridge	DD	512
PDT–11/130 Cartridge	PD	512
DECtape	DT	578
RX02 Diskette	DY	988
RF11 Disk	RF	1024 per platter
RJS03/4 Disk	DS	1024/ 2048
RK05 Disk	RK	4800
RL01/2 Disk	DL	10210/ 20450
RP02/3 Disk	DP	40000/ 80000
RK06/7 Disk	DM	27102/ 53724

Map Output Device

The map output device receives the link maps that result from the link process. It is often useful to list the link maps on the terminal or a line printer, in which case you should specify TT: or LP: for the map output device.

If you specify the name of a block-replaceable device, such as a disk, you can send the link maps as files to that device. It is a good practice to send the link maps to a disk (instead of a terminal or line printer) so that you will be sure they are saved. DIGITAL requires that you include a link map and SYCND listing if you submit a Software Performance Report (SPR) for a monitor created by SYSGEN.

You can suppress the link maps altogether by sending the output to the null device (answer with the name NL: to SYSGEN question 155). However, DIGITAL strongly recommends saving the link maps.

Figure 8–3: All Files on One Disk for System Build

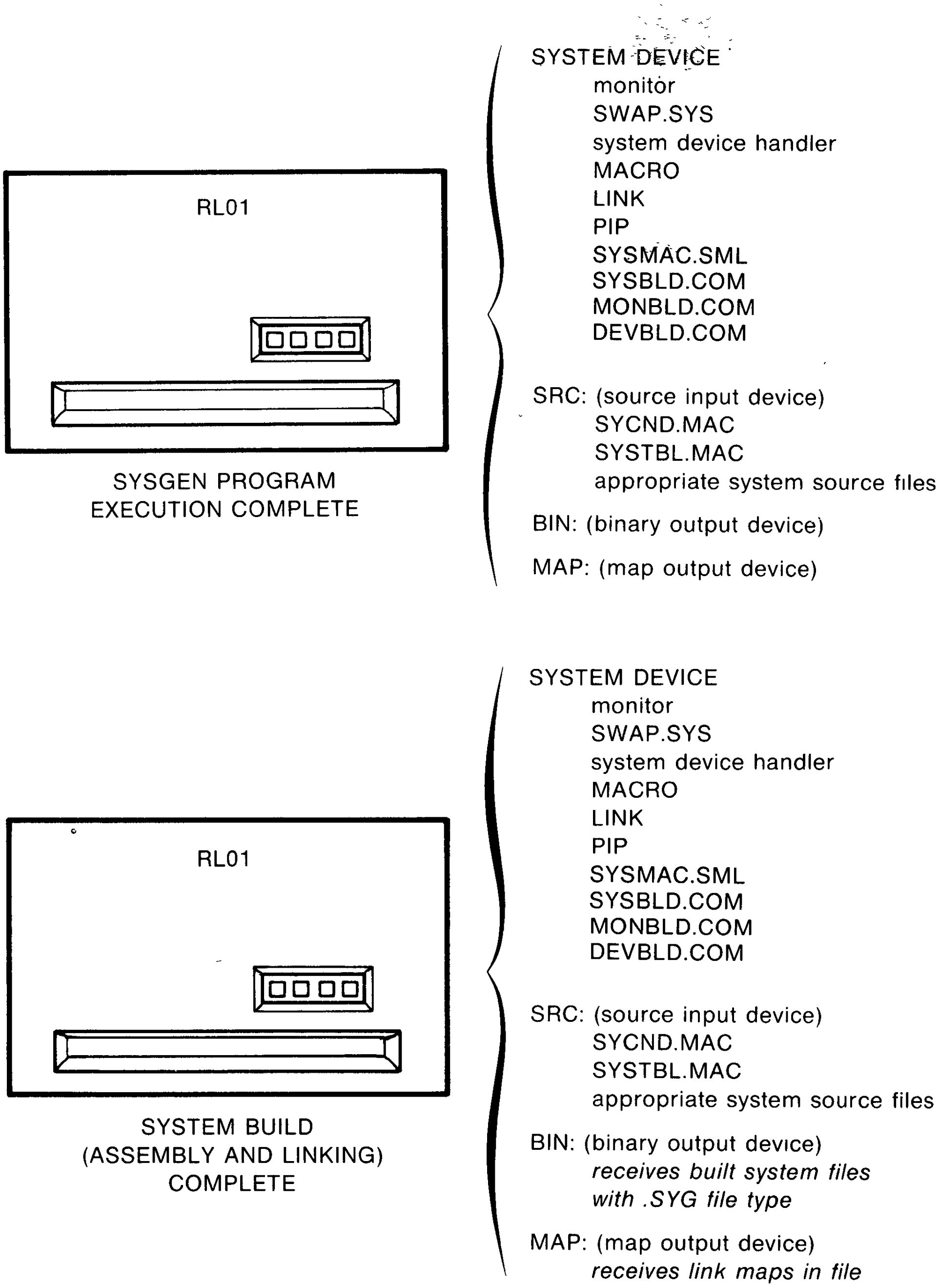


Figure 8–4: Source Files on Second Disk for System Build

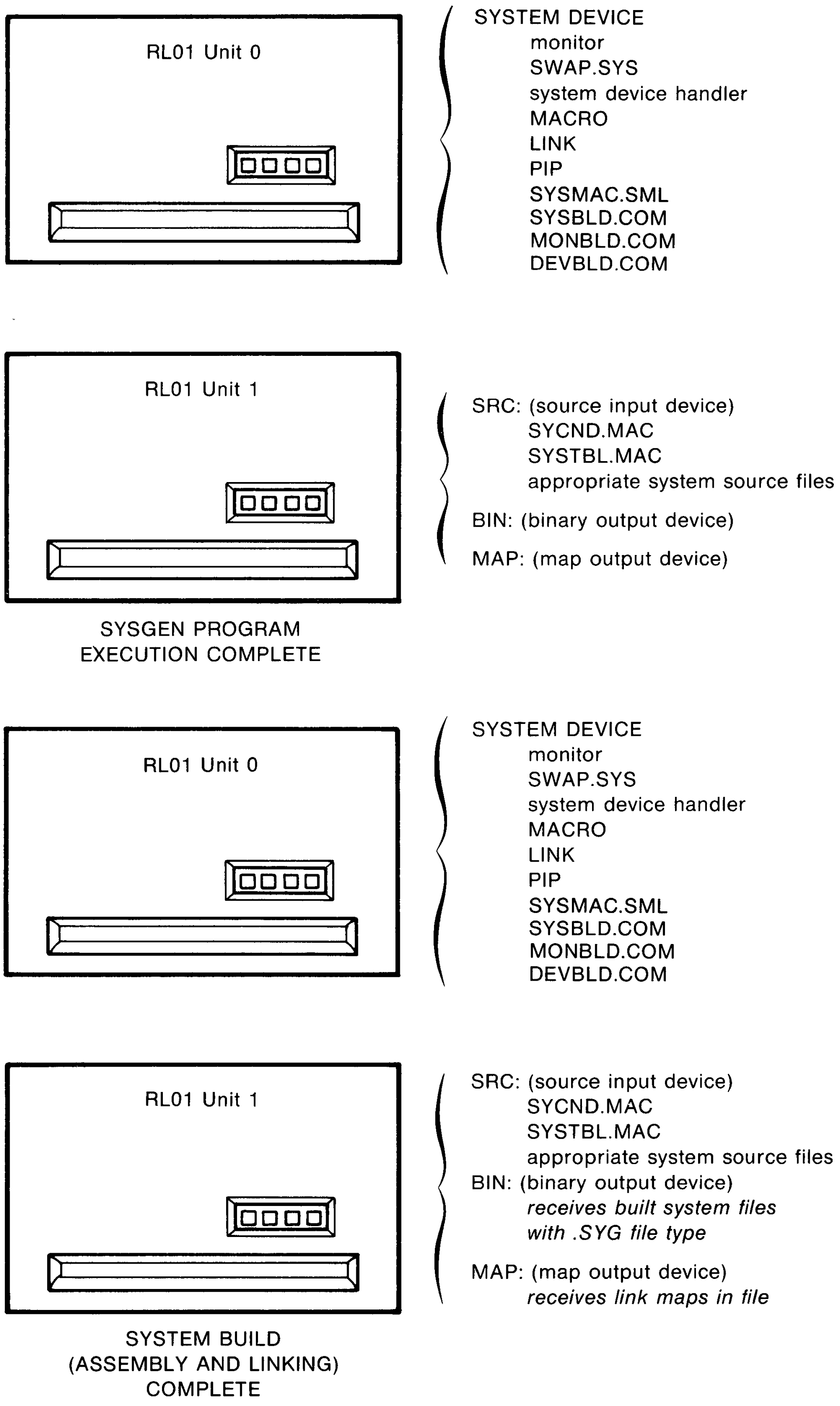
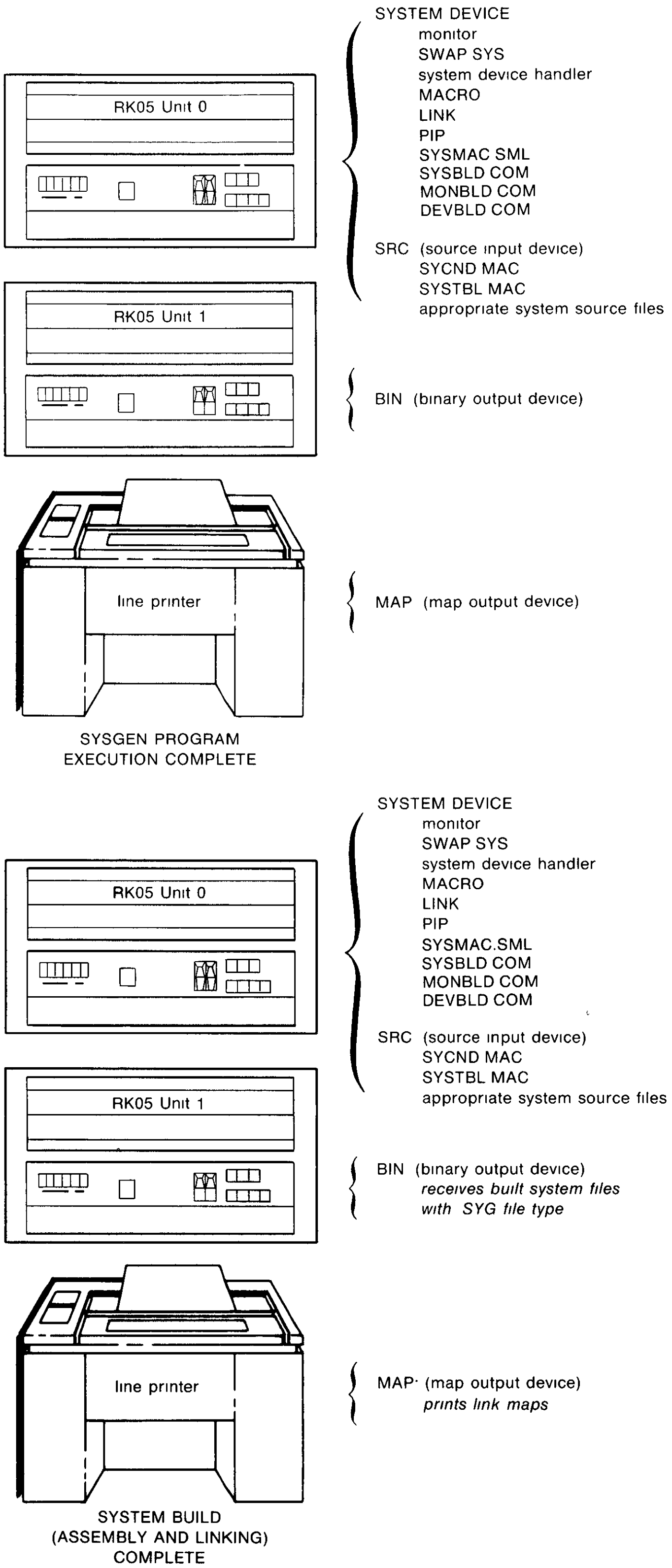


Figure 8–5: Second Disk, Line Printer Receive System Build Output



SYSGEN PROGRAM
EXECUTION COMPLETE

SYSTEM BUILD
(ASSEMBLY AND LINKING)
COMPLETE

8.2.5.2 "Manual" System Build — You may need to use a sequence of individual commands to build the system or one or more components of the system. Use the "manual" method if you do not have enough mass storage space to execute the entire command file or if you need to rebuild a particular component because an error occurred when you built it.

If you intend to perform the system generation process on an RX01 diskette system, a PDT–11/150 diskette system, or an RX02 diskette system, you will have to use the manual system build procedure.

In this case, specify devices included in your configuration when you answer questions 152, 153, and 154 of the SYSGEN dialogue. However, you need not plan the arrangement of files on SRC:, BIN:, and MAP:, since you will not be using the indirect command files MONBLD.COM, DEVBLD.COM, and SYSBLD.COM to build the system automatically. You will need the indirect files for information and you will need to arrange specific files on diskettes in order to perform the system build procedure.

If you are generating a system on one of these small systems, you cannot fit all the required files on two diskettes. You must arrange the files so that you can copy them to your system volume a few at a time and perform the component assemblies. Chapter 10 provides step-by-step instructions for generating special monitors and handlers on diskette systems. Those procedures require a special organization of files on diskettes for the assembly procedure. Study Chapter 10 if you need to use manual system build procedures.

8.3 Studying the SYSGEN Dialogue

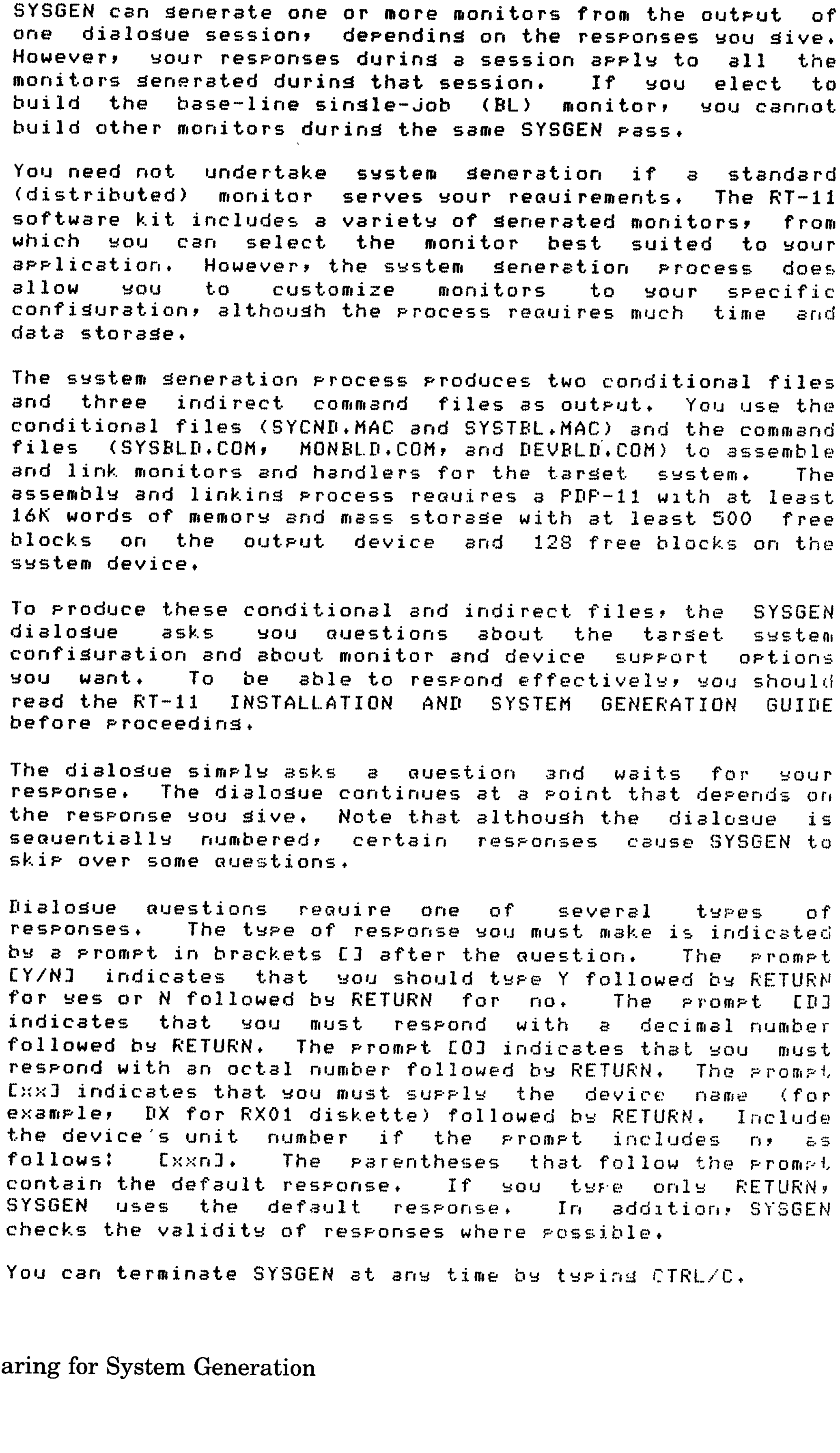
The SYSGEN dialogue is reproduced here so that you can study it. When you run the program SYSGEN.SAV, most of this dialogue prints on your terminal. Some dialogue questions appear only if you respond to a previous question in a specific way. Therefore, SYSGEN may not ask you all the questions shown in this section, which represents the complete list. The dialogue that prints on your terminal when you run SYSGEN appears below as actual computer output. Additional information is interspersed with the actual SYSGEN dialogue.

```
RT-11 SYSTEM GENERATION PROGRAM V04.00

The System  Generation  (SYSGEN)  program  functions  as  an
interactive  dialogue.   The  program  asks  you a series of
questions.  Your answers to those  questions  establish  the
characteristics  and  features  of  the  monitor(s)  and the
handlers  that  the  program  generates.   The  dialogue  is
available  to  you in two forms.  The short form simply asks
the questions.  The long form includes detailed explanations
of the questions.  If you choose the long form, you can type
CTRL/O to prevent  an  explanation  from  printing.   SYSGEN
stops  printing  the explanation, prints the question, waits
for your response, and resumes the long form of dialogue.

1.    Do you want the long form of the dialogue [Y/N] (Y)?
```

Respond yes to **question 1** if you want to read the explanatory text with each question. Note that you should not type CTRL/O while a question is printing. In this case, you would not be able to read the question, but SYSGEN would wait for a response.



```
SYSGEN can generate one or more monitors from the output of
one dialogue session, depending on the responses you give.
However, your responses during a session apply to all the
monitors generated during that session.  If you elect to
build the base-line single-job (BL) monitor, you cannot
build other monitors during the same SYSGEN pass.

You need not undertake system generation if a standard
(distributed) monitor serves your requirements.  The RT-11
software kit includes a variety of generated monitors, from
which you can select the monitor best suited to your
application.  However, the system generation process does
allow you to customize monitors to your specific
configuration, although the process requires much time and
data storage.

The system generation process produces two conditional files
and three indirect command files as output.  You use the
conditional files (SYCND.MAC and SYSTBL.MAC) and the command
files (SYSBLD.COM, MONBLD.COM, and DEVBLD.COM) to assemble
and link monitors and handlers for the target system.  The
assembly and linking process requires a PDP-11 with at least
16K words of memory and mass storage with at least 500 free
blocks on the output device and 128 free blocks on the
system device.

To produce these conditional and indirect files, the SYSGEN
dialogue asks you questions about the target system
configuration and about monitor and device support options
you want.  To be able to respond effectively, you should
read the RT-11 INSTALLATION AND SYSTEM GENERATION GUIDE
before proceeding.

The dialogue simply asks a question and waits for your
response.  The dialogue continues at a point that depends on
the response you give.  Note that although the dialogue is
sequentially numbered, certain responses cause SYSGEN to
skip over some questions.

Dialogue questions require one of several types of
responses.  The type of response you must make is indicated
by a prompt in brackets [] after the question.  The prompt
[Y/N] indicates that you should type Y followed by RETURN
for yes or N followed by RETURN for no.  The prompt [D]
indicates that you must respond with a decimal number
followed by RETURN.  The prompt [O] indicates that you must
respond with an octal number followed by RETURN.  The prompt
[xx] indicates that you must supply the device name (for
example, DX for RX01 diskette) followed by RETURN.  Include
the device's unit number if the prompt includes n, as
follows: [xxn].  The parentheses that follow the prompt
contain the default response.  If you type only RETURN,
SYSGEN uses the default response.  In addition, SYSGEN
checks the validity of responses where possible.

You can terminate SYSGEN at any time by typing CTRL/C.
```

```
MONITOR OPTIONS:
The standard monitors, distributed in the RT-11 software
kits, have been generated through the system generation
process. The first choice you must make in generating
custom monitors is the type of monitor or monitors you need.
The following questions let you choose the base-line
single-job monitor, the single-job monitor, the
foreground/background monitor, or the extended memory
monitor. Then, you can select additional options for the
monitors you generate.

BASE-LINE MONITOR:
The base-line single-job (BL) monitor is the smallest
monitor that lets you execute standard system functions. It
is intended for systems with only 8K words of memory, or for
larger systems that require the smallest possible monitor.
The BL and SJ monitors support the same set of programmed
requests--predefined assembler macro calls that generate an
EMT instruction, which the monitor interprets. (Programmed
requests make available to user programs monitor services
that RT-11 system programs use.) However, the BL monitor
limits device support and does not include some features of
the SJ monitor, such as BATCH and VT11 support. If you
answer YES, you cannot build any other monitors during this
SYSGEN session.

2.     Do you want the base-line single-job (BL)
       monitor [Y/N] (N)?
```

Respond yes to **question 2** if you want to generate a base-line single-job monitor. However, unless you are constrained by a very large application program and resultant space problems, you will probably want to select the single-job monitor. The SJ monitor offers significantly more features than the BL monitor in a relatively small size. Review Section 1.1.2 to compare features of the two monitors. Note that you can use only RX01 diskette, RK05 disk, DECtape, or PDT–11 volume as the system device with the BL monitor. Also, if you select the BL monitor, you cannot create any other monitors during the same SYSGEN pass. The default answer to this question is no.

```
SINGLE-JOB MONITOR:
The single-job (SJ) monitor provides an environment suitable
for developing simple FORTRAN or BASIC applications. The SJ
monitor includes many of the same features as the
foreground/background (FB) monitor--it supports all hardware
devices except the memory management unit, many programmed
requests, and all the utility programs. It offers the most
features in the smallest size. You can add features during
SYSGEN, but at the cost of increased monitor size.

3.     Do you want the single-job (SJ) monitor [Y/N] (Y)?
```

Respond yes to **question 3** if you want to generate a single-job monitor. You can build any of the other monitors, except BL, during the same SYSGEN pass. Review Section 1.1.2 to compare the advantages of the various monitors. In addition, Appendix A gives the SYSGEN answers that will duplicate the distributed monitors. The default response to this question is yes.

```
FOREGROUND/BACKGROUND MONITOR:
The foreground/background (FB) monitor offers an extended
set of features that aid real-time or data acquisition
applications. In addition to the foreground job capability,
this monitor supports serialized asynchronous I/O and
additional programmed requests (including .MRKT and .CMKT).

4.     Do you want the foreground/background (FB)
       monitor [Y/N] (Y)?
```

Respond yes to **question 4** if you want to generate a foreground/background monitor. You can build any of the other monitors, except BL, during the same SYSGEN pass. Review Section 1.1.2 to compare the advantages of the various monitors. The default response to this question is yes.

```
EXTENDED MEMORY MONITOR:
The extended memory (XM) monitor, which supports up to 124K
words of memory, is a version of the FB monitor. It
requires KT11 hardware (the memory management unit) and the
extended instruction set (EIS). The XM monitor supports
additional programmed requests that allow you to extend a
program's logical addressing space. However, the XM monitor
is larger than the FB monitor and requires that the user
service routine (USR) and the device handlers be resident.

5.     Do you want the extended memory (XM)
       monitor [Y/N] (N)?
```

Respond yes to **question 5** if you want to generate an extended memory monitor. Review Section 1.1.2 to compare the advantages of the various monitors. The default response to this question is no.

You must respond yes to at least one of questions 2 through 5. Each SYSGEN run must build at least one monitor. SYSGEN builds the monitor and handlers according to the same parameters. Consequently, you cannot build device support without building a monitor. Nor can you *rebuild* device support alone. If you respond no to questions 2 through 5, SYSGEN issues an error message and terminates the SYSGEN session. Rerun the SYSGEN program in this case.

Note also that when you select certain options, SYSGEN automatically enables other options, even if you respond no to the questions. For example, if you respond no to SJ timer support and device time-out support but yes to multi-terminal support, SYSGEN enables SJ timer support and device time-out support anyway.

```
You can select certain additional monitor options, as
follow:

SJ TIMER SUPPORT OPTION:
The SJ monitor normally does not include timer support. You
can include this support, which enables you to use the .MRKT
(mark time) and .CMKT (cancel mark time) programmed
requests.

6.     Do you want timer support in the SJ monitor [Y/N] (N)?
```

Respond yes to **question 6** if you want to use the .MRKT and .CMKT programmed requests with the SJ monitor. The FB and XM monitors support these programmed requests, which provide timer capability. However, you must specifically select this support for SJ since it adds approximately 360 (decimal) words to the resident monitor. Timer support reduces response time slightly because of the additional interrupt-level overhead. If your application requires timer support, you must select it. Note also that the SJ monitor increments the date at midnight only if you select this support. RT–11 does not require it. The default response to this question is no.

```
DEVICE TIME-OUT SUPPORT OPTION:
The device time-out option permits device handlers to  issue
.MRKT  (mark  time)  programmed requests.  RT-11 handlers do
not currently use this feature, but DECNET support  requires
it.

7.     Do you want device time-out support [Y/N] (N)?
```

Respond yes to **question 7** if you are a DECnet user or if your application uses device time-out support. RT–11 does not use this support. This support adds approximately 40 (decimal) words to the resident FB and XM monitors; it adds approximately 360 (decimal) words to the resident SJ monitor, since enabling this support automatically enables SJ timer support also. The default response to this question is no.

```
ERROR MESSAGE ON SYSTEM I/O ERRORS OPTION:
The SJ monitor normally halts if a fatal  system  I/O  error
occurs.   You  can  replace  this  halt  with a system error
message.  DIGITAL  highly  recommends  this  option  if  the
system  will  be  used  by  anyone  but the most experienced
individuals.

8.     Do you want an error message on system I/O
       errors [Y/N] (Y)?
```

Respond yes to **question 8** if you want an error message instead of a system halt when the SJ monitor detects a fatal I/O error. Although this option adds approximately 30 (decimal) words to the resident monitor, DIGITAL strongly recommends this option for all but the most space-conscious applications. The confusion this option saves is well worth the small amount of additional memory it requires. The FB and XM monitors include this support. The default response to this question is yes.

```
SYSTEM JOB OPTION:
The system job option is a conditional assembly  of  the  FB
and  XM  monitors  that  allows  you  to  run  up  to  eight
simultaneously active jobs.  The error logger and the  QUEUE
program  can  be  either foreground jobs or system jobs.  If
you want to run both simultaneously, or if you want  to  run
either  one  along with a foreground job you need the system
job feature.

9.     Do you want system job support [Y/N] (N)?
```

Respond yes to **question 9** if you want system job support. You need this support if you use the error logger or the device queue program (QUEUE) along with a foreground job, or if you use both the error logger and QUEUE. You can also use system job support to create application-specific system jobs. However, DIGITAL *supports* only the error logger and the device queue program as system jobs. This support adds approximately 300 (decimal) words to the resident monitor. The default response to this question is no.

```
SAVE/SET MAIN-LINE PC AND PS OPTION:
The save/set main-line PC and PS  option is a  conditionally
assembled programmed request for the FB and XM monitors. The
.SPCPS request changes the flow of control of main-line code
by   saving  the   main-line code   PC   and   PS   and   changing
the main-line PC to a new value.  This request may be useful
in multi-user applications to control switching among users.

10.    Do you want to use the .SPCPS request [Y/N] (N)?
```

Respond yes to **question 10** if you want to use the .SPCPS request. Refer to the *RT–11 Programmer's Reference Manual* for more information about this programmed request. The default response to this question is no.

```
IDLE LOOP LIGHT PATTERN OPTION:
This option causes the FB and XM  monitors'  scheduler  idle
loop  to  display  a  moving  light  pattern  in the display
register of any PDP-11/45, 11/50, 11/55, or 11/70 processor.

11.    Do you want idle loop light pattern [Y/N] (N)?
```

Respond yes to **question 11** if you want this support and have the appropriate hardware. This option, which applies only to the FB and XM monitors, shows you visually how busy the central processor is. However, it adds approximately 25 (decimal) words to the resident monitor. SYSGEN asks this question only if you select the FB or XM monitor. The default response to this question is no.

```
MULTI-TERMINAL SUPPORT OPTION:
The multi-terminal option lets you  use  special  programmed
requests  to  do  I/O  to  more  than  one  terminal.  RT-11
normally supports only one terminal (interfaced through  the
console KL11 or DL11) which is shared by both background and
foreground jobs.  You can select multi-terminal support  for
up  to  16 terminals interfaced through a choice of DL11 and
DZ11 interfaces.  Subsequent questions  will  establish  the
number and type of interfaces.

12.    Do you want multi-terminal support [Y/N] (N)?
```

Respond yes to **question 12** if you want multi-terminal support and are generating an SJ, FB, or XM monitor. This option selects monitor support for more than one terminal (as many as 16). However, if the application itself provides support for additional terminals and if you require only console support in the monitor, you should not select this support. MU BASIC–11 requires that you respond yes to this question.

You need not select multi-terminal support to use a hard copy terminal as a line printer. You need only the serial line printer handler, LS.SYS, which is included in the software kit.

Multi-terminal support adds considerably to the space requirements of the resident monitor. It adds a minimum of 750 (decimal) words for the first additional terminal and approximately 100 (decimal) words for each additional terminal. However, the exact number of words added depends on the interfaces, the configuration, and the other options you select. Also, note that when you select multi-terminal support, SJ timer support and device time-out support features are automatically enabled.

This option applies only to the SJ, FB, and XM monitors, and SYSGEN asks this question only if you select one of these monitors. The default response to this question is no.

```
ASYNCHRONOUS TERMINAL STATUS OPTION:
The asynchronous terminal status option provides a program
with the updated status of a terminal and modem. When a
program attaches a terminal, the program can supply a status
word that the monitor updates as changes in the terminal
status (double CTRL/C, input available, output buffer empty,
carrier present) occur. This support is required for MU
BASIC and CTS-300 applications.

13.     Do you want asynchronous terminal status [Y/N] (Y)?
```

Respond yes to **question 13** if you are an MU BASIC or CTS–300 user or if your application uses asynchronous terminal status information. If you are in doubt about this option, respond yes. This support applies only to multi-terminal applications, so SYSGEN asks you this question only if you respond yes to question 12. The default response to this question is yes.

```
MULTI-TERMINAL TIME-OUT OPTION:
Multi-terminal time-out support causes the monitor to reset
(at regular intervals) any terminal that may have gone
off-line. This action helps to minimize the impact of
static and similar problems. DIGITAL recommends selecting
this option if your application requires maximum terminal
availability.

14.     Do you want multi-terminal time-out support [Y/N] (Y)?
```

Respond yes to **question 14** for maximum terminal availability. If you are an MU BASIC–11 user, be sure to select this support.

This support applies only to multi-terminal applications, so SYSGEN asks you this question only if you respond yes to question 12. DIGITAL recommends that you select this option unless you have space problems. The default response to this question is yes.

```
RING BUFFER SIZE OPTION:
The RT-11 terminal service requires a set of input and
output ring buffers for each terminal supported. The input
ring is a buffer in the monitor that holds the characters
that you type at a terminal until a program requests them.
The output ring is a buffer in the monitor that holds
characters until the terminal can print them (a program can
output characters faster than a terminal can print them).
```

```
The input ring buffer default size is 134 characters. The
output ring buffer default size is 40 characters. However,
you can select the size of buffers you need. You may want
to change the input ring size, in particular, since the size
you need depends on the terminal's width and the amount of
type-ahead you expect.

15.    Size of the output buffers in characters [D] (40)?
```

Respond to **question 15** with the number (decimal) of characters that you want the monitor to hold in the output ring buffers. Each character adds n bytes to the resident monitor size, where n is the total number of.terminal lines. Note that you must count the carriage return/line feed combination as two characters. The range for valid responses is from 10 to 134. The default response to this question is 40.

```
16.    Size of the input buffers in characters [D] (134)?
```

Respond to **question 16** with the number (decimal) of characters that you want the monitor to hold in the input ring buffers. If you specify an odd number of characters, SYSGEN rounds the number down. Each character adds n bytes to the resident monitor size, where n is the total number of terminal lines.

You should make the input ring buffers large enough to hold at least one line of input. If the largest line accepted by an application is greater than 134 characters, you can adjust the size of the input buffers. The minimum acceptable input buffer size is 74 characters. If you specify less than 74, an error occurs at monitor assembly time. The monitors require 82-character buffers for efficient command operation. Buffers larger than 82 characters are useful if you anticipate using the type-ahead feature.

The RT–11 keyboard monitor and the .GTLIN programmed request accept up to 80 characters. However, if your program uses the VT100 132-character mode, you should select support for an input ring buffer of 134 characters.

The range for valid responses is from 74 to 254. The default response to this question is 134.

```
MONTH ROLLOVER OPTION:
When you run a system continuously, over a long period of
time, you normally have to reset the date and time at the
beginning of each month. You can select a feature that
automatically sets the correct date and time. However, this
option greatly increases the size of the monitor.

17.    Do you want end of month and year date
       rollover [Y/N] (N)?
```

Respond yes to **question 17** if you want the date automatically reset at the beginning of a month or year and if space is not a serious concern. This option adds approximately 50 (decimal) words to the resident monitor. In addition, SJ timer support is automatically enabled if you select this option. The default response to this question is no.

```
KEYBOARD MONITOR COMMANDS OPTION:
The keyboard monitor commands option lets you choose the
keyboard monitor commands your generated monitor will
support. You will probably find all of the commands useful.
However, you can reduce KMON size and assembly time if you
select support for a subset of the available commands. You
have a choice of three subsets (or any combination of
subsets). The three command subsets you can choose are:
the UTILITY PROGRAM COMMANDS, the LANGUAGE COMMANDS, and the
MINIMAL COMMANDS. The RT-11 INSTALLATION AND SYSTEM
GENERATION GUIDE lists the specific commands included in
each subset. If you do not choose the complete set of
commands or any of the subsets, you will be able to use only
the RUN command.
```

```
18.    Do you want all the keyboard monitor
       commands [Y/N] (Y)?
```

Respond yes to **question 18** if you want support for all the keyboard monitor commands. All the commands available with RT–11 are in the following list. If you choose the complete set of keyboard commands, the disk (or other device) file image of the SJ monitor is increased by 14 blocks and the image of the FB or XM monitor by 18 blocks. If you choose one of the subsets, the monitor disk file images will be smaller. The default response to this question is yes.

NOTE

> If you want just one or several commands, use the conditionals in Appendix F and edit SYCND.MAC.

APL
ASSIGN
B
BASIC
BOOT
CLOSE
COMPILE
COPY
CREATE
D
DATE
DEASSIGN
DELETE
DIBOL
DIFFERENCES
DIRECTORY
DUMP
E
EDIT
EXECUTE
FOCAL
FORMAT
FORTRAN
FRUN
GET
GT
HELP
INITIALIZE
INSTALL
LIBRARY
LINK
LOAD
MACRO
PRINT
R
REENTER
REMOVE
RENAME
RESET
RESUME
RUN
SAVE
SET
SHOW
SQUEEZE
SRUN
START
SUSPEND
TIME
TYPE
UNLOAD

```
19.    Do you want the UTILITY PROGRAM subset of keyboard
       monitor commands [Y/N] (Y)?
```

Respond yes to **question 19** if you want support for the UTILITY PROGRAM subset of monitor commands. The following commands are included in this subset. Choosing this subset increases the size of the disk (or other device) image of the monitor file by only 6 blocks (rather than the 14 to 18 blocks for all the commands). Note that you can select more than one subset; SYSGEN asks you about the other subsets. However, SYSGEN does not ask this question if you respond yes to question 18, which asks whether you want *all* the commands. Note also that the R and RUN commands cannot be removed; they are included in all the sets of commands. The default response to this question is yes.

BOOT
COPY
CREATE
DELETE
DIFFERENCES
DIRECTORY
DUMP
EDIT
FORMAT
INITIALIZE
PRINT
RENAME
SHOW
SQUEEZE
TYPE

```
20.   Do you want the LANGUAGE subset of keyboard monitor
      commands [Y/N] (Y)?
```

Respond yes to **question 20** if you want support for the LANGUAGE subset of monitor commands. The following commands are included in this subset. Choosing this subset increases the size of the disk (or other device) image of the monitor file by only 4 blocks (rather than the 14 to 18 blocks for all the commands). Note that you can select more than one subset. SYSGEN does not ask this question if you respond yes to question 18. The default response to this question is yes.

APL
BASIC
COMPILE
DIBOL
EXECUTE
FOCAL
FORTRAN
LIBRARY
LINK
MACRO

```
21.   Do you want the MINIMAL subset of keyboard monitor
      commands [Y/N] (Y)?
```

Respond yes to **question 21** if you want support for the MINIMAL subset of keyboard monitor commands. The execution code for all the commands in the MINIMAL subset is resident in KMON. If you select this option but do not select FB or XM monitor support, SYSGEN does not include the commands FRUN, RESUME, and SUSPEND. If you select this subset but do not select support for VT–11 graphics display, SYSGEN does not include the GT command. If you do not select system job support, SYSGEN does not include the SRUN command. Note, however, that you can select more than one subset. Choosing this subset increases the size of the disk (or other device) image of the SJ monitor file by only 6 blocks and the FB or XM monitor file by only 10 blocks (rather than the 14 to 18 blocks for all the commands). The following commands are included in the MINIMAL subset. SYSGEN does not ask this question if you respond yes to question 18. The default response to this question is yes.

ASSIGN
B
CLOSE
D
DATE
DEASSIGN
E
FRUN

GET
GT
HELP
INSTALL
LOAD
R
REENTER
REMOVE
RESET
RESUME
RUN
SAVE
SET
SRUN
START
SUSPEND
TIME
UNLOAD

```
50 HZ CLOCK OPTION:
A line clock generates periodic interrupts to allow the
system to keep track of the time.  The number of ticks per
second depends on the power line frequency, 60 Hz or 50 Hz.
RT-11 assumes a 60 Hz line frequency, but you can select
support for a 50 Hz clock.  The 50 Hz frequency has
specialized uses and is the common frequency in Europe.

22.    Do you want the optional 50 Hz clock
       support [Y/N] (N)?
```

Respond yes to **question 22** if your computer's line frequency is 50 Hz. The default response to this question is no.

```
PROGRAMMABLE CLOCK AS SYSTEM CLOCK OPTION:
RT-11 normally uses a line clock for the system clock.  You
can substitute the KW11-P programmable clock as the system
clock, but the KW11-P will not then be available for program
use.  The programmable clock normally allows you to program
interrupts at preset intervals.

23.    Do you want to use the KW11-P clock as the system
       clock [Y/N] (N)?
```

Respond yes to **question 23** if your configuration includes the KW11–P clock and you want to use it as the system clock. The monitor uses the system clock for certain functions (time-of-day calculations, timer requests, and FB and XM scheduling). If the system clock is the programmable clock, your application cannot access the programmable clock or change its rate without affecting monitor function. Therefore, DIGITAL recommends that you use a line clock, if available, for the system clock. The default response to this question is no.

```
STARTUP INDIRECT COMMAND FILE OPTION:
This option causes the bootstrap to execute an indirect
command file (with the name STARTx.COM, where x identifies
the monitor) when starting the system.  An indirect command
file contains monitor commands that the monitor processes
(in the order in which they appear) when the file is
executed.  A startup indirect command file is particularly
useful for setting up initial conditions (for example,
assigning the default device to the data device, installing
a device into the system tables when the device was not
originally built into the system, or running a specific
program).

24.    Do you want the startup indirect file [Y/N] (Y)?
```

Respond yes to **question 24** if you want the monitor bootstrap to execute a startup indirect file every time you boot the system. If you select this support, the bootstrap looks for a file STARTx.COM to execute (where x is S, F, or X for single-job, foreground/background, or extended memory monitor). You create this file, with commands to perform whatever operations you choose. If no STARTx.COM file resides on a volume, the monitor prints an error message when you boot that volume. This option requires no additional resident monitor space and the default response to this question is yes.

```
FLOATING POINT OPTION:
If your configuration  includes floating point hardware  and
your application requires  this feature, you  should  enable
this  option.  Floating point hardware is available for many
PDP-11 processors.  This  is  especially useful for FORTRAN,
BASIC,  and  APL  users  whose  applications  perform   data
manipulations.

25.    Do you want floating point support [Y/N] (N)?
```

Respond yes to **question 25** if your configuration includes the appropriate hardware and your application uses floating point manipulations. If you try to perform floating point manipulations but have not selected this support, system failure results.

If you select this support, the monitor intercepts all floating point traps. If your program has not set up an exception handler, the monitor prints an informative message and aborts the program. If your program has set up an exception handler, that routine is entered with the FPU status (if appropriate) on the stack. The monitor makes sure that the correct job's context is set up.

In addition, if you select floating point support, the .SFPA programmed request (refer the the *RT–11 Programmer's Reference Manual*) is enabled. If you do not choose floating point support, none of the functions of .SFPA is performed; for example, the floating point registers are never context switched.

If you do not include floating point support, all programs that use the floating point instructions must set up and .PROTECT or .CNTXSW the vector. FORTRAN requires monitor floating point support if you use one of the floating point libraries.

Floating point support adds approximately 200 (decimal) words to the resident FB monitor and approximately 300 (decimal) words to the resident XM monitor. The default response to this question is no.

```
MEMORY PARITY SUPPORT OPTION:
If your configuration includes memory parity  hardware,  you
should  select  this  option.  Memory parity hardware checks
for memory errors and this option enables RT-11 support  for
the  hardware.  RT-11  issues an error message and supplies
the location of access when a memory error occurs.   If  you
have  this  hardware  but  do  not  enable this support, the
```

```
system halts when memory errors occur.  If you select  error
logging  as  well as memory parity support, the error logger
logs parity errors as well as device errors.

26.    Do you want memory parity support [Y/N] (N)?
```

Respond yes to **question 26** if your configuration includes memory parity hardware. Memory parity support adds approximately 8 (decimal) words to the resident SJ monitor and approximately 25 (decimal) words to the resident FB and XM monitors. If you have this hardware, you need RT–11 memory parity support both to make use of the hardware's capabilities and to avoid unwanted system halts. The default response to this question is no.

```
POWER FAILURE MESSAGE OPTION:
The monitor normally halts on power recovery startup after a
power  failure.  You  can elect to have the monitor print a
message explaining  that  power  failure  caused  the  halt.
However,  you  must  not  select  this  option  if  you have
semiconductor memory, which is volatile.

27.    Do you want power failure messages [Y/N] (Y)?
```

Respond yes to **question 27** if you want an error message on startup after system halt to report the cause of the halt. Select this option only if your hardware configuration includes core memory or battery back-up, not semiconductor memory (which is volatile). This option adds approximately 90 (decimal) words to the resident monitor. However, DIGITAL strongly recommends this option since it quickly identifies the cause of failure. Frequently, system halts are caused by power failures too brief to detect visually. The default response to this question is yes.

```
BATCH SUPPORT OPTION:
You can select support for the BATCH job  control  language,
which  allows RT-11 to operate unattended.  Once you prepare
a BATCH stream, you can leave it for an  operator  to  start
and  run,  and  the  BATCH  stream  will execute programs or
monitor  commands  without  your   intervention.   Indirect
command  file support, which offers similar capabilities, is
separately available in all monitors.  You need  not  select
BATCH support to obtain indirect command file support.

28.    Do you want BATCH support [Y/N] (N)?
```

Respond yes to **question 28** if you want BATCH support. RT–11 BATCH support is similar to indirect command file support, but it offers certain advantages. BATCH produces a log file, allows job-stream programmability, and permits the operator to interact with the job during execution. Unless you need the features not provided by indirect command file support, you need not select this support. The base-line single-job monitor does not support BATCH processing, so SYSGEN does not ask you this question if you select the BL monitor. The default response to this question is no.

```
ERROR LOGGING OPTION:
The error logging option  creates  the  error  logging  (EL)
system  job,  and  incorporates error logging support in the
device  handlers.  Error  logging  reports  device,  memory
```

```
parity (if selected), and memory cache errors. When error
logging is enabled, most device handlers call EL on each
successful transfer and on each error. The EL job retrieves
information from the handlers that is later available to you
in summary report format.

29.   Do you want error logging [Y/N] (N)?
```

Respond yes to **question 29** if you want to use the error logging system job. This option adds approximately 2000 (decimal) words to the resident monitor and 50 (decimal) words to each device handler you generate. However, error logging is useful in monitoring system reliability. The default response to this question is no.

```
NUMBER OF UNITS SUPPORTED BY ERROR LOGGING OPTION:
The error logger can handle up to 34 individual device
units. You can conserve space by reducing the number of
units the logger can handle to the specific number of
supported units in the target configuration.

30.   How many device units does error log job
      support [D] (10)?
```

Respond to **question 30** with the number (decimal) of supported device units in the system you are generating. For example, if the target system has three RK05 drives and a dual diskette drive, the response is 5. Each unit adds seven words to the error logger. The range for valid responses is from 1 to 34. SYSGEN asks this question only if you respond yes to question 30. The default response to this question is 10.

```
PERIPHERAL DEVICE OPTIONS:
The device options let you select the peripheral devices
that the generated system will support. By selecting
support for a specific device, you make the device known to
the monitor's device tables, and you cause SYSGEN to
generate a device handler (named xx.SYS, where xx is the
physical device name) for the device. If you do not select
a specific device at this time, you will have to create the
device handler separately and you will have to use the
monitor INSTALL command before you can access the device.

Since the system device handler is not a part of the monitor
image, you need not select a system device during system
generation. Any device in the following list of valid
system devices can serve as your system device. To serve as
a system device, a device needs a monitor file, the
appropriate device handler, the file SWAP.SYS, and the
bootstrap. However, you should be sure to select support
for the device that will serve as your system device, so
that SYSGEN will generate a device handler for it and
permanently install support for it in the monitor. If you
have selected the extended memory monitor, SYSGEN will
generate special device handlers (named xxX.SYS, where xx is
the physical device name).

Valid RT-11 system devices are:
        DX   RX01 Single-Density Flexible Diskette
        PD   PDT-11 Intelligent Terminal
        DT   DECtape
        RK   RK05 Cartridge Disk
```

```
DX   RX01 Single-Density Flexible Diskette
DY   RX02 Double-Density Flexible Diskette
DD   TU58 DECtape II
PD   PDT-11 Intelligent Terminal
DT   DECtape
RF   RF11 Disk
DS   RJS03/4 Massbus Fixed-head Disk
RK   RK05 Cartridge Disk
DL   RL01 Cartridge Disk
DP   RP11 Disk Pack
DM   RK06/RK07 Cartridge Disk
```

If you have selected support for the base-line single-job monitor, SYSGEN prints the short list of valid system devices. If you have selected another monitor, SYSGEN prints the long list.

When you respond to the peripheral device options, select support for only the devices that are included in the target system.

Keep in mind also that RX01 diskettes and PDT–11/150 diskettes look alike but use different hardware and that TU58 DECtape II cartridges and PDT–11/130 cartridges look alike but use different hardware. If you have either PDT–11 device, select PDT–11 support (question 50), not RX01 or TU58.

```
31.   Do you want RX11/RX01 single-density flexible
      diskette support [Y/N] (N)?
```

Respond yes to **question 31** if your target configuration includes an RX11/RX01 single-density flexible diskette subsystem. The default response to this question is no.

```
The RX01 single-density flexible diskette subsystem supports
only two drives for each controller. If your system
contains four diskette drives, a second controller governs
the third and fourth units, and you must select RT-11
support for it.
```

```
32.   Do you want support for a second RX11
      controller [Y/N] (N)?
```

Respond yes to **question 32** if your target configuration includes four RX01 diskette drives. Note that you can boot RT–11 only from Unit 0 or Unit 1. SYSGEN asks this question only if you respond yes to question 31. The default response to this question is no.

```
33.   What is the CSR address for the first RX11
      controller [O] (177170)?
```

Respond to **question 33** with the address (octal) of the control register for the first RX11 controller. The range for valid responses is from 160000 to 177570. The Field Service Engineer who installs your hardware system is responsible for supplying you with a written record of addresses at which he or she installs each device. The default response to this question is 177170.

```
34.   What is the vector address for the first
      RX11 controller [O] (264)?
```

Respond to **question 34** with the vector address (octal) for the first RX11. The range for valid responses is from 100 to 474. The default response to this question is 264.

```
35.    What is the CSR address for the second RX11
       controller [O] (177174)?
```

Respond to **question 35** with the address (octal) of the control register for the second RX11 controller. The range for valid responses is from 160000 to 177570. The default response to this question is 177174.

```
36.    What is the vector address for the second
       RX11 controller [O] (270)?
```

Respond to **question 36** with the vector address (octal) for the second RX11. The range for valid responses is from 100 to 474. The default response to this question is 270.

```
37.    Do you want RX211/RX02 double-density flexible
       diskette support [Y/N] (N)?
```

Respond yes to **question 37** if your target configuration includes an RX211/RX02 double-density flexible diskette subsystem. The default response to this question is no.

```
The RX02 double-density diskette subsystem supports only two
drives for each controller. If your system contains four
drives, a second controller governs the third and fourth
units, and you must select RT-11 support for it.
```

```
38.    Do you want support for a second RX02
       controller [Y/N] (N)?
```

Respond yes to **question 38** if your target system includes four RX02 diskette drives. Note that you can boot RT–11 only from Unit 0 or Unit 1. SYSGEN asks this question only if you respond yes to question 37. The default response to this question is no.

```
The RX02 diskette supports both single and double densities
by default. If you select double-density only support, you
can slightly improve the performance and reduce the size of
the device handler.
```

```
39.    Do you want RX02 double density only
       support [Y/N] (N)?
```

Respond yes to **question 39** if you plan to use only double-density diskettes. The default response to this question is no.

```
40.    What is the CSR address for the first RX02
       controller [O] (177170)?
```

Respond to **question 40** with the address (octal) of the control register for the first RX211 controller. The range for valid responses is from 160000 to 177570. The Field Service Engineer who installs your hardware system is responsible for supplying you with a written record of addresses at which he or she installs each device. The default response to this question is 177170.

```
41.    What is the vector address for the first
       RX02 controller [O] (264)?
```

Respond to **question 41** with the vector address (octal) of the first RX211. The range for valid responses is from 100 to 474. The default response to this question is 264.

```
42.   What is the CSR address for the second RX02
      controller [O] (177150)?
```

Respond to **question 42** with the address (octal) of the control register for the second RX211 controller. The range for valid responses is from 160000 to 177570. The default response to this question is 177150.

```
43.   What is the vector address for the second RX02
      controller [O] (270)?
```

Respond to **question 43** with the vector address (octal) of the second RX211. The range for valid responses is from 100 to 474. The default response to this question is 270.

```
44.   Do you want TU58 DECtape II cartridge support [Y/N] (N
```

Respond yes to **question 44** if your target configuration includes a TU58 DECtape II cartridge subsystem. The default response to this question is no.

```
The TU58 DECtape II cartridge subsystem  supports  only  two
drives  for  each  controller.  If your system contains four
drives, a second controller governs  the  third  and  fourth
units, and you must select RT-11 support for it.
```

```
45.   Do you want support for a second DECtape II
      controller [Y/N] (N)?
```

Respond yes to **question 45** if your target configuration includes four DECtape II drives. Note that you can boot RT–11 only from Unit 0 or Unit 1. SYSGEN asks this question only if you respond yes to question 44. The default response to this question is no.

```
46.   What is the CSR address for the first DECtape II
      controller [O] (176500)?
```

Respond to **question 46** with the address (octal) of the control register for the first DECtape II controller. The range for valid responses is from 160000 to 177570. The Field Service Engineer who installs your hardware system is responsible for supplying you with a written record of addresses at which he or she installs each device. The default response to this question is 176500.

```
47.   What is the vector address for the first DECtape II
      controller [O] (300)?
```

Respond to **question 47** with the vector address (octal) for the first DECtape II. The range for valid responses is from 100 to 474. The default response to this question is 300.

```
48.   What is the CSR address for the second DECtape II
      controller [O] (176510)?
```

Respond to **question 48** with the address (octal) of the control register for the second DECtape II controller. The range for valid responses is from 160000 to 177570. The default response to this question is 176510.

```
49.   What is the vector address for the second DECtape II
      controller [O] (310)?
```

Respond to **question 49** with the vector address (octal) for the second DECtape II. The range for valid responses is from 100 to 474. The default response to this question is 310.

```
50.    Do you want support for the PDT-11 series [Y/N] (N)?
```

Respond yes to **question 50** if your target configuration includes a PDT–11 Intelligent Terminal. The PDT–11 Intelligent Terminal contains a processor, so if you have an intelligent terminal, you do not have a PDP–11 processor. The default response to this question is no.

```
51.    Do you want TC11 DECtape support [Y/N] (N)?
```

Respond yes to **question 51** if your target configuration includes a TC11/TU56 DECtape subsystem. The default response to this question is no.

```
52.    Do you want RF11 fixed-head disk support [Y/N] (N)?
```

Respond yes to **question 52** if your target configuration includes an RF11/RS11 disk subsystem. The default response to this question is no.

```
53.    How many disk platters are installed on the
       RF11 controller [D] (1)?
```

Respond to **question 53** with the number (decimal) of RS11 platters included in your RF11 subsystem. SYSGEN asks this question only if you respond yes to question 52. The range for valid responses is from 1 to 8 and the default response is 1.

```
54.    Do you want RJS03 or RJS04 disk support [Y/N] (N)?
```

Respond yes to **question 54** if your target system includes either an RJS03 or an RJS04 disk subsystem. The default response is no.

```
55.    Do you want RJS03 rather than RJS04 support [Y/N] (Y)?
```

Respond yes to **question 55** if your target system includes an RJS03 disk subsystem rather than an RJS04. SYSGEN asks this question only if you respond yes to question 54. The default response to this question is yes.

```
RJS04 support assumed.

56.    Do you want RK05/RK05F disk support [Y/N] (N)?
```

Respond yes to **question 56** if your target configuration includes an RK11/RK05/RK05F disk subsystem. The default response to this question is no.

```
57.    Do you want RL01/RL02 disk support [Y/N] (N)?
```

Respond yes to **question 57** if your target configuration includes an RL11/RL01/RL02 disk subsystem. The default response to this question is no.

```
58.    How many RL01/RL02 units are to be supported [D] (2)?
```

Respond to **question 58** with the number (decimal) of RL01 or RL02 units included in your RL01/RL02 subsystem. SYSGEN asks this question only if you respond yes to question 57. The range for valid responses is from 1 to 4 and the default response is 2.

```
59.   Do you want RP11/RPR02 or RP11/RP03 disk
      support [Y/N] (N)?
```

Respond yes to **question 59** if your target configuration includes an RP11/RPR02 or RP11/RP03 disk subsystem. The default response to this question is no.

```
60.   Do you want RPR02 rather than RPR02/RP03
      support [Y/N] (Y)?
```

Respond yes to **question 60** if your target system includes an RPR02 disk subsystem rather than RPR02/RP03. SYSGEN asks this question only if you respond yes to question 59. The default response to this question is yes.

```
RPR02 and RP03 support assumed.

61.   Do you want RK06/RK07 disk support [Y/N] (N)?
```

Respond yes to **question 61** if your target configuration includes an RK611/RK06/RK07 disk subsystem. The default response to this question is no.

```
62.   Do you want magnetic tape support [Y/N] (N)?
```

Respond yes to **question 62** if your target configuration includes an industry-compatible TM11, TS11, or TJU16 magtape subsystem. The default response to this question is no.

```
63.   Do you want TM11 (UNIBUS) magtape support [Y/N] (Y)?
```

Respond yes to **question 63** if your magtape controller is a TM11 family controller. The default response to this question is yes.

```
FILE-STRUCTURED MAGTAPE OPTION:
RT-11 magtape  support  is  available  in  two  forms.  The
standard form is file-structured magtape support, which uses
a subset of ANSI file format.  The  file-structured  handler
stores  and retrieves data in a file format usable with most
RT-11 system programs.  The second form of magtape  support,
hardware  handler  support, omits file structuring and reads
and writes data directly in variable length records.  While
this   handler   is   significantly   smaller   than   the
file-structured handler, you cannot use it with  any  system
programs.  The  file-structured  handler  can  also perform
direct hardware functions.

64.   Do you want the file-structured magtape
      handler [Y/N] (Y)?
```

Respond yes to **question 64** if you want standard RT–11 magtape support. That is, make this selection if you intend to use any of the system programs, RT–11 commands, FORTRAN programs, or BASIC programs with the magtape. The default response to this question is yes.

```
NUMBER OF MAGTAPE UNITS OPTION:
The magtape handler requires table space for each
magtape unit. You can minimize the size of the magtape
handler by limiting the number of units it can
simultaneously handle to the number actually present on the
target machine.

65.   How many magtape units are to be supported [D] (2)?
```

Respond to **question 65** with the number (decimal) of magtape drives included in your magtape subsystem. SYSGEN asks this question only if you respond yes to question 64. The range for valid responses is from 1 to 8 and the default response is 2.

```
66.   Do you want TJU16 (MASSBUS) magtape
      support [Y/N] (N)?
```

Respond yes to **question 66** if your magtape controller is a TJU16 controller. The default response to this question is no.

```
FILE-STRUCTURED MAGTAPE OPTION:
RT-11 magtape support is available in two forms. The
standard form is file-structured magtape support, which uses
a subset of ANSI file format. The file-structured handler
stores and retrieves data in a file format usable with most
RT-11 system programs. The second form of magtape support,
hardware handler support, omits file structuring and reads
and writes data directly in variable length records. While
this handler is significantly smaller than the
file-structured handler, you cannot use it with any system
programs. The file-structured handler can also perform
direct hardware functions.

67.   Do you want the file-structured magtape
      handler [Y/N] (Y)?
```

Respond yes to **question 67** if you want standard RT–11 magtape support. That is, make this selection if you intend to use any of the system programs, RT–11 commands, FORTRAN programs, or BASIC programs with the magtape. The default response to this question is yes.

```
NUMBER OF MAGTAPE UNITS OPTION:
The magtape handler requires table space for each
magtape unit. You can minimize the size of the magtape
handler by limiting the number of units it can
simultaneously handle to the number actually present on the
target machine.


68.   How many magtape units are to be supported [D] (2)?
```

Respond to **question 68** with the number (decimal) of magtape drives included in your magtape subsystem. SYSGEN asks this question only if you respond yes to question 67. The range for valid responses is from 1 to 8 and the default response is 2.

```
69.    Do you want TS11 (UNIBUS) magtape
       support [Y/N] (N)?
```

Respond yes to **question 69** if your magtape controller is a TS11 controller. The default response to this question is no.

```
FILE-STRUCTURED MAGTAPE OPTION:
RT-11 magtape support is available in two forms. The
standard form is file-structured magtape support, which uses
a subset of ANSI file format. The file-structured handler
stores and retrieves data in a file format usable with most
RT-11 system programs. The second form of magtape support,
hardware handler support, omits file structuring and reads
and writes data directly in variable length records. While
this handler is significantly smaller than the
file-structured handler, you cannot use it with any system
programs. The file-structured handler can also perform
direct hardware functions.
```

```
70.    Do you want the file-structured magtape
       handler [Y/N] (Y)?
```

Respond yes to **question 70** if you want standard RT–11 magtape support. That is, make this selection if you intend to use any of the system programs, RT–11 commands, FORTRAN programs, or BASIC programs with the magtape. The default response to this question is yes.

```
NUMBER OF MAGTAPE UNITS OPTION:
The magtape handler requires table space for each
magtape unit. You can minimize the size of the magtape
handler by limiting the number of units it can
simultaneously handle to the number actually present on the
target machine.
```

```
71.    How many magtape units are to be supported [D] (2)?
```

Respond to **question 71** with the number (decimal) of magtape drives included in your magtape subsystem. SYSGEN asks this question only if you respond yes to question 70. The range for valid responses is from 1 to 8 and the default response is 2.

```
Each TS11 unit requires two contiguous unibus addresses for
status registers.
```

```
72.    What is the address of the first
       unit [O] (172522)?
```

Respond to **question 72** with the address (octal) of the first TS11 unit. The range for valid responses is from 160000 to 177570. The Field Service Engineer who installs your hardware system is responsible for supplying you with a written record of addresses at which he or she installs each device. The default response to this question is 172522.

```
73.    What is the vector address of the first unit [O] (224)?
```

Respond to **question 73** with the vector address (octal) of the first TS11 unit. The range for valid responses is from 100 to 474. The default response to this question is 224.

SYSGEN may or may not ask questions 74 through 87, depending on your answer to question 71, which indicates the number of TS11 units to be supported. SYSGEN asks you for the addresses of as many units as you specified.

```
74.   What is the address of the second unit [O] (172526)?

75.   What is the vector address of the second
      unit [O] (300)?

76.   What is the address of the third unit [O] (172532)?

77.   What is the vector address of the third
      unit [O] (304)?

78.   What is the address of the fourth unit [O] (172536)?

79.   What is the vector address of the fourth
      unit [O] (310)?

80.   What is the address of the fifth unit [O] (172542)?

81.   What is the vector address of the fifth
      unit [O] (314)?

82.   What is the address of the sixth unit [O] (172546)?

83.   What is the vector address of the sixth
      unit [O] (320)?

84.   What is the address of the seventh unit [O] (172552)?

85.   What is the vector address of the seventh
      unit [O] (324)?

86.   What is the address of the eighth unit [O] (172556)?

87.   What is the vector address of the eighth
      unit [O] (330)?

88.   Do you want TA11 cassette support [Y/N] (N)?
```

Respond yes to **question 88** if your target configuration includes a TA11/TU60 cassette subsystem. The default response to this question is no.

```
89.   Do you want line printer support [Y/N] (Y)?
```

Respond yes to **question 89** if your target configuration includes a parallel line printer. The default response to this question is yes.

```
The standard line printer vector  address  is  200  and  the
standard   CSR  address  is  177514.   If  your  printer  is
installed at another vector  or  CSR  address, specify  the
correct values.

90.   Does your printer have a nonstandard vector or CSR
      address [Y/N] (N)?
```

Respond yes to **question 90** if the line printer control register address is not 177514 or the vector is not 200. These addresses usually are nonstandard only if the printer is an LA180S. The default response to this question is no.

```
91.    What is the CSR address for the printer [0] (177514)?
```

Respond to **question 91** with the address (octal) of the control register for the line printer. The range for valid responses is from 160000 to 177570. The Field Service Engineer who installs your hardware system is responsible for supplying you with a written record of addresses at which he or she installs each device. The default response to this question is 177514.

```
92.    What is the vector address for the printer [0] (200)?
```

Respond to **question 92** with the vector address (octal) for the line printer. The range for valid responses is from 100 to 474. The default response to this question is 200.

```
93.    Do you want serial line printer support [Y/N] (N)?
```

Respond yes to **question 93** if your target configuration includes a serial line printer. The default response to this question is no.

```
94.    What is the CSR address for the serial line printer
       controller [0] (176500)?
```

Respond to **question 94** with the address (octal) of the control register for the serial line printer. The range for valid responses is from 160000 to 177570. The Field Service Engineer who installs your hardware system is responsible for supplying you with a written record of addresses at which he or she installs each device. The default response to this question is 176500.

```
95.    What is the vector address for the serial line printer
       controller [0] (300)?
```

Respond to **question 95** with the vector address (octal) for the serial line printer. The range for valid responses is from 60 to 474. The default response to this question is 300.

```
The PC11 is a high speed paper tape reader/punch unit.  The
PR11  is  a  similar  unit  with  a high speed reader but no
punch.
```

```
96.    Do you want PC11 high-speed paper tape reader/punch
       support [Y/N] (N)?
```

Respond yes to **question 96** if your target configuration includes a high-speed paper tape reader and punch. If the system has a high-speed reader only, respond no. The default response to this question is no.

```
97.    Do you want PR11 high-speed paper tape reader
       support [Y/N] (N)?
```

Respond yes to **question 97** if your target configuration includes a high-speed reader. The default response to this question is no.

```
98.   Do you want card reader support [Y/N] (N)?
```

Respond yes to **question 98** if your target configuration includes a CR11 or CM11 card reader. The default response to this question is no.

```
The null handler is a software module  that  allows  logical
input  and output to take place without actual physical data
transfers.  It is most useful for debugging applications.

99.   Do you want the null handler [Y/N] (Y)?
```

Respond yes to **question 99** if you want to include the null handler, NL.SYS, in the target system. The default response to this question is yes.

```
100.  Do you want VT11 or VS60 graphics support [Y/N] (N)?
```

Respond yes to **question 100** if your target configuration includes a VT11 or VS60 graphics subsystem. RT–11 support for VS60 is minimal. The display handler, VTHDLR, supports VT11 but does not support any special features of VS60. The default response to this question is no.

```
101.  Do you want VS60 support [Y/N] (N)?
```

Respond yes to **question 101** if the graphics subsystem is VS60. The default response to this question is no.

```
VT11 hardware is assumed.

Many devices have floating CSR and  vector  addresses.   The
presence or absence of such floating address devices affects
the addresses at which the other floating address devices in
the  system are installed.  Floating address devices must be
installed in a standard sequence.

The  standard  VT11/VS60  vector  address  is  320  and  the
standard CSR address is 172000.  However, VT11/VS60s can use
floating addresses.   If  your  VT11/VS60  is  installed  at
nonstandard  vector  and  CSR  addresses, supply the correct
values.

102.  What is the CSR address for the
      VT11/VS60 [O] (172000)?
```

Respond to **question 102** with the address (octal) of the control register for the VT11 or VS60. The range for valid responses is from 160000 to 177570. The Field Service Engineer who installs your hardware system is responsible for supplying you with a written record of addresses at which he or she installs each device. The default response to this question is 172000.

```
103.  What is the vector address for the
      VT11/VS60 [O] (320)?
```

Respond to **question 103** with the vector address (octal) for the VT11 or VS60. The range for valid responses is from 100 to 474. The default response to this question is 320.

```
EXTRA DEVICE SLOT OPTION:
SYSGEN allocates space in the monitor for only  the  devices
```

```
specified.  You can allocate additional space by requesting
empty device slots.  Allocate one empty slot for each device
you  intend  to  add  to the system after it is built.  Also
keep in mind that the number of logical  device  assignments
you  will  be able to make in the running system is equal to
the number of devices plus empty device slots in the  system
you  have  generated.  You may need to allocate extra device
slots for logical device assignment purposes.
```

```
104.  How many extra device slots do you want [D] (0)?
```

Respond to **question 104** with the number (decimal) of extra device slots you need. If a device will be supported only by an in-line service routine in a particular application and not by a system handler, you need not set aside a device slot for it. Remember also to consider the number of logical device assignments you will want to make. The range for valid responses is from 0 to 10. The default response to this question is 0.

```
This completes the device query.
```

SYSGEN asks questions 105 through 151 only if you respond yes to question 12 (for multi-terminal support).

```
TERMINAL INTERFACE OPTIONS:
The terminal interface options allow  you  to  select  RT-11
support  for  the  terminal  interfaces  installed  in  your
system.  RT-11 supports a combination of serial asynchronous
interfaces, which include the KL11, DLV11, DLV11-E, DLV11-F,
DLV11-J, and the DL11 series.  It also supports  DL11-E  and
DLV11-E  interfaces,  with  modem capabilities, and the DZ11
and DZV11 series of asynchronous line multiplexors.

The dialogue asks first for the number of local DL11  lines,
then  the  number of remote DL11 lines.  The total number of
lines is the sum of local and remote lines.  SYSGEN  assigns
physical unit numbers of the DL11 lines first to local lines
and then to remote lines.  This assignment is permanent  and
you cannot change its order.

After SYSGEN has established the number  of  lines  of  each
type,  it  must  establish  the CSR and vector addresses for
each line.  The first such question corresponds to the first
local  line  (the  console),  the second to the second local
line, etc., until all local lines  are  accounted  for.   At
that  point,  the  next question applies to the first remote
line, the second to the second remote line, etc., until  all
remote lines are accounted for.
```

```
105.  How many local DL11 lines, including the console, are
      to be supported, [D] (1)?
```

Respond to **question 105** with the total number of LOCAL DL11 terminals included in the target configuration. Since the console is always a local DL11, this number is never less than 1. Do not include REMOTE DL11 lines in this number. The range for valid responses is from 1 to 8. The default response to this question is 1.

NOTE

If your interfaces are DL11–W interfaces, they should be a specific revision. In order to work with RT–11, DL11–W interfaces should be REV E or higher modules. If they are not

REV E or higher, an Engineering Change Order (ECO DEC–O–LOG M7856–S0002) must be applied to the M7856 module. The field service representative who installs your hardware should apply the ECO, if necessary.

```
106.  How many remote DL11 lines are to be
      supported [D] (0)?
```

Respond to **question 106** with the total number of REMOTE DL11 terminals included in the target configuration. Do not include LOCAL DL11 lines in this number. This response must not be greater than 7, and the sum of this response and the response to question 105 must be less than or equal to 8. The default response to this question is 0.

```
107.  What is the CSR address for the first (console)
      DL11 [O] (177560)?
```

Respond to **question 107** with the address (octal) of the keyboard control register for the console terminal. The range for valid responses is from 160000 to 177570. The Field Service Engineer who installs your hardware system is responsible for supplying you with a written record of addresses at which he or she installs each device. The default response to this question is 177560.

```
108.  What is the vector address for the first (console)
      DL11 [O] (60)?
```

Respond to **question 108** with the keyboard vector address (octal) of the console terminal. The range for valid responses is from 60 to 474. The default response to this question is 60.

The next questions that SYSGEN asks depend on your responses to questions 105 and 106. SYSGEN asks you for the CSR and vector addresses for as many LOCAL interfaces as you specified and for as many REMOTE interfaces as you specified. Respond with the addresses for all the LOCAL interfaces before supplying the addresses for all the REMOTE interfaces. Remember that the default response is in parentheses after the question. After SYSGEN asks you about all the DL11 interfaces, the dialogue asks about DZ11 and DZV11 multiplexors, starting with question 137.

```
109.  What is the CSR address for the next (local)
      DL11 [O] (176500)?
```

Respond to **question 109** with the address (octal) of the keyboard control register for the next LOCAL DL11 interface. The range for valid responses is from 160000 to 177570. The default response to this question is 176500.

```
110.  What is the vector address for the next (local)
      DL11 [O] (300)?
```

Respond to **question 110** with the keyboard vector address (octal) of the LOCAL DL11 interface whose control register was identified in question 109. The range for valid responses is from 60 to 474. The default response to this question is 300.

SYSGEN may or may not ask questions 111 through 136, depending on your answer to questions 105 and 106.

```
111.   What is the CSR address for the next (remote)
       DL11 [0] (175610)?

112.   What is the vector address for the next (remote)
       DL11 [0] (300)?

113.   What is the CSR address for the next (local)
       DL11 [0] (176510)?

114.   What is the vector address for the next (local)
       DL11 [0] (310)?

115.   What is the CSR address for the next (remote)
       DL11 [0] (175620)?

116.   What is the vector address for the next (remote)
       DL11 [0] (310)?

117.   What is the CSR address for the next (local)
       DL11 [0] (176520)?

118.   What is the vector address for the next (local)
       DL11 [0] (320)?

119.   What is the CSR address for the next (remote)
       DL11 [0] (175630)?

120.   What is the vector address for the next (remote)
       DL11 [0] (320)?

121.   What is the CSR address for the next (local)
       DL11 [0] (176530)?

122.   What is the vector address for the next (local)
       DL11 [0] (330)?

123.   What is the CSR address for the next (remote)
       DL11 [0] (175640)?

124.   What is the vector address for the next (remote)
       DL11 [0] (330)?

125.   What is the CSR address for the next (local)
       DL11 [0] (176540)?

126.   What is the vector address for the next (local)
       DL11 [0] (340)?

127.   What is the CSR address for the next (remote)
       DL11 [0] (175650)?

128.   What is the vector address for the next (remote)
       DL11 [0] (340)?

129.   What is the CSR address for the next (local)
       DL11 [0] (176550)?
```

```
130.  What is the vector address for the next (local)
      DL11 [O] (350)?

131.  What is the CSR address for the next (remote)
      DL11 [O] (175660)?

132.  What is the vector address for the next (remote)
      DL11 [O] (350)?

133.  What is the CSR address for the next (local)
      DL11 [O] (176560)?

134.  What is the vector address for the next (local)
      DL11 [O] (360)?

135.  What is the CSR address for the next (remote)
      DL11 [O] (175670)?

136.  What is the vector address for the next (remote)
      DL11 [O] (360)?
```

```
You can select  RT-11  support  for  one  DZ11-A  or  DZ11-B
eight-line  multiplexor  with an additional DZ11-C or DZ11-D
eight-line multiplexor, for a maximum of 16 lines.   Or  you
can  select one to four DZV11 four-line multiplexors.  Modem
support is available on remote lines, but is limited to Bell
103  type  modems  or  equivalent.  RT-11 support for modems
requires  the  modems  to  operate  in   auto-answer   mode.
Therefore,  the  'common  carrier and clear to send' options
must   have   been   installed   (during   manufacture    or
installation).
```

```
137.  Do you want DZ11 or DZV11 multiplexor
      support [Y/N] (N)?
```

Respond yes to **question 137** if your configuration includes a DZ11 or DZV11 multiplexor. The default response to this question is no.

```
138.  Do you want DZ11 multiplexor support [Y/N] (Y)?
```

Respond yes to **question 138** if your configuration includes a DZ11 multiplexor. SYSGEN asks you this question only if you respond yes to question 137. The default response to this question is yes. If you respond no to this question, SYSGEN prints the following message.

```
DZV11 four-line multiplexor support is assumed.
```

```
139.  How many DZ multiplexors are to be supported [D] (1)?
```

Respond to **question 139** with the number (decimal) of DZ multiplexors in the target configuration. SYSGEN asks you this question only if you respond yes to question 137. The default response to this question is 1.

```
The following dialogue asks for the number of local DZ11  or
DZV11  lines,  then  the  number of remote lines.  The total
number of lines is  the  sum  of  local  and  remote  lines.
SYSGEN  assigns  physical  unit numbers of the DZ11 or DZV11
lines first to local lines and then to remote  lines.   This
assignment is permanent and you cannot change its order.
```

```
140.  How many local DZ lines are to be
      supported altogether [D] (1)?
```

Respond to **question 140** with the total number (decimal) of DZ lines that are local terminals. The range for valid responses is from 0 to 16. The default response to this question is 1.

```
141.  How many remote DZ lines are to be
      supported altogether [D] (0)?
```

Respond to **question 141** with the total number (decimal) of DZ lines that are remote terminals. The range for valid responses is from 0 to 16. The default response to this question is 0.

```
The interface vectors and CSR addresses are assigned to  the
floating device region and vary with each installation.

142.  What is the CSR address for the
      first DZ multiplexor [O] (160010)?
```

Respond to **question 142** with the address (octal) of the keyboard control register for the first DZ multiplexor. The range for valid responses is from 160000 to 177570. The Field Service Engineer who installs your hardware system is responsible for supplying you with a written record of addresses at which he or she installs each device. The default response to this question is 160010.

```
143.  What is the vector address for the
      first DZ multiplexor [O] (300)?
```

Respond to **question 143** with the keyboard vector address (octal) of the first DZ multiplexor. The range for valid responses is from 60 to 474. The default response to this question is 300.

```
144.  What is the CSR address for the
      second DZ multiplexor [O] (160020)?
```

Respond to **question 144** with the address (octal) of the keyboard control register for the second DZ multiplexor. The range for valid responses is from 160000 to 177570. The Field Service Engineer who installs your hardware system is responsible for supplying you with a written record of addresses at which he or she installs each device. The default response to this question is 160020.

```
145.  What is the vector address for the
      second DZ multiplexor [O] (310)?
```

Respond to **question 145** with the keyboard vector address (octal) of the second DZ multiplexor. The range for valid responses is from 60 to 474. The default response to this question is 310.

```
146.  What is the CSR address for the
      third DZ multiplexor [O] (160030)?
```

Respond to **question 146** with the address (octal) of the keyboard control register for the third DZ multiplexor. The range for valid responses is 160000 to 177570. The default response to this question is 160030.

```
147.  What is the vector address for the
      third DZ multiplexor [O] (320)?
```

Respond to **question 147** with the keyboard vector address (octal) of the third DZ multiplexor. The range for valid responses is from 60 to 474. The default response to this question is 320.

```
148.  What is the CSR address for the
      fourth DZ multiplexor [O] (160040)?
```

Respond to **question 148** with the address (octal) of the keyboard control register for the fourth DZ multiplexor. The range for valid responses is 160000 to 177570. The default response to this question is 160040.

```
149.  What is the vector address for the
      fourth DZ multiplexor [O] (340)?
```

Respond to **question 149** with the keyboard vector address (octal) of the fourth DZ multiplexor. The range for valid responses is from 60 to 474. The default response to this question is 340.

```
The lines must be initialized to a specific baud rate.  You
can select 300, 150, or 110 baud.

150.  Do you want the lines initialized
      to 300 baud [Y/N] (Y)?
```

Respond yes to **question 150** if you want the baud rate to be initialized to 300. All DZ11 or DZV11 lines must be initialized to the same baud rate. The default response to this question is yes.

NOTE

> The maximum single terminal data rate for input is 300 baud. The maximum total data rate of all the terminals in a multi-terminal system is 4800 baud.

```
151.  Do you want the lines initialized
      to 150 baud [Y/N] (Y)?
```

Respond yes to **question 151** if you want the baud rate to be initialized to 150. The default response to this question is yes.

```
110 baud is assumed.

SYSTEM BUILD OPTIONS:
The SYSGEN assembling and linking process requires a  source
input   device   and   a   binary   output   device, as well as an
output device for monitor link maps.  You must  specify  the
physical   name   and   unit number for each (for example, RK1,
DP4,   etc.).   For    more    explanation,    see    the    RT-11
INSTALLATION AND SYSTEM GENERATION GUIDE.

152.  What is the PHYSICAL name and unit of the source
      input device [xxn] (RK1)?
```

Respond to **question 152** with the physical device and unit number for the device on which you want the system sources to reside during system generation. SYSGEN assigns the logical device SRC: to the physical device you specify. Then, the SYSGEN command files use SRC: for source input. Review Section 8.2.5.1 if you are unsure of the answer. The default response to this question is RK1.

```
153.  What is the PHYSICAL name and unit of the binary
      output device [xxn] (RK0)?
```

Respond to **question 153** with the physical device and unit number for the device that you want to receive binary and system output. SYSGEN assigns the logical device BIN: to the physical device you specify. Then, the SYSGEN command files send to BIN: the .OBJ and system files output during the system generation process. This device can be the same as the source input device; it can also be an independent output device or a system device. In any case, there must be sufficient space for all the files. Review Section 8.2.5.1 if you are unsure of the answer. The default response to this question is RK0.

```
154.  What is the PHYSICAL name and unit of the map
      output device [xxn] (TT)?
```

Respond to **question 154** with the physical device and unit number for the device that you want to list the link maps that result when you assemble and link the system components. SYSGEN assigns the logical device MAP: to the physical device you specify. You can specify the console terminal or a line printer. Or, you can build a file for the map listings by specifying the physical name of a block-replaceable device such as a disk. Usually, the binary output device is suitable for the map files as well. Note that if you submit an SPR to DIGITAL for a system that you created through the system generation process, you must include the link maps (and the file SYCND.MAC) for that system. Review Section 8.2.5.1 if you are unsure of the answer. The default response to this question is TT, the console terminal.

```
RETAIN SYSTEM OBJS OPTION:
The indirect command files that the SYSGEN program generates
delete  the  object modules (from which the system is built)
when the object mooules are no longer needed.  This  measure
serves  to  conserve  disk  space  while the system is being
built.  However, the object modules are often  useful  later
when  you  patch  the  system.  If  your  output device has
sufficient free space, you can elect to  retain  all  system
object modules for future use.

155.  Do you want to retain the system OBJs [Y/N] (Y)?
```

Respond yes to **question 155** if you have abundant mass storage (approximately 500 blocks per monitor being built) and if you want to retain the system .OBJ files for later patching purposes. Note that DIGITAL distributes monitor patches in source form. If in doubt, respond no. The default response to this question is yes.

```
To build an entire  system,  mount  the  source  and  binary
volumes,  copy  the  files  SYCND.MAC  and SYSTBL.MAC to the
source  volume,  and  type  @SYSBLD.   To  build  just  the
monitors,  type @MONBLD.  To build just the device handlers,
type  @DEVBLD.   For  more  information,  read   the   RT-11
INSTALLATION AND SYSTEM GENERATION GUIDE.

END OF SYSGEN PROGRAM --

      Do not forget to copy DK:SYCND.MAC and DK:SYSTBL.MAC to
      the  source  disk before executing the command file(s).
```

Figure 8–6: System Generation Worksheet

□ 1. Long form of dialogue
□ 2. Base-line monitor
□ 3. Single-job monitor
□ 4. Foreground/background monitor
□ 5. Extended memory monitor
□ 6. SJ timer support
□ 7. Device time-out support
□ 8. Message on system I/O errors
□ 9. System job option
□ 10. .SPCPS programmed request
□ 11. Idle loop light pattern
□ 12. Multi-terminal support
□ 13. Asynch. terminal status
□ 14. Multi-term. time out
□ 15. Output ring buffers size __________
□ 16. Input ring buffers size __________
□ 17. Month rollover
□ 18. All keyboard monitor commands
□ 19. Utility program commands
□ 20. Language commands
□ 21. Minimal commands
□ 22. 50 Hz clock
□ 23. Programmable clock — sys.clock
□ 24. Startup indirect command file
□ 25. Floating point
□ 26. Memory parity support
□ 27. Power failure message
□ 28. BATCH
□ 29. Error logging
□ 30. No. units error log supts __________

Peripheral Devices in Target
(31 through 104)

Device	Vector	CSR

□ 32. Second RX11 controller
□ 38. Second RX211 controller
□ 39. Double-density only RX02
□ 45. Second TU58 controller
53. No. of RF11 platters __________
□ 55. RJS03 not RJS04 __________
58. No. of RL01/RL02 units __________
□ 60. RPR02 not RPR02/RP03 __________
□ 64. File-structured TM11 magtape
65. No. TM11 magtape units __________
□ 67. File-structured TJU16 magtape
68. No. TJU16 magtape units __________
□ 70. File-structured TS11 magtape
71. No. TS11 magtape units __________
104. No. of extra device slots __________

Terminal interfaces in target
(105 through 151)

Local	Vector	CSR

Remote	Vector	CSR

105. No. of local DL11 lines __________
106. No. of remote DL11 lines __________
139. No. of DZ multiplexors __________
140. No. of local DZ lines __________
141. No. of remote DZ lines __________

150. Baud rate of lines __________

System Build

152. Source input device __________
153. Binary output device __________
154. Map output device __________
□ 155. Retain system OBJ's

Source Files Required

____
____
____
____
____
____
____
____
____
____
____
____
____
____

Chapter 9
Performing the System Generation Process on a Disk System

To perform the system generation process on a disk system, perform the steps summarized in the following list and described in Sections 9.1 through 9.4.

1. Run the program SYSGEN.SAV.
2. Edit the file SYCND.MAC if necessary.
3. Collect the appropriate files on the appropriate media.
4. Assemble and link monitor(s) and handlers.

The following sections correspond to each of these four steps and describe the procedures involved in each step. Chapter 10 describes the procedures for performing the system generation process on diskette configurations.

9.1 Running the Program SYSGEN.SAV

The first step in the system generation process is to run the SYSGEN program and answer the dialogue questions.

Make sure that the files SYSGEN.SAV, SYSTBL.CND, and SYSGEN.CND are on your system device and that the system device is not write protected. The .CND files can reside on another volume (except magtape or DECtape II) as long as the volume has 2000 free blocks and you assign the default DK: to that unit. The procedures in this section assume that these files are on the system volume. Use the following command.

```
.R SYSGEN(RET)
```
(the dialogue prints)

Use the worksheet you compiled when you read Chapter 8, and respond to each dialogue question. The dialogue asks a question, waits for your response, and resumes at the appropriate point, depending on the response you give. If you are not familiar with the Version 4 SYSGEN dialogue, respond yes to the first question so that the long form of the dialogue prints. The long form includes explanations of the various questions.

SYSGEN may skip some questions. SYSGEN prompts you for a particular type of response. The prompt appears in brackets [] after each question, in the following format:

[Y/N]	type Y or N for yes or no
[D]	type a decimal number
[O]	type an octal number
[xx]	type device name (such as DX for RX01)
[xxn]	type device name and unit number

Follow all responses with RETURN. The parentheses in the question contain the default response. After you respond to the last question, the end message prints on the terminal and control returns to the monitor. You can terminate SYSGEN by typing CTRL/C.

NOTE

> Type only RETURN to select the default response. The parentheses in the question contain the default response.

If you choose the long form of dialogue, you can type CTRL/O to prevent an explanation from printing. SYSGEN stops printing the explanation, prints the question, waits for your response, and resumes the long form of dialogue. Note, however, that you should not type CTRL/O while a question is printing. In this case, you would not be able to read the question, but SYSGEN would wait for a response.

If you give an inappropriate response to a dialogue question, SYSGEN prints an error message. Table 9–1 lists the most likely SYSGEN error messages. SYSGEN error messages follow FORTRAN format, since SYSGEN is a FORTRAN program. If, while running SYSGEN, you receive an error message not listed in Table 9–1, check the the *RT–11 System Message Manual* for that message. After SYSGEN issues an error message, it repeats the question. If you want to change a response to a question, rerun SYSGEN.

Table 9–1: SYSGEN Error Messages

Message	Cause
28 Open failed for file in routine "MAIN" line 17	SYSGEN.CND not present on DK:.
28 Open failed for file in routine "MAIN" line 19	SYSTBL.CND not present on DK:.
62 FORTRAN start fail	Insufficient memory to run SYSGEN.
?SYSGEN-W-Answer value too large	Your response to a question contained a value greater than the greatest acceptable one. Refer to the specific question and its explanation in Section 8.3 for the range of valid responses.
?SYSGEN-W-Answer value too small	Your response to a question contained a value less than the smallest acceptable one. Refer to the specific question and its explanation in Section 8.3 for the range of valid responses.
?SYSGEN-W-Inappropriate answer	Your response was not of the type required. For example, you may have answered yes when a numeric response was required.
?SYSGEN-F-Input error	SYSGEN experienced an error in reading the file SYSGEN.CND.
?SYSGEN-F-Insert error	SYSGEN experienced an error in reading the file SYSTBL.CND.
?SYSGEN-F-No monitor requested	You did not select one of the monitors.
?SYSGEN-F-Output error	You do not have sufficient space on the default device (DK:) to store the command and conditional files, or you already have one of the output files on DK: and it is a protected file. This message may also result from a hard I/O error in those files.

9.2 Editing SYCND.MAC

If you need to make any edits to SYCND.MAC (see Section 8.2.4), do so at this time, but take care to edit the file precisely.

9.3 Collecting the Appropriate Files on the Appropriate Media

At this point, prepare to copy files to devices according to the arrangement you planned in Section 8.2.5. Gather patched distribution backup copies that contain the source files you will need. Make sure the command files MONBLD.COM and DEVBLD.COM (and SYSBLD.COM if you can use it) are on device DK:.

9.4 Assembling and Linking Monitor(s) and Handlers

SYSGEN creates command files to assemble and link the monitor and handlers you are generating. These command files consist of assembly and link commands (plus file deletion commands if you chose not to retain the .OBJ files). You can execute the command files as they are, or you can alter them, as long as the final commands perform the required functions. For example, you might want to alter the command files as follows:

1. You can change the map output device to a block-replaceable device and create a file instead of a listing. You can suppress the link maps by sending the output to the null device, NL:. However, you will need to include a link map and SYCND.MAC listing with any SPR you might submit to DIGITAL for a monitor created by SYSGEN. DIGITAL recommends saving the link maps in a file on a disk, if possible, so they will not be misplaced.
2. You can change the devices on which individual source or binary files are to reside to optimize mass storage use.

You can choose the indirect command file(s) you want to use to build the system. SYSGEN creates SYSBLD.COM, MONBLD.COM, and DEVBLD.COM. If you want to build both the handlers and the monitors in the same operation and you have sufficient free storage, use SYSBLD.COM, which invokes MONBLD.COM and DEVBLD.COM. On the other hand, if you want to build the handlers separately from the monitors, use MONBLD.COM to build the monitors and DEVBLD.COM to build the handlers.

Edit the indirect command files at this point, if necessary. Then proceed to the section that describes the procedure you need to use:

- Using SYSBLD to build the system
- Using MONBLD and DEVBLD to build the system
- Building handlers separately

NOTE

> To complete the monitor build procedure as rapidly as possible, execute the SYSBLD or MONBLD file under the SJ monitor with SET USR SWAP (default) and no unnecessary loaded handlers.

9.4.1 Using SYSBLD to Build the System

Generally, if you have sufficient space on the device BIN: (approximately 1500 free blocks), you should use SYSBLD to build the system. Make sure that the disks you planned to serve as SRC: and BIN: are mounted. Refer to Section 8.2.5 to review your plans for the system build procedure.

Copy the appropriate source files to the device that is to serve as SRC:. In the following command, xxn: is your distribution backup device, unless you put source files on one of your working system volumes when you installed RT–11; yyn: is SRC:.

```
.COPY xxn:filnam.MAC yyn:filnam.MAC(RET)
.
```

Then copy SYCND.MAC and SYSTBL.MAC to the source disk, if necessary. In the following command, xxn: is the device that contains SYCND and SYSTBL and yyn: is the device you specified as SRC: when you answered question 152 of the SYSGEN dialogue.

```
.COPY xxn:SYCND.MAC yyn:SYCND.MAC(RET)

.COPY xxn:SYSTBL.MAC yyn:SYSTBL.MAC(RET)
.
```

Then consolidate the free space on the binary disk. This step is very important if you have space problems. In this command, zzn: is the device you specified as BIN: when you answered question 153 of the SYSGEN dialogue.

```
.SQUEEZE zzn:(RET)
zzn:/Squeeze; Are you sure? Y(RET)
.
```

Invoke the indirect command file SYSBLD.COM. Each command line in SYSBLD, MONBLD, and DEVBLD logs on the terminal as the system executes the command. The following example shows terminal output similar to the output that will print on your terminal when you invoke SYSBLD.

```
.@SYSBLD(RET)
@MONBLD
ASSIGN RK1 SRC
ASSIGN RK0 BIN
ASSIGN TT MAP
MACRO/OBJ:BIN:KMFB SRC:(FB+SYCND+EDTGBL+KMON+KMOVLY)
MACRO/OBJ:BIN:RMFB SRC:(FB+SYCND+EDTGBL+USR+RMONFB)
MACRO/OBJ:BIN:TBFB SRC:(FB+SYCND+EDTGBL+SYSTBL)
MACRO/OBJ:BIN:BTFB SRC:(FB+SYCND+EDTGBL+BSTRAP)
MACRO/OBJ:BIN:MEFB SRC:(FB+SYCND+EDTGBL+MTTEMT)
MACRO/OBJ:BIN:MIFB SRC:(FB+SYCND+EDTGBL+MTTINT)
LINK/EXE:BIN:RT11FB.SYG/BOU:1000/PROMPT/MAP:MAP:RT11FB BIN:BTFB
BIN:RMFB,KMFB,MEFB,MIFB,TBFB//
OVLY0
@DEVBLD
ASSIGN RK1 SRC
ASSIGN RK0 BIN
MACRO/OBJ:BIN:BA SRC:(SYCND+BA)
LINK/EXE:BIN:BA.SYG BIN:BA
MACRO/OBJ:BIN:DX SRC:(SYCND+DX)
LINK/EXE:BIN:DX.SYG BIN:DX
MACRO/OBJ:BIN:RK SRC:(SYCND+RK)
LINK/EXE:BIN:RK.SYG BIN:RK
MACRO/OBJ:BIN:LP SRC:(SYCND+LP)
LINK/EXE:BIN:LP.SYG BIN:LP
.
```

The build process will take approximately half an hour. The most common errors that occur during the build process are listed in Table 9–2.

Table 9–2: System Build Errors

Type of Error	Cause[1]
Assembly errors	Incorrect or conflicting responses to SYSGEN dialogue. Reexamine your responses to SYSGEN. Often, reading the conditional files helps you spot errors. If the SYSGEN responses seem correct, make sure that you have the proper versions of the source files on the source disk. Check that you have not introduced an error if you edited the source files. Compare the files on the source disk to the originals in the distribution.
?MACRO-F-File not found DEV:FILNAM.TYP	Failure to copy the SYCND.MAC or SYSTBL.MAC files to the source disk or failure to include all necessary source modules on the disk.
?MACRO-F-Output device full on DEV:FILNAM.TYP	You do not have sufficient space on the binary output device to accommodate all output files.
?MACRO-F-I/O error on workfile	You do not have sufficient space on the binary output device. Check the following possibilities: You did not squeeze the binary device before starting the build. Try squeezing it and rebuilding. Try rerunning SYSGEN to specify not retaining .OBJ files and then rebuild. Try deleting other unnecessary files on the binary device (but be sure they are unnecessary). If all these techniques fail, you must build the system manually, entering the commands individually and building one component at a time. Refer to Chapter 10.
?MACRO-F-I/O error on DEV:FILNAM.TYP	Bad volumes or write protected devices. If the device is write protected, enable it. If the volume is bad, try another.

[1] If you encounter an error you cannot explain or correct, send an SPR to DIGITAL with a listing of the conditional files SYCND.MAC and SYSTBL.MAC, along with the console output.

Once you have built the system, copy and store the conditional, command, and output files from the generation process, if possible. You must retain at least the conditional and command files since all SPRs must be accompanied by a listing of these files (as well as link maps) if you use a generated monitor. If you keep the object files, you can reassemble a particular component and relink manually, if necessary; you need not rebuild the entire system each time you make a patch.

The .SYG files that result from the system build are masters for your generated system and you should preserve them. Refer to the appropriate installation chapter, and perform the procedures to install the generated system. Rename .SYG files, monitors and handlers, to .SYS. You can also rename the monitor to distinguish it from standard monitors. You will have to copy the bootstrap for the generated monitor to your system device if you want that monitor to boot. Note that you can always identify a user-generated monitor by the (S) that prints in the boot message.

9.4.2 Using MONBLD and DEVBLD to Build the System

If you use MONBLD and DEVBLD to build the system, you need approximately 1500 free blocks on device BIN:. Make sure that the disks you planned to serve as SRC: and BIN: are mounted. Refer to Section 8.2.5 to review your plans for the system build procedure.

Copy the source files for building the monitor(s) to the device that is to serve as SRC:. In the following command, xxn: is your distribution backup device, unless you included the source files in your working system; yyn: is SRC:.

```
.COPY xxn:filnam.MAC yyn:filnam.MAC(RET)
.
```

Copy all the appropriate source files in this way. Then copy SYCND.MAC and SYSTBL.MAC to the source disk. In the following command, xxn: is the device that contains SYCND and SYSTBL and yyn: is the device you specified as SRC: when you answered question 152 of the SYSGEN dialogue.

```
.COPY xxn:SYCND.MAC yyn:SYCND.MAC(RET)

.COPY xxn:SYSTBL.MAC yyn:SYSTBL.MAC(RET)
.
```

Then consolidate the free space on the binary disk. In this command, zzn: is the device you specified as BIN: when you answered question 153 of the SYSGEN dialogue.

```
.SQUEEZE zzn:(RET)
zzn:/Squeeze; Are you sure? Y(RET)
.
```

Invoke the indirect command file MONBLD.COM to build the monitors. The currently running monitor logs the commands on the terminal as it executes them. The following example shows terminal output similar to the output that will print on your terminal when you invoke MONBLD.

```
.@MONBLD(RET)
ASSIGN RK1 SRC
ASSIGN RK0 BIN
ASSIGN TT MAP
MACRO/OBJ:BIN:KMFB SRC:(FB+SYCND+EDTGBL+KMON+KMOVLY)
MACRO/OBJ:BIN:RMFB SRC:(FB+SYCND+EDTGBL+USR+RMONFB)
MACRO/OBJ:BIN:TBFB SRC:(FB+SYCND+EDTGBL+SYSTBL)
MACRO/OBJ:BIN:BTFB SRC:(FB+SYCND+EDTGBL+BSTRAP)
MACRO/OBJ:BIN:MEFB SRC:(FB+SYCND+EDTGBL+MTTEMT)
MACRO/OBJ:BIN:MIFB SRC:(FB+SYCND+EDTGBL+MTTINT)
LINK/EXE:BIN:RT11FB.SYG/BOU:1000/PROMPT/MAP:MAP:RT11FB BIN:BTFB
BIN:RMFB,KMFB,MEFB,MIFB,TBFB//
OVLY0
.
```

The build process will take approximately half an hour. The most common errors that occur during the build process are listed in Table 9–2.

Now delete the source files that you needed to build the monitors from SRC:. Copy the source files for building the handlers to SRC:. Do not delete SYCND.MAC or SYSTBL.MAC. In the following command, xxn: is your distribution backup device and yyn: is SRC:.

```
.DELETE yyn:aaaaaa.MAC(RET)
 Files deleted:
yyn:aaaaaa.MAC ? Y(RET)
.
```

Continue by deleting all the source files for building the monitor(s). Then copy the sources for building the handlers.

```
.COPY xxn:aa.MAC yyn:aa.MAC(RET)
                 .
                 .
.COPY xxn:zz.MAC yyn:zz.MAC(RET)
.
```

Copy all the appropriate source files in this way. Next, invoke the indirect command file DEVBLD.COM to build the device handlers. The monitor logs the commands on the terminal as it executes them. The following example shows terminal output similar to the output that will print on your terminal when you invoke DEVBLD.

```
.@DEVBLD(RET)
@DEVBLD
ASSIGN RK1 SRC
ASSIGN RK0 BIN
MACRO/OBJ:BIN:BA SRC:(SYCND+BA)
LINK/EXE:BIN:BA.SYG BIN:BA
MACRO/OBJ:BIN:DX SRC:(SYCND+DX)
LINK/EXE:BIN:DX.SYG BIN:DX
```

```
MACRO/OBJ:BIN:RK SRC:(SYCND+RK)
LINK/EXE:BIN:RK.SYG BIN:RK
MACRO/OBJ:BIN:LP SRC:(SYCND+LP)
LINK/EXE:BIN:LP.SYG BIN:LP
.
```

The most common errors that occur during the build process are listed in Table 9–2.

Once you have built the system, copy and store the conditional, command, and output files from the generation process if possible. You must retain at least the conditional and command files, since all SPRs must be accompanied by a listing of these files (as well as link maps) if you use a generated monitor. If you keep the object files, you can reassemble a particular component and relink manually, if necessary; you need not rebuild the entire system each time you make a patch.

The .SYG files that result from the system build are masters for your generated system and you should preserve them. Refer to the appropriate installation chapter, and perform the procedures to install the generated system. Rename .SYG files, monitors and handlers, to .SYS. You can also rename the monitor to distinguish it from standard monitors. You will have to copy the bootstrap for the generated monitor to your system device if you want that monitor to boot. Note that you can always identify a user-generated monitor by the (S) that prints in the boot message.

9.4.3 Building Handlers Separately

To keep the monitor as small as possible, you may want to build a monitor with only a few device slots, build many device handlers to go along with that monitor, but use only a few devices at a time. When you select a device during the SYSGEN dialogue, SYSGEN allocates a slot for it in the monitor device tables and adds commands to DEVBLD that build it. To build device handlers separately, you need to run SYSGEN twice.

First, execute SYSGEN, selecting monitor options and support for the few devices for which you want to allocate slots. Use the resulting conditional and command files to build the basic system. Then, rerun SYSGEN, specifying the identical monitor options, but selecting only the additional devices. Use the version of DEVBLD created during the second SYSGEN session to build the additional handlers. Discard MONBLD and SYSBLD files from the second SYSGEN run.

You can use one of two techniques to install additional device handlers for the devices you want to use during a particular session.

1. Use the REMOVE and INSTALL commands.
2. Use the bootstrap routine's automatic install capability by simply replacing device handlers on the system device and rebooting.

Refer to Section 2.8.13 for a description of both techniques.

Chapter 10

Performing the System Generation Process on a Small System

If you intend to perform the system generation process on an RX01 diskette system, a PDT–11/150 diskette system, or an RX02 diskette system, follow the procedures in this chapter. If your system has two diskette units and a hard-copy terminal or line printer, you can perform the system generation process, but it is not a recommended procedure. It takes a long time and is not automated. If you do want to attempt the procedure, follow the steps in this chapter very carefully. Do not try to generate a system for a target configuration that is different from the diskette configuration on which you are performing the system generation. The procedure in this chapter may not work if you try to generate support for many devices.

10.1 Creating a Working System for This Procedure

Begin the system generation procedure by creating a system diskette with the single-job monitor.

Include the following files (from your working system and distribution backups) on the system diskette:

RT11SJ.SYS
xx.SYS (DX.SYS, PD.SYS, or DY.SYS)
SWAP.SYS
TT.SYS
LP.SYS (if appropriate)
SYSGEN.SAV
SYSGEN.CND
SYSTBL.CND
DIR.SAV
PIP.SAV
DUP.SAV

Be sure to squeeze the diskette and copy the bootstrap to it. Then boot this system. Do not forget to set the time and date.

10.2 Running the Program SYSGEN.SAV

This procedure describes building only one monitor (and its associated device handlers) at a time. Run SYSGEN once for each monitor you need. Each time you run SYSGEN, answer yes to only one of the questions that offer the various monitors (questions 2 through 5). Follow the procedures in this chapter to build that monitor and its associated handlers. Then run SYSGEN again to choose another monitor. Repeat the procedures in this chapter to build the monitor and handlers, and so on.

Use the following command to run the SYSGEN program.

```
.R SYSGEN(RET)
```

(the dialogue prints)

Use the worksheet you compiled when you read Chapter 8, and respond to each dialogue question. The dialogue asks a question, waits for your response, and resumes at the appropriate point, depending on the response you give. If you are not familiar with the Version 4 SYSGEN dialogue, respond yes to the first question so that the long form of the dialogue prints. The long form includes explanations of the various questions.

Answer the system build questions (152 through 155) with the names of devices in your configuration. Since you read the indirect command files (not run them), it does not matter which device is SRC:, BIN:, or MAP:.

SYSGEN may skip some questions. SYSGEN prompts you for a particular type of response. The prompt appears in brackets [] after each question in the following format:

[Y/N]	type Y or N for yes or no
[D]	type a decimal number
[O]	type an octal number
[xx]	type device name (such as DX for RX01)
[xxn]	type device name and unit number

Follow all responses with RETURN. After you respond to the last question, the end message prints on the terminal and control returns to the monitor. You can terminate SYSGEN by typing CTRL/C.

NOTE

> Type only RETURN to select the default response. The parentheses in the question contain the default response.

If you choose the long form of dialogue, you can type CTRL/O to prevent an explanation from printing. SYSGEN stops printing the explanation, prints the question, waits for your response, and resumes the long form of dialogue. Note, however, that you should not type CTRL/O while a question is printing. In this case, you would not be able to read the question, but SYSGEN would wait for a response.

If you give an inappropriate response to a dialogue question, SYSGEN prints an error message. Refer to Table 10–1, which lists the most likely SYSGEN error messages. SYSGEN error messages follow FORTRAN format, since SYSGEN is a FORTRAN program. If, while running SYSGEN, you receive an error message not listed in Table 10–1, check the *RT–11 System Message Manual* for this message. After SYSGEN issues an error message, it repeats the question. If you want to change a response to a question, rerun SYSGEN.

Table 10–1: SYSGEN Error Messages

Message	Cause
28 Open failed for file in routine "MAIN" line 17	SYSGEN.CND not present on DK:.
28 Open failed for file in routine "MAIN" line 19	SYSTBL.CND not present on DK:.
62 FORTRAN start fail	Insufficient memory to run SYSGEN.
?SYSGEN-W-Answer value too large	Your response to a question contained a value greater than the greatest acceptable one. Refer to the specific question and its explanation in Section 8.3 for the range of valid responses.
?SYSGEN-W-Answer value too small	Your response to a question contained a value less than the smallest acceptable one. Refer to the specific question and its explanation in Section 8.3 for the range of valid responses.
?SYSGEN-W-Inappropriate answer	Your response was not of the type required. For example, you may have answered yes when a numeric response was required.
?SYSGEN-F-Input error	SYSGEN experienced an error in reading the file SYSGEN.CND.
?SYSGEN-F-Insert error	SYSGEN experienced an error in reading the file SYSTBL.CND.
?SYSGEN-F-No monitor requested	You did not select one of the monitors.
?SYSGEN-F-Output error	You do not have sufficient space on the default device (DK:) to store the command and conditional files, or you already have one of the output files on DK: and it is a protected file. This message may also result from a hard I/O error in those files.

When SYSGEN completes, the following new files should appear in your system diskette's directory.

SYCND.MAC
SYSTBL.MAC
SYSBLD.COM
MONBLD.COM
DEVBLD.COM

Using the following command, examine your directory to make sure that these files have been created.

```
.DIRECTORY(RET)
(the directory prints)
.
```

10.3 Collecting the Appropriate Files on the Appropriate Media

Because you cannot fit all the files required to build your system on two diskettes, you must arrange the files so that you can copy them to your system diskette one at a time and perform the component assemblies separately. First you will need to study the indirect command files to identify the files you will need for system build. Then you will need to create four diskettes to hold the various files during the procedure.

10.3.1 System Build Indirect Command Files

To ascertain the files you need, list the files MONBLD.COM and DEVBLD.COM on the terminal or line printer. These files contain the names of all the source files you need to perform the assemblies to build the system. They also contain all the keyboard monitor commands that assemble and link it.

on terminal

```
.TYPE MONBLD.COM(RET)
.TYPE DEVBLD.COM(RET)
```

on line printer

```
.PRINT MONBLD.COM(RET)
.PRINT DEVBLD.COM(RET)
```

Figure 10–1 shows an example of MONBLD and DEVBLD (for a multiterminal foreground/background system). Certain lines in this example are numbered for reference. For example, Lines 1, 2, and 3 show the commands that assign various devices to SRC:, BIN:, and MAP:.

Figure 10–1: Example MONBLD and DEVBLD

```
     !       MONBLD.COM
     !
     ! THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY ONLY BE
     ! USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.
     !
     !      COPYRIGHT (C) 1979 BY DIGITAL EQUIPMENT CORPORATION
     !
1    ASSIGN RK1 SRC
2    ASSIGN RK0 BIN
3    ASSIGN TT MAP
4    MACRO/OBJ:BIN:KMFB SRC:(FB+SYCND+EDTGBL+KMON+KMOVLY)
5    MACRO/OBJ:BIN:RMFB SRC:(FB+SYCND+EDTGBL+USR+RMONFB)
6    MACRO/OBJ:BIN:TBFB SRC:(FB+SYCND+EDTGBL+SYSTBL)
7    MACRO/OBJ:BIN:BTFB SRC:(FB+SYCND+EDTGBL+BSTRAP)
8    MACRO/OBJ:BIN:MEFB SRC:(FB+SYCND+EDTGBL+MTTEMT)
9    MACRO/OBJ:BIN:MIFB SRC:(FB+SYCND+EDTGBL+MTTINT)
10   LINK/EXE:BIN:RT11FB.SYG/BOU:1000/PROMPT/MAP:MAP:RT11FB
11   BIN:BTFB BIN:RMFB,KMFB,MEFB,MIFB,TBFB//
12   OVLYO
```

```
     !       DEVBLD.COM
     !
     ! THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY ONLY BE
     ! USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.
     !
     !       COPYRIGHT (C) 1979 BY DIGITAL EQUIPMENT CORPORATION
     !

13   ASSIGN RK1 SRC
14   ASSIGN RK0 BIN
15   MACRO/OBJ:BIN:BA SRC:(SYCND+BA)
16   LINK/EXE:BIN:BA.SYG BIN:BA
17   MACRO/OBJ:BIN:DX SRC:(SYCND+DX)
18   LINK/EXE:BIN:DX.SYG BIN:DX
19   MACRO/OBJ:BIN:RK SRC:(SYCND+RK)
20   LINK/EXE:BIN:RK.SYG BIN:RK
21   MACRO/OBJ:BIN:LP SRC:(SYCND+LP)
22   LINK/EXE:BIN:LP.SYG BIN:LP
```

Line 4 assembles the keyboard monitor. In the command line, MACRO/OBJ:BIN: invokes the MACRO assembler and specifies the output device for the object file that results from this operation. BIN: is the device; note that Line 2 assigns RK0: to BIN:. Therefore, if you were to invoke MONBLD, the object file would go to RK0:.

KMFB is the output file name. The file type is .OBJ, since MACRO creates object files by default.

On the input side of the command string, SRC: is the input device. Line 1 assigns RK1: to SRC:. The parentheses contain the source files to be assembled. The source input files in this command string are FB.MAC, SYCND.MAC, EDTGBL.MAC, KMON.MAC and KMOVLY.MAC. The system assumes .MAC files as input to MACRO. Because the files are separated by plus signs (rather than commas), one object file (KMFB.OBJ) results from this assembly.

MONBLD contains commands similar to the ones in lines 8 and 9 only if you are generating a multi-terminal system.

Lines 10, 11, and 12 link the monitor object modules. In the command string, LINK invokes the linker. The option /EXE:BIN:RT11FB.SYG specifies a device (BIN:) and file name (RT11FB.SYG) for the executable file that results from this operation. The monitor created by this example is RT11FB, but it has the .SYG file type since it is created through the system generation process. The option /BOU:1000 specifies the address boundary on which LINK should start a program section to be named when LINK asks for its name (Line 12). /PROMPT allows additional lines to be entered when the command string is too long to fit on one line. If you were to execute MONBLD, the system would prompt with an asterisk (*) at Line 11. Additional lines can be entered until the double slashes (//) indicate the end of the command string.

On the input side of the command, BIN: is the input device. The default LINK input file type is .OBJ. BTFB is the object file BTFB.OBJ that results from the assembly performed at Line 7; RMFB is the object file RMFB.OBJ that results from the assembly performed at Line 5; and so on.

Line 12 is the name of the program section whose address boundary is specified in the /BOU:1000 option. This line responds to the system prompt **Boundary section?** which would appear (if you were to execute MONBLD) once the command string is completed.

The command lines in the versions of MONBLD and DEVBLD that result from your SYSGEN session are variations of these commands. You can identify all the source files that you will need for your assemblies if you study the command files.

When you actually assemble components, you will type each command line in the indirect file. Before typing the command, however, you must copy to the system diskette the source files you need to perform that assembly. When you type the command, you must substitute specific devices (DX0:, PD0:, or DY0:) for SRC: and (DX1:, PD1: or DY1:) for BIN:. You will not make the device assignments in Lines 1, 2, 3, 13, and 14.

10.3.2 System Build Diskettes

Create four diskettes for the system build procedure. SYCND.MAC and SYSTBL.MAC should be on your system diskette (since you have run SYSGEN). The other source files are on your backup distribution diskettes.

When you initialize each diskette, give it an identifying volume ID. For example, to initialize Diskette 1, type:

```
.INITIALIZE/BAD/VOL xx1:(RET)
xx1:/Initialize; Are you sure? Y(RET)
?DUP-I-No bad blocks detected xx1:
Volume ID? DISKETTE 1(RET)
Owner? (RET)
.
```

Then, you can easily identify each diskette. Type:

```
.DIRECTORY/VOL(RET)
DISKETTE 1
.
```

The four diskettes for this procedure should contain the files (but *only* the files) in the following lists.

Diskette 1
The system diskette for the assembly procedure should contain:

RT11SJ.SYS
xx.SYS (DX.SYS, PD.SYS, DY.SYS)
SWAP.SYS
TT.SYS
LP.SYS
MACRO.SAV
SYCND.MAC (from the SYSGEN session you just performed)
PIP.SAV
DUP.SAV
DIR.SAV
SYSMAC.SML
yy.MAC (where yy is the monitor you are building, SJ, FB, or XM)
EDTGBL.MAC
KMON.MAC

Diskette 2
The second diskette will store two .MAC source files for the second assembly. It will also receive .OBJ files produced during assemblies. It should contain:

USR.MAC
RMONxx.MAC (yy is SJ or FB)

Diskette 3
The third diskette will store most of the .MAC source files for the assemblies until they are needed on the system diskette. It must contain:

xx.MAC	(DX.MAC, PD.MAC, DY.MAC)
SYSTBL.MAC	(from the SYSGEN session you just performed)
BSTRAP.MAC	
zz.MAC	(zz is device name for support specified during SYSGEN)

.
.
.

zz.MAC (zz is device name for support specified during SYSGEN)
other source files, except KMOVLY.MAC (refer to MONBLD and DEVBLD)
LINK.SAV

Diskette 4
The fourth diskette should contain the KMON overlays for the KMON assembly. It will receive the KMON object file, and you will copy the other object files to it as you assemble them on Diskette 2. Diskette 4 should contain:

KMOVLY.MAC

Review the installation procedures in Chapter 3 or 6, if necessary, when you create these diskettes. If you run out of room when you create a diskette, squeeze the diskette to consolidate free space. When you copy files from nonbootable diskettes, use the COPY/SYSTEM/WAIT command. For example, type:

```
.COPY/SYSTEM/WAIT xx1:filnam.typ xx0:filnam.typ(RET)
Mount input volume in xx1:; Continue?
```

Place the diskette that contains the file you want to copy in Unit 1.

```
Y(RET)

Mount output volume in xx0:; Continue?
```

Replace the system diskette in Unit 0 with the diskette to which you want to copy filnam.typ.

```
Y(RET)

Mount system volume in xx0:; Continue?
```

Replace the system diskette in Unit 0.

```
Y(RET)
```

Do not forget to copy the bootstrap to Diskette 1 and to squeeze the diskettes to consolidate free space. Do not use diskettes with bad blocks.

Use the hardware bootstrap to boot Diskette 1, and set the date and time.

```
.DATE dd-mmm-yy(RET)
.TIME hh:mm:ss(RET)
.
```

10.4 Assembling and Linking Monitor and Handlers

Now that your files are arranged on the four diskettes, assign the logical name WF: to Unit 1 (RX01, PDT–11, or RX02). The MACRO assembler uses a work file during some assemblies.

```
.ASSIGN xx1: WF:(RET)
.
```

Insert Diskette 4 in diskette Unit 1. Refer to your listing of MONBLD.COM for the first command. Ignore the command lines that begin with ASSIGN.

When you type the command, you will substitute DXn:, PDn:, or DYn: for BIN: and SRC: as shown in the following example. You must also make other adjustments to this first command line because of the special size constraints of the keyboard monitor assembly (always the first assembly). For example, you must add the ALLOCATE:150 option to the MACRO command, and you must indicate that KMOVLY resides on the diskette in Unit 1. The source files required for this assembly are on Diskette 1 and 4.

The command varies according to type of monitor being built. For example, if you were building a foreground/background monitor, the first assembly command in MONBLD would be Example 1 below. You would type Example 2. Study this example and your MONBLD listing; then execute the first assembly. Do not forget to terminate every line with RETURN.

Example 1

```
MACRO/OBJ:BIN:KMFB SRC:(FB+SYCND+EDTGBL+KMON+KMOVLY)
```

Example 2

```
.MACRO/OBJ:xx1:KMFB/ALLOCATE:150 xx0:(FB+SYCND+EDTGBL+KMON)+xx1:KMOVLY(RET)
ERRORS DETECTED:  0
.
```

NOTE

Refer to Table 10–2 if errors occur during any assemblies.

The first assembly will take as long as half an hour. The exact amount of time depends on the amount of memory — the more memory in your configuration, the faster the assembly.

When the system completes the assembly, delete the source files KMON and KMOVLY (see the following examples) from Diskette 1 and Diskette 4. Leave SYCND and EDTGBL on Diskette 1; you need SYCND for all assemblies and EDTGBL for most assemblies. After you delete these files, squeeze Diskette 1 to consolidate free space.

```
.DELETE/NOQUERY xx0:KMON.MAC(RET)
.DELETE/NOQUERY xx1:KMOVLY.MAC(RET)
.SQUEEZE xx0:(RET)
xx0:/Squeeze; Are you sure? Y(RET)
RT-11SJ V04-00x
(Followed by any startup file commands.)
.
```

Replace Diskette 4 in Unit 1 with Diskette 2. Look at the next command line in MONBLD. Among the source files you need for the next assembly are USR.MAC and RMONxx.MAC (where xx is SJ or FB). These two source files are on Diskette 2. The other source files you need for this assembly are on Diskette 1. Since some of the source files are on Diskette 1 and some are on Diskette 2, you must alter the command line appropriately. For example, if the command line were Example 1, you would type Example 2.

Example 1

```
MACRO/OBJ:BIN:RMFB SRC:(FB+SYCND+EDTGBL+USR+RMONFB)
```

Example 2

```
.MACRO/OBJ:xx1:RMFB xx0:(FB+SYCND+EDTGBL)+xx1:(USR+RMONFB)(RET)
ERRORS DETECTED:   0
.
```

Delete the two source files from Diskette 2 in Unit 1. For example:

```
.DELETE/NOQUERY xx1:(USR,RMONFB).MAC(RET)
.
```

Copy the object (from this assembly) from Diskette 2 to Diskette 4.

```
.COPY/WAIT xx1:RMFB.OBJ xx0:RMFB.OBJ(RET)
Mount input volume in xx1:; Continue? Y(RET)
Mount output volume in xx0:; Continue? Y(RET)
Mount system volume in xx0:; Continue?Y(RET)
```

Then delete the object file from Diskette 2.

```
.DELETE/NOQUERY xx1:RMFB.OBJ(RET)
.
```

Look at the next command line in MONBLD and copy the source files you need for that assembly from Diskette 3 in Unit 1 to Diskette 1 in Unit 0. For example, type:

```
.COPY xx1:SYSTBL.MAC xx0:SYSTBL.MAC(RET)
.
```

Replace Diskette 3 in Unit 1 with Diskette 2. Enter the command line from MONBLD, making the required substitutions. For example, if the command line were Example 1, you would type example 2.

Example 1

```
MACRO/OBJ:BIN:TBFB SRC:(FB+SYCND+EDTGBL+SYSTBL)
```

Example 2

```
.MACRO/OBJ:xx1:TBFB xx0:(FB+SYCND+EDTGBL+SYSTBL)(RET)
ERRORS DETECTED: 0
.
```

Then, delete the source file (that you previously copied) from Diskette 1.

```
.DELETE/NOQUERY xx0:SYSTBL.MAC(RET)
.SQUEEZE xx0:(RET)
```

Copy the object file created during this assembly from Diskette 2 to Diskette 4 and delete it from Diskette 2.

```
.COPY/WAIT xx1:TBFB.OBJ xx0:TBFB.OBJ(RET)
Mount input volume in xx1:; Continue? Y(RET)
Mount output volume in xx0:; Continue? Y(RET)
Mount system volume in xx0:; Continue?Y(RET)
.DELETE/NOQUERY xx1:TBFB.OBJ(RET)
.
```

Use the same technique to perform all the assemblies in MONBLD and DEVBLD (but do not perform the link operations yet). Copy source files from Diskette 3 to Diskette 1 and delete them from Diskette 1 after the assembly is complete. After you perform each assembly, be sure to copy the object file to Diskette 4 and delete it from Diskette 2.

Ignore the ASSIGN commands. When you do perform the link operations, the object files will already be on Diskette 4. The following list summarizes the sequence of operations for each subsequent assembly.

1. Examine the next command line. If it invokes MACRO, establish which source files you need for the assembly. If the command does not invoke MACRO, ignore the command line.
2. Copy these source files from Diskette 3 in Unit 1 to Diskette 1 in Unit 0.
3. Replace Diskette 3 in Unit 1 with Diskette 2.
4. Type the command, substituting xx0: (where xx is DX, PD, or DY) for SRC: and xx1: for BIN:.
5. Delete the source files you copied in step 2 from Diskette 1, unless you need them for the next assembly.
6. Copy the object file from Diskette 2 to Diskette 4.
7. Delete the object file from Diskette 2.
8. Repeat steps 1 through 7 for each assembly command line.

Now you are ready to perform the link operations.

Copy LINK from Diskette 3 to Diskette 1.

```
.COPY xx1:LINK.SAV xx0:(RET)
.
```

Mount Diskette 4 in Unit 1. Find the monitor LINK command in MONBLD. Type the command, substituting xx1: for BIN:. Substitute LP: for MAP: if you want to send the link map to the line printer. Substitute TT: for MAP: if you want to send the link map to the terminal. For example, if the command line were Example 1, you would type Example 2.

NOTE

Save the link map that results. If you ever need to submit an SPR to DIGITAL for your specially generated monitor, you must include the monitor link map with the SPR.

Example 1

```
LINK/EXE:BIN:RT11FB.SYG/BOU:1000/PROMPT/MAP:MAP:RT11FB BIN:BTFB
BIN:RMFB,KMFB,MEFB,MIFB,TBFB//
OVLY0
```

Example 2

```
.LINK/EXE:xx1:RT11FB.SYG/BOU:1000/PROMPT/MAP:LP:RT11FB xx1:BTFBRET
*BIN:RMFB,KMFB,MEFB,MIFB,TBFB//RET
Boundary section? OVLY0RET
```

Now find the next LINK command, in DEVBLD. The LINK commands in DEVBLD link the device handlers for which you requested support when you ran SYSGEN. Type all the link commands, making substitutions for BIN:.

The resulting linked components on Diskette 4 have the file type .SYG. The .SYG files are masters for your generated monitors and handlers, and you should preserve them. Back up the generated system and store the backup.

Copy the .SYG files to another diskette, along with any other components (for example, utility programs) you need in your working system. Refer to the appropriate installation chapter, if necessary, to install the generated system. Rename the .SYG files , monitors and handlers, to .SYS. You can also rename the monitor to distinguish it from standard monitors. You will have to copy the bootstrap for the generated monitor to your working system device if you want that monitor to boot. Note that you can always identify a user-generated monitor by the (S) that prints in the boot message.

Once you have built the system, store the conditional, command, and output files from the generation process. You must retain at least the conditional and command files, since all SPRs must be accompanied by a listing of these files, as well as the link maps, if a custom monitor is used. In addition, if you keep the object files, you can reassemble a particular component and relink it, if it ever requires a patch; you do not need to rebuild the entire system each time you make a patch.

NOTE

DIGITAL strongly recommends that you keep the object files to facilitate patching. Some patches are distributed in source form, and often, only one system component requires a patch. You can assemble just that one component and link the new object file for it with the old object files from your original system build.

Table 10–2: System Build Errors

Type of Error	Cause[1]
Assembly errors	Incorrect or conflicting responses to SYSGEN dialogue. Reexamine your responses to SYSGEN. Often, reading the conditional files helps you spot errors. If the SYSGEN responses seem correct, make sure that you have the proper versions of the source files on the source disk. Check that you have not introduced an error if you edited the source files. Compare the files on the source disk to the originals in the distribution.
?MACRO-F-File not found DEV:FILNAM.TYP	Failure to copy the SYCND.MAC or SYSTBL.MAC files to the source disk or failure to include all necessary source modules on the disk.
?MACRO-F-Output device full on DEV:FILNAM.TYP	You do not have sufficient space on the binary output device to accommodate all output files.
?MACRO-F-I/O error on workfile	You do not have sufficient space on the binary output device. Check the following possibilities: You performed system build manually but did not assign the workfile (or the workfile needs more space). You did not squeeze the binary device before starting the build. Try squeezing it and rebuilding.
?MACRO-F-I/O error on DEV:FILNAM.TYP	Bad volumes or write protected devices. If the device is write protected, enable it. If the volume is bad, try another.

[1] If you encounter an error you cannot explain or correct, send an SPR to DIGITAL with a listing of the conditional files SYCND.MAC and SYSTBL.MAC, along with the console output.

Appendix A
SYSGEN Answers That Will Duplicate the Standard Monitors

Sections A.1 and A.2 list the answers DIGITAL gave to create the standard (distributed) monitors. Section A.3 lists all the SYSGEN questions, without the descriptive text.

A.1 Base-Line Single-Job Monitor

1	N	40	skipped	79	skipped	118	skipped
2	Y	41	skipped	80	skipped	119	skipped
3	skipped	42	skipped	81	skipped	120	skipped
4	skipped	43	skipped	82	skipped	121	skipped
5	skipped	44	skipped	83	skipped	122	skipped
6	skipped	45	skipped	84	skipped	123	skipped
7	skipped	46	skipped	85	skipped	124	skipped
8	skipped	47	skipped	86	skipped	125	skipped
9	skipped	48	skipped	87	skipped	126	skipped
10	skipped	49	skipped	88	skipped	127	skipped
11	skipped	50	skipped	89	Y	128	skipped
12	skipped	51	Y	90	N	129	skipped
13	skipped	52	skipped	91	skipped	130	skipped
14	skipped	53	skipped	92	skipped	131	skipped
15	40	54	skipped	93	Y	132	skipped
16	134	55	skipped	94	176500	133	skipped
17	N	56	Y	95	300	134	skipped
18	Y	57	skipped	96	Y	135	skipped
19	skipped	58	skipped	97	skipped	136	skipped
20	skipped	59	skipped	98	skipped	137	skipped
21	skipped	60	skipped	99	Y	138	skipped
22	N	61	skipped	100	skipped	139	skipped
23	skipped	62	skipped	101	skipped	140	skipped
24	Y	63	skipped	102	skipped	141	skipped
25	N	64	skipped	103	skipped	142	skipped
26	skipped	65	skipped	104	0	143	skipped
27	skipped	66	skipped	105	skipped	144	skipped
28	skipped	67	skipped	106	skipped	145	skipped
29	skipped	68	skipped	107	skipped	146	skipped
30	skipped	69	skipped	108	skipped	147	skipped
31	Y	70	skipped	109	skipped	148	skipped
32	N	71	skipped	110	skipped	149	skipped
33	177170	72	skipped	111	skipped	150	skipped
34	264	73	skipped	112	skipped	151	skipped
35	skipped	74	skipped	113	skipped	152	RK1
36	skipped	75	skipped	114	skipped	153	RK0
37	skipped	76	skipped	115	skipped	154	TT
38	skipped	77	skipped	116	skipped	155	Y
39	skipped	78	skipped	117	skipped		

A.2 Single-Job and Foreground/Background Monitors

1	N
2	N
3	Y
4	Y
5	N
6	N
7	N
8	Y
9	N
10	N
11	N
12	N
13	skipped
14	skipped
15	40
16	134
17	N
18	Y
19	skipped
20	skipped
21	skipped
22	N
23	N
24	Y
25	Y
26	N
27	Y
28	Y
29	N
30	skipped
31	Y
32	N
33	177170
34	264
35	skipped
36	skipped
37	Y
38	N
39	N
40	177170
41	264
42	skipped
43	skipped
44	Y
45	N
46	176500
47	300
48	skipped
49	skipped
50	Y
51	Y
52	Y
53	2
54	Y
55	Y
56	Y
57	Y
58	2
59	Y
60	N
61	Y
62	Y
63	Y
64	Y
65	2
66	Y
67	Y
68	2
69	Y
70	Y
71	1
72	172522
73	224
74	skipped
75	skipped
76	skipped
77	skipped
78	skipped
79	skipped
80	skipped
81	skipped
82	skipped
83	skipped
84	skipped
85	skipped
86	skipped
87	skipped
88	Y
89	Y
90	N
91	skipped
92	skipped
93	Y
94	176500
95	300
96	Y
97	skipped
98	Y
99	Y
100	Y
101	N
102	172000
103	320
104	0
105	skipped
106	skipped
107	skipped
108	skipped
109	skipped
110	skipped
111	skipped
112	skipped
113	skipped
114	skipped
115	skipped
116	skipped
117	skipped
118	skipped
119	skipped
120	skipped
121	skipped
122	skipped
123	skipped
124	skipped
125	skipped
126	skipped
127	skipped
128	skipped
129	skipped
130	skipped
131	skipped
132	skipped
133	skipped
134	skipped
135	skipped
136	skipped
137	skipped
138	skipped
139	skipped
140	skipped
141	skipped
142	skipped
143	skipped
144	skipped
145	skipped
146	skipped
147	skipped
148	skipped
149	skipped
150	skipped
151	skipped
152	RK1
153	RK0
154	TT
155	Y

A.3 SYSGEN Questions

```
 1.    Do you want the long form of the dialogue [Y/N] (Y)?

 2.    Do you want the base-line single-job (BL)
       monitor [Y/N] (N)?

 3.    Do you want the single-job (SJ) monitor [Y/N] (Y)?

 4.    Do you want the foreground/background (FB)
       monitor [Y/N] (Y)?

 5.    Do you want the extended memory (XM)
       monitor [Y/N] (N)?

 6.    Do you want timer support in the SJ monitor [Y/N] (N)?

 7.    Do you want device time-out support [Y/N] (N)?

 8.    Do you want an error message on system I/O
       errors [Y/N] (Y)?

 9.    Do you want system job support [Y/N] (N)?

10.    Do you want to use the .SPCPS request [Y/N] (N)?

11.    Do you want idle loop light pattern [Y/N] (N)?

12.    Do you want multi-terminal support [Y/N] (N)?

13.    Do you want asynchronous terminal status [Y/N] (Y)?

14.    Do you want multi-terminal time-out support [Y/N] (Y)?

15.    Size of the output buffers in characters [D] (40)?

16.    Size of the input buffers in characters [D] (134)?

17     Do you want end of month and year date
       rollover [Y/N] (N)?

18.    Do you want all the keyboard monitor
       commands [Y/N] (Y)?

19.    Do you want the UTILITY PROGRAM subset of keyboard
       monitor commands [Y/N] (Y)?

20.    Do you want the LANGUAGE subset of keyboard monitor
       commands [Y/N] (Y)?

21.    Do you want the MINIMAL subset of keyboard monitor
       commands [Y/N] (Y)?

22.    Do you want the optional 50 Hz clock
       support [Y/N] (N)?

23.    Do you want to use the KW11-P clock as the system
       clock [Y/N] (N)?

24.    Do you want the startup indirect file [Y/N] (Y)?

25.    Do you want floating point support [Y/N] (N)?
```

```
26.   Do you want memory parity support [Y/N] (N)?

27.   Do you want power failure messages [Y/N] (Y)?

28.   Do you want BATCH support [Y/N] (N)?

29.   Do you want error logging [Y/N] (N)?

30.   How many device units does error log job
      support [D] (10)?

31.   Do you want RX11/RX01 single-density flexible
      diskette support [Y/N] (N)?

32    Do you want support for a second RX11
      controller [Y/N] (N)?

33.   What is the CSR address for the first RX11
      controller [O] (177170)?

34.   What is the vector address for the first
      RX11 controller [O] (264)?

35.   What is the CSR address for the second RX11
      controller [O] (177174)?

36.   What is the vector address for the second
      RX11 controller [O] (270)?

37.   Do you want RX211/RX02 double-density flexible
      diskette support [Y/N] (N)?

38.   Do you want support for a second RX02
      controller [Y/N] (N)?

39.   Do you want RX02 double density only
      support [Y/N] (N)?

40.   What is the CSR address for the first RX02
      controller [O] (177170)?

41.   What is the vector address for the first
      RX02 controller [O] (264)?

42.   What is the CSR address for the second RX02
      controller [O] (177150)?

43.   What is the vector address for the second RX02
      controller [O] (270)?

44.   Do you want TU58 DECtape II cartridge support [Y/N] (N)?


45.   Do you want support for a second DECtape II
      controller [Y/N] (N)?

46.   What is the CSR address for the first DECtape II
      controller [O] (176500)?

47.   What is the vector address for the first DECtape II
      controller [O] (300)?

48.   What is the CSR address for the second DECtape II
      controller [O] (176510)?
```

```
49.   What is the vector address for the second DECtape II
      controller [O] (310)?

50.   Do you want support for the PDT-11 series [Y/N] (N)?

51.   Do you want TC11 DECtape support [Y/N] (N)?

52.   Do you want RF11 fixed-head disk support [Y/N] (N)?

53.   How many disk platters are installed on the
      RF11 controller [D] (1)?

54.   Do you want RJS03 or RJS04 disk support [Y/N] (N)?

55.   Do you want RJS03 rather than RJS04 support [Y/N] (Y)?

56.   Do you want RK05/RK05F disk support [Y/N] (N)?
57.   Do you want RL01/RL02 disk support [Y/N] (N)?

58.   How many RL01/RL02 units are to be supported [D] (2)?

59    Do you want RP11/RPR02 or RP11/RP03 disk
      support [Y/N] (N)?

60.   Do you want RPR02 rather than RPR02/RP03
      support [Y/N] (Y)?

61.   Do you want RK06/RK07 disk support [Y/N] (N)?

62.   Do you want magnetic tape support [Y/N] (N)?

63.   Do you want TM11 (UNIBUS) magtape support [Y/N] (Y)?

64.   Do you want the file-structured magtape
      handler [Y/N] (Y)?

65.   How many magtape units are to be supported [D] (2)?

66.   Do you want TJU16 (MASSBUS) magtape
      support [Y/N] (N)?

67.   Do you want the file-structured magtape
      handler [Y/N] (Y)?

68.   How many magtape units are to be supported [D] (2)?

69.   Do you want TS11 (UNIBUS) magtape
      support [Y/N] (N)?

70.   Do you want the file-structured magtape
      handler [Y/N] (Y)?

71.   How many magtape units are to be supported [D] (2)?

72.   What is the address of the first
      unit [O] (172522)?

73.   What is the vector address of the first unit [O] (224)?

74.   What is the address of the second unit [O] (172526)?

75.   What is the vector address of the second
      unit [O] (300)?
```

```
76.   What is the address of the third unit [O] (172532)?

77.   What is the vector address of the third
      unit [O] (304)?

78.   What is the address of the fourth unit [O] (172536)?

79.   What is the vector address of the fourth
      unit [O] (310)?

80.   What is the address of the fifth unit [O] (172542)?

81.   What is the vector address of the fifth
      unit [O] (314)?

82.   What is the address of the sixth unit [O] (172546)?

83.   What is the vector address of the sixth
      unit [O] (320)?

84.   What is the address of the seventh unit [O] (172552)?

85.   What is the vector address of the seventh
      unit [O] (324)?

86.   What is the address of the eighth unit [O] (172556)?

87.   What is the vector address of the eighth
      unit [O] (330)?

88.   Do you want TA11 cassette support [Y/N] (N)?

89.   Do you want line printer support [Y/N] (Y)?

90.   Does your printer have a nonstandard vector or CSR
      address [Y/N] (N)?

91.   What is the CSR address for the printer [O] (177514)?

92.   What is the vector address for the printer [O] (200)?

93.   Do you want serial line printer support [Y/N] (N)?

94.   What is the CSR address for the serial line printer
      controller [O] (176500)?

95.   What is the vector address for the serial line printer
      controller [O] (300)?

96.   Do you want PC11 high-speed paper tape reader/punch
      support [Y/N] (N)?

97.   Do you want PR11 high-speed paper tape reader
      support [Y/N] (N)?

98.   Do you want card reader support [Y/N] (N)?

99.   Do you want the null handler [Y/N] (Y)?

100.  Do you want VT11 or VS60 graphics support [Y/N] (N)?

101.  Do you want VS60 support [Y/N] (N)?
```

```
102.  What is the CSR address for the
      VT11/VS60 [O] (172000)?

103.  What is the vector address for the
      VT11/VS60 [O] (320)?

104.  How many extra device slots do you want [D] (0)?

105.  How many local DL11 lines, including the console, are
      to be supported, [D] (1)?

106.  How many remote DL11 lines are to be
      supported [D] (0)?

107.  What is the CSR address for the first (console)
      DL11 [O] (177560)?

108.  What is the vector address for the first (console)
      DL11 [O] (60)?

109.  What is the CSR address for the next (local)
      DL11 [O] (176500)?

110.  What is the vector address for the next (local)
      DL11 [O] (300)?

111.  What is the CSR address for the next (remote)
      DL11 [O] (175610)?

112.  What is the vector address for the next (remote)
      DL11 [O] (300)?

113.  What is the CSR address for the next (local)
      DL11 [O] (176510)?

114.  What is the vector address for the next (local)
      DL11 [O] (310)?

115.  What is the CSR address for the next (remote)
      DL11 [O] (175620)?

116.  What is the vector address for the next (remote)
      DL11 [O] (310)?

117.  What is the CSR address for the next (local)
      DL11 [O] (176520)?

118.  What is the vector address for the next (local)
      DL11 [O] (320)?

119.  What is the CSR address for the next (remote)
      DL11 [O] (175630)?

120.  What is the vector address for the next (remote)
      DL11 [O] (320)?

121.  What is the CSR address for the next (local)
      DL11 [O] (176530)?

122.  What is the vector address for the next (local)
      DL11 [O] (330)?

123.  What is the CSR address for the next (remote)
      DL11 [O] (175640)?
```

124. What is the vector address for the next (remote)
DL11 [O] (330)?

125. What is the CSR address for the next (local)
DL11 [O] (176540)?

126. What is the vector address for the next (local)
DL11 [O] (340)?

127. What is the CSR address for the next (remote)
DL11 [O] (175650)?

128. What is the vector address for the next (remote)
DL11 [O] (340)?

129. What is the CSR address for the next (local)
DL11 [O] (176550)?

130. What is the vector address for the next (local)
DL11 [O] (350)?

131. What is the CSR address for the next (remote)
DL11 [O] (175660)?

132. What is the vector address for the next (remote)
DL11 [O] (350)?

133. What is the CSR address for the next (local)
DL11 [O] (176560)?

134. What is the vector address for the next (local)
DL11 [O] (360)?

135. What is the CSR address for the next (remote)
DL11 [O] (175670)?

136. What is the vector address for the next (remote)
DL11 [O] (360)?

137. Do you want DZ11 or DZV11 multiplexor
support [Y/N] (N)?

138. Do you want DZ11 multiplexor support [Y/N] (Y)?

139. How many DZ multiplexors are to be supported [D] (1)?

139. How many DZ multiplexors are to be supported [D] (1)?

140. How many local DZ lines are to be
supported altogether [D] (1)?

141. How many remote DZ lines are to be
supported altogether [D] (0)?

142. What is the CSR address for the
first DZ multiplexor [O] (160010)?

143. What is the vector address for the
first DZ multiplexor [O] (300)?

144. What is the CSR address for the
second DZ multiplexor [O] (160020)?

```
145.  What is the vector address for the
      second DZ multiplexor [0] (310)?

146.  What is the CSR address for the
      third DZ multiplexor [0] (160030)?

147.  What is the vector address for the
      third DZ multiplexor [0] (320)?

148.  What is the CSR address for the
      fourth DZ multiplexor [0] (160040)?

149.  What is the vector address for the
      fourth DZ multiplexor [0] (340)?

150.  Do you want the lines initialized
      to 300 baud [Y/N] (Y)?

151.  Do you want the lines initialized
      to 150 baud [Y/N] (Y)?

152.  What is the PHYSICAL name and unit of the source
      input device [xxn] (RK1)?

153.  What is the PHYSICAL name and unit of the binary
      output device [xxn] (RK0)?

154.  What is the PHYSICAL name and unit of the map
      output device [xxn] (TT)?

155.  Do you want to retain the system OBJs [Y/N] (Y)?
```

Appendix B
SYSGEN Answers That Create Example Systems

The following sections list SYSGEN answers that create example systems. These examples create an extended memory monitor and a multi-terminal system, neither of which is a standard distributed monitor. Note that these sections are intended to be *examples* and only examples. It is highly improbable that the examples would exactly meet your needs or that the CSR and vector addresses given in these examples would be the same for your hardware configuration. DIGITAL does not recommend using these examples as is. Refer to Appendix A for a list of the SYSGEN questions.

B.1 Answers to Create an Example Extended Memory Monitor

This example creates an extended memory monitor but none of the other monitors. Otherwise, it supports the same options and devices that the standard monitors support.

1	Y
2	N
3	N
4	N
5	Y
6	skipped
7	N
8	skipped
9	N
10	N
11	N
12	N
13	skipped
14	skipped
15	40
16	134
17	N
18	Y
19	skipped
20	skipped
21	skipped
22	N
23	N
24	Y
25	Y
26	N
27	Y
28	Y
29	N
30	skipped
31	Y
32	N
33	177170
34	264
35	skipped
36	skipped
37	Y
38	N
39	N
40	177170
41	264
42	skipped
43	skipped
44	Y
45	N
46	176500
47	300
48	skipped
49	skipped
50	Y
51	Y
52	Y
53	2
54	Y
55	Y
56	Y
57	Y
58	2
59	Y
60	N
61	Y
62	Y
63	Y
64	Y
65	2
66	Y
67	Y
68	2
69	Y
70	Y
71	1
72	172522
73	224
74	skipped
75	skipped
76	skipped
77	skipped
78	skipped
79	skipped
80	skipped
81	skipped
82	skipped
83	skipped
84	skipped
85	skipped
86	skipped
87	skipped
88	Y
89	Y
90	N
91	skipped
92	skipped
93	Y
94	176500
95	300
96	Y
97	skipped
98	Y
99	Y
100	Y
101	N
102	172000
103	320
104	0
105	skipped
106	skipped
107	skipped
108	skipped
109	skipped
110	skipped
111	skipped
112	skipped
113	skipped
114	skipped
115	skipped
116	skipped
117	skipped
118	skipped
119	skipped
120	skipped
121	skipped
122	skipped
123	skipped
124	skipped
125	skipped
126	skipped
127	skipped
128	skipped
129	skipped
130	skipped
131	skipped
132	skipped
133	skipped
134	skipped
135	skipped
136	skipped
137	skipped
138	skipped
139	skipped
140	skipped
141	skipped
142	skipped
143	skipped
144	skipped
145	skipped
146	skipped
147	skipped
148	skipped
149	skipped
150	skipped
151	skipped
152	RK1
153	RK0
154	TT
155	Y

B.2 Answers to Create an Example Multi-terminal System

This example creates a foreground/background multi-terminal system. Device support includes RK05 disk, RX01 diskette, and eight terminals with a DZ11–A 8-line multiplexor.

1	Y	40	skipped	79	skipped	118	skipped
2	N	41	skipped	80	skipped	119	skipped
3	N	42	skipped	81	skipped	120	skipped
4	Y	43	skipped	82	skipped	121	skipped
5	N	44	N	83	skipped	122	skipped
6	skipped	45	skipped	84	skipped	123	skipped
7	N	46	skipped	85	skipped	124	skipped
8	skipped	47	skipped	86	skipped	125	skipped
9	N	48	skipped	87	skipped	126	skipped
10	N	49	skipped	88	N	127	skipped
11	N	50	N	89	Y	128	skipped
12	Y	51	N	90	N	129	skipped
13	Y	52	N	91	skipped	130	skipped
14	Y	53	skipped	92	skipped	131	skipped
15	40	54	N	93	N	132	skipped
16	134	55	skipped	94	skipped	133	skipped
17	N	56	Y	95	skipped	134	skipped
18	Y	57	N	96	N	135	skipped
19	skipped	58	skipped	97	N	136	skipped
20	skipped	59	N	98	N	137	Y
21	skipped	60	skipped	99	N	138	Y
22	N	61	N	100	skipped	139	1
23	N	62	N	101	skipped	140	8
24	Y	63	skipped	102	skipped	141	0
25	Y	64	skipped	103	skipped	142	160100
26	N	65	skipped	104	5	143	300
27	Y	66	skipped	105	1	144	skipped
28	Y	67	skipped	106	0	145	skipped
29	N	68	skipped	107	177560	146	skipped
30	skipped	69	skipped	108	60	147	skipped
31	Y	70	skipped	109	skipped	148	skipped
32	N	71	skipped	110	skipped	149	skipped
33	177170	72	skipped	111	skipped	150	Y
34	264	73	skipped	112	skipped	151	skipped
35	skipped	74	skipped	113	skipped	152	RK1
36	skipped	75	skipped	114	skipped	153	RK0
37	N	76	skipped	115	skipped	154	TT
38	skipped	77	skipped	116	skipped	155	Y
39	skipped	78	skipped	117	skipped		

Appendix C
Building System Programs

If your application requires changes to RT–11 system programs other than the monitors and device handlers, you will need the information in this appendix to assemble and link the components. Use RT–11 MACRO as the assembler and RT–11 LINK as the linker for all components. Make sure that all assemblies (except ODT and VDT) and links (except where otherwise noted) are error free.

NOTE

> The default system library, SYSLIB.OBJ, is required to link many of these components.

The following sections contain BATCH streams that generate the following components:

- BATCH
- BINCOM
- CREF
- DIR
- DUMP
- DUP
- EDIT
- Error Logging
- FILEX
- FORMAT
- HELP
- LIBR
- LINK
- MACRO and MAC8K
- MDUP, MBOOT, and MSBOOT
- ODT
- PAT
- PATCH
- PIP
- QUEUE

- RESORC
- SIPP
- SLP
- SRCCOM
- SWAP
- SYSGEN
- SYSTEM Library
- System MACRO library
- TECO
- Utilities Library
- VT11 Library and PLOT55

You can use indirect command files or direct keyboard monitor commands if you do not want to use BATCH, as long as you use equivalent commands. However, you should be familiar with the following conventions used in these listings.

1. Default file types are not explicitly specified. The file type for source files is .MAC; for assembler output, .OBJ; and for linker output, .SAV.
2. The system macro library, SYSMAC.SML, must be on the system device during all these assemblies.
3. Sources are assumed to reside on logical device SRC:, binary output to go to BIN:, object files to go to OBJ:, and listing and map files to go to LST:. You can use any appropriate device for these files.
4. SYSLIB refers to a system library that contains only the FORTRAN OTS routines.

You can alter the command strings in these listings to take full advantage of all RT–11 MACRO and LINK commands. In addition, you can substitute equivalent monitor commands for the CSI level commands shown here.

The indirect command files that SYSGEN creates contain assembly and link instructions for the monitor and handlers. Examine MONBLD.COM and DEVBLD.COM after a SYSGEN run for assembly and link instructions.

C.1 BATCH

```
$JOB/RT11/TIME  ! ABATCH.BAT
        TTYIO
$MES ASSEMBLE BATCH
.R MACRO
*OBJ:BATCH,LST:BATCH/L:MEB/C=SRC:BATCH
$EOJ

$JOB/RT11/TIME  ! LBATCH.BAT
        TTYIO
$MES LINK BATCH
.R LINK
*BIN:BATCH,MAP:BATCH/W=OBJ:BATCH
$EOJ
```

C.2 BINCOM

```
$JOB/RT11/TIME   ! ABINCO.BAT
        TTYIO
$MES ASSEMBLE BINCOM
.R MACRO
*OBJ:BINCOM,LST:BINCOM/C=SRC:ULBMAC,BINCOM
$EOJ

$JOB/RT11/TIME   ! LBINCO.BAT
        TTYIO
$MES LINK BINCOM
.R LINK
*BIN:BINCOM,MAP:BINCOM/W=OBJ:BINCOM,BIN:ULBLIB
$EOJ
```

C.3 CREF

```
$JOB/RT11/TIME   ! ACREF.BAT
        TTYIO
$MES ASSEMBLE CREF
.R MACRO
*OBJ:CREF,LST:CREF/C=SRC:CREF
$EOJ

$JOB/RT11/TIME   ! LCREF.BAT
        TTYIO
$MES LINK CREF
.R LINK
*BIN:CREF,MAP:CREF/W=OBJ:CREF
$EOJ
```

C.4 DIR

```
$JOB/RT11/TIME   ! ADIR.BAT
        TTYIO
$MES ASSEMBLE DIR
.R MACRO
*OBJ:DIR1ST,LST:DIR1ST/C=SRC:DIR1ST
*OBJ:DIRDAT,LST:DIRDAT/C=SRC:DIRPRE,DIRDAT
*OBJ:DIRMAN,LST:DIRMAN/C=SRC:DIRPRE,DIRMAN
*OBJ:DIRT11,LST:DIRT11/C=SRC:DIRPRE,DIRT11
*OBJ:DIRSWT,LST:DIRSWT/C=SRC:DIRPRE,DIRSWT
*OBJ:DIRMAT,LST:DIRMAT/C=SRC:DIRPRE,DIRMAT
*OBJ:DIRDK,LST:DIRDK/C=SRC:DIRPRE,DIRDK
*OBJ:DIRMT,LST:DIRMT/C=SRC:DIRPRE,DIRMT
*OBJ:DIRCT,LST:DIRCT/C=SRC:DIRPRE,DIRCT
*OBJ:DIRSUP,LST:DIRSUP/C=SRC:DIRPRE,DIRSUP
*OBJ:DIRSRT,LST:DIRSRT/C=SRC:DIRPRE,DIRSRT
*OBJ:DIROUT,LST:DIROUT/C=SRC:DIRPRE,DIROUT
*OBJ:DIRERR,LST:DIRERR/C=SRC:DIRPRE,DIRERR
$EOJ

$JOB/RT11/TIME   ! LDIR.BAT
        TTYIO
$MES LINK DIR
.R LINK
*BIN:DIR,MAP:DIR/W=OBJ:DIR1ST,DIRDAT,DIRMAN,DIRT11,DIRSWT,DIRMAT//
*OBJ:DIRDK,DIRMT,DIRCT,DIRSUP,DIRSRT,DIROUT
*OBJ:DIRERR,BIN:ULBLIB
*//
$EOJ
```

C.5 DUMP

```
$JOB/RT11/TIME   ! ADUMP.BAT
        TTYIO
$MES ASSEMBLE DUMP
.R MACRO
*OBJ:DUMP,LST:DUMP/C=SRC:DUMP
$EOJ

$JOB/RT11/TIME   ! LDUMP.BAT
        TTYIO
$MES LINK DUMP
.R LINK
*BIN:DUMP,MAP:DUMP/W=OBJ:DUMP
$EOJ
```

C.6 DUP

```
$JOB/RT11/TIME   ! ADUP.BAT
        TTYIO
$MES ASSEMBLE DUP
.R MACRO
*OBJ:DUPROT,LST:DUPROT/C=SRC:DUPROT
*OBJ:DUPOV1,LST:DUPOV1/C=SRC:DUPPRE,DUPOV1
*OBJ:DUPIN1,LST:DUPIN1/C=SRC:DUPPRE,DUPIN1
*OBJ:DUPIN2,LST:DUPIN2/C=SRC:DUPPRE,DUPIN2
*OBJ:DUPOV2,LST:DUPOV2/C=SRC:DUPPRE,DUPOV2
*OBJ:DUPCRE,LST:DUPCRE/C=SRC:DUPPRE,DUPCRE
*OBJ:DUPUNI,LST:DUPUNI/C=SRC:DUPPRE,DUPUNI
*OBJ:DUPIMA,LST:DUPIMA/C=SRC:DUPPRE,DUPIMA
*OBJ:DUPSCN,LST:DUPSCN/C=SRC:DUPPRE,DUPSCN,DUPMRG
*OBJ:DUPOV5,LST:DUPOV5/C=SRC:DUPPRE,DUPOV5
*OBJ:DUPBOT,LST:DUPBOT/C=SRC:DUPPRE,DUPBOT
*OBJ:DUPWBT,LST:DUPWBT/C=SRC:DUPPRE,DUPWBT
*OBJ:DUPSQU,LST:DUPSQU/C=SRC:DUPPRE,DUPSQU
*OBJ:DUPOV7,LST:DUPOV7/C=SRC:DUPPRE,DUPOV7
*OBJ:DUPVOL,LST:DUPVOL/C=SRC:DUPPRE,DUPVOL
*OBJ:DUPZMC,LST:DUPZMC/C=SRC:DUPPRE,DUPZMC
*OBJ:DUPZRO,LST:DUPZRO/C=SRC:DUPPRE,DUPZRO,DUPMRG
$EOJ

$JOB/RT11/TIME   ! LDUP.BAT
        TTYIO
$MES LINK DUP
.R LINK
*BIN:DUP,MAP:DUP/W=OBJ:DUPROT,BIN:ULBLIB//
*OBJ:DUPOV1,DUPIN1,DUPIN2/O:1
*OBJ:DUPOV2,DUPCRE,DUPUNI/O:1
*OBJ:DUPIMA/O:1
*OBJ:DUPSCN/O:1
*OBJ:DUPOV5,DUPBOT,DUPWBT/O:1
*OBJ:DUPSQU/O:1
*OBJ:DUPOV7,DUPVOL,DUPZMC/O:1
*OBJ:DUPZRO/O:1
*//
$EOJ
```

C.7 EDIT

```
$JOB/RT11/TIME   ! AEDIT.BAT
        TTYIO
$MES ASSEMBLE EDIT
.R MACRO
*OBJ:VTCED1,LST:VTCED1/C=SRC:EDITDF,VTCAL1
*OBJ:VTCED4,LST:VTCED4/C=SRC:EDITDF/P:1,VTCAL4
*OBJ:VTBEDT,LST:VTBEDT/C=SRC:EDITDF/P:1,VTBASE
*OBJ:EDIT,LST:EDIT/C=SRC:VTMAC,EDIT
$EOJ

$JOB/RT11/TIME   ! LEDIT.BAT
        TTYIO
$MES LINK EDIT
.R LINK
*BIN:EDIT,MAP:EDIT/W=OBJ:VTCED1,VTCED4,VTBEDT,EDIT
$EOJ
```

C.8 Error Logging

```
$JOB/RT11/TIME   ! AERLOG.BAT
        TTYIO
$MES ASSEMBLE ERROR LOGGING
.R MACRO
*OBJ:ELCOPY,LST:ELCOPY/C=SRC:ELCOPY
*OBJ:ELTASK,LST:ELTASK/C=SRC:ELTASK
*OBJ:ELINIT,LST:ELINIT/C=SRC:ELINIT
*OBJ:ERROUT,LST:ERROUT/C=SRC:ERROUT
*OBJ:ERRTXT,LST:ERRTXT/C=SRC:ERRTXT
$EOJ

$JOB/RT11/TIME   ! LERLOG.BAT
        TTYIO
$MES LINK ERROR LOGGER
.R LINK
*BIN:EL,MAP:EL/W=OBJ:ELCOPY,ELTASK/R
*BIN:ELINIT,MAP:ELINIT/W=OBJ:ELINIT
*BIN:ERROUT,MAP:ERROUT/W=OBJ:ERROUT,ERRTXT
$EOJ
```

C.9 FILEX

```
$JOB/RT11/TIME   ! AFILEX.BAT
        TTYIO
$MES ASSEMBLE FILEX
.R MACRO
*OBJ:FILEX,LST:FILEX/L:MEB/C=SRC:FILEX
$EOJ

$JOB/RT11/TIME   ! LFILEX.BAT
        TTYIO
$MES LINK FILEX
.R LINK
*BIN:FILEX,MAP:FILEX/W=OBJ:FILEX
$EOJ
```

C.10 FORMAT

```
$JOB/RT11/TIME   ! AFORMAT.BAT
        TTYIO
$MES ASSEMBLE FORMAT
.R MACRO
*OBJ:FORMAT,LST:FORMAT/C=SRC:FORMAT
*OBJ:FMTDEV,LST:FMTDEV/C=SRC:FMTDEV
*OBJ:FMTDM,LST:FMTDM/C=SRC:FMTDM
*OBJ:FMTRK,LST:FMTRK/C=SRC:FMTRK
*OBJ:FMTDY,LST:FMTDY/C=SRC:FMTDY
*OBJ:FMTDP,LST:FMTDP/C=SRC:FMTDP
*OBJ:FMTDL,LST:FMTDL/C=SRC:FMTDL
*OBJ:FMTDX,LST:FMTDX/C=SRC:FMTDX
*OBJ:FMTDT,LST:FMTDT/C=SRC:FMTDT
*OBJ:FMTDD,LST:FMTDD/C=SRC:FMTDD
$EOJ

$JOB/RT11/TIME   ! LFORMAT.BAT
        TTYIO
$MES LINK FORMAT
.R LINK
*BIN:FORMAT,MAP:FORMAT/W=OBJ:FORMAT,FMTDEV//
*OBJ:FMTDM/O:1
*OBJ:FMTRK/O:1
*OBJ:FMTDY/O:1
*OBJ:FMTDP/O:1
*OBJ:FMTDL/O:1
*OBJ:FMTDX/O:1
*OBJ:FMTDT/O:1
*OBJ:FMTDD/O:1
*//
$EOJ
```

C.11 HELP

```
$JOB/RT11/TIME   ! AHELP.BAT
        TTYIO
$MES ASSEMBLE HELP
.R MACRO
*OBJ:HELP,LST:HELP/C=SRC:HELP
$EOJ

$JOB/RT11/TIME   ! LHELP.BAT
        TTYIO
$MES LINK HELP
.R LINK
*BIN:HELP.EXE,MAP:HELP/W=OBJ:HELP
$EOJ

$JOB/RT11/TIME   ! BHELP.BAT
        TTYIO
$MES BUILD HELP.TXT LIBRARY
.R LIBR
*OBJ:HELP.TXT=SRC:HELP.MLB/M
$EOJ
```

C.12 LIBR

```
$JOB/RT11/TIME   ! ALIBR.BAT
        TTYIO
$MES ASSEMBLE LIBR
.R MACRO
*OBJ:LIBR0,LST:LIBR0/C=SRC:LIBR0
*OBJ:LIBR1,LST:LIBR1/C=SRC:LIBR1
*OBJ:LIBR2,LST:LIBR2/C=SRC:LIBR2
*OBJ:LIBR3,LST:LIBR3/C=SRC:LIBR3
```

```
*OBJ:LIBR4,LST:LIBR4/C=SRC:LIBR4
*OBJ:LIBR5,LST:LIBR5/C=SRC:LIBR5
*OBJ:LIBR6,LST:LIBR6/C=SRC:LIBR6
*OBJ:LBREM,LST:LBREM/C=SRC:LBREM
$EOJ

$JOB/RT11/TIME   ! LLIBR.BAT
        TTYIO
$MES LINK LIBR
.R LINK
*BIN:LIBR,MAP:LIBR/W=OBJ:LIBR0//
*OBJ:LIBR1/O:1
*OBJ:LIBR2/O:1
*OBJ:LIBR3/O:1
*OBJ:LIBR4/O:1
*OBJ:LIBR5/O:1
*OBJ:LIBR6/O:1
*OBJ:LBREM/O:1
*//
$EOJ
```

C.13 LINK

```
$JOB/RT11/TIME   ! ALINK.BAT
        TTY10
$MES ASSEMBLE LINK
.R MACRO
*OBJ:LINK0,LST:LINK0/C=SRC:LINK0
*OBJ:LINK1,LST:LINK1/C=SRC:LINK1
*OBJ:LINK2,LST:LINK2/C=SRC:LINK2
*OBJ:LINK3,LST:LINK3/C=SRC:LINK3
*OBJ:LINK4,LST:LINK4/C=SRC:LINK4
*OBJ:LINK5,LST:LINK5/C=SRC:LINK5
*OBJ:LINK6,LST:LINK6/C=SRC:LINK6
*OBJ:LINK7,LST:LINK7/C=SRC:LINK7
*OBJ:LNKEM,LST:LNKEM/C=SRC:LNKEM
$EOJ

$JOB/RT11/TIME   ! LLINK.BAT
        TTY10
$MES LINK LINK
.R LINK
*BIN:LINK,MAP:LINK/W=OBJ:LINK0//
*OBJ:LINK1/O:1
*OBJ:LINK2/O:1
*OBJ:LINK3/O:1
*OBJ:LINK4/O:1
*OBJ:LINK5/O:1
*OBJ:LINK6/O:1
*OBJ:LINK7/O:1
*OBJ:LNKEM/O:1
*//
$EOJ
```

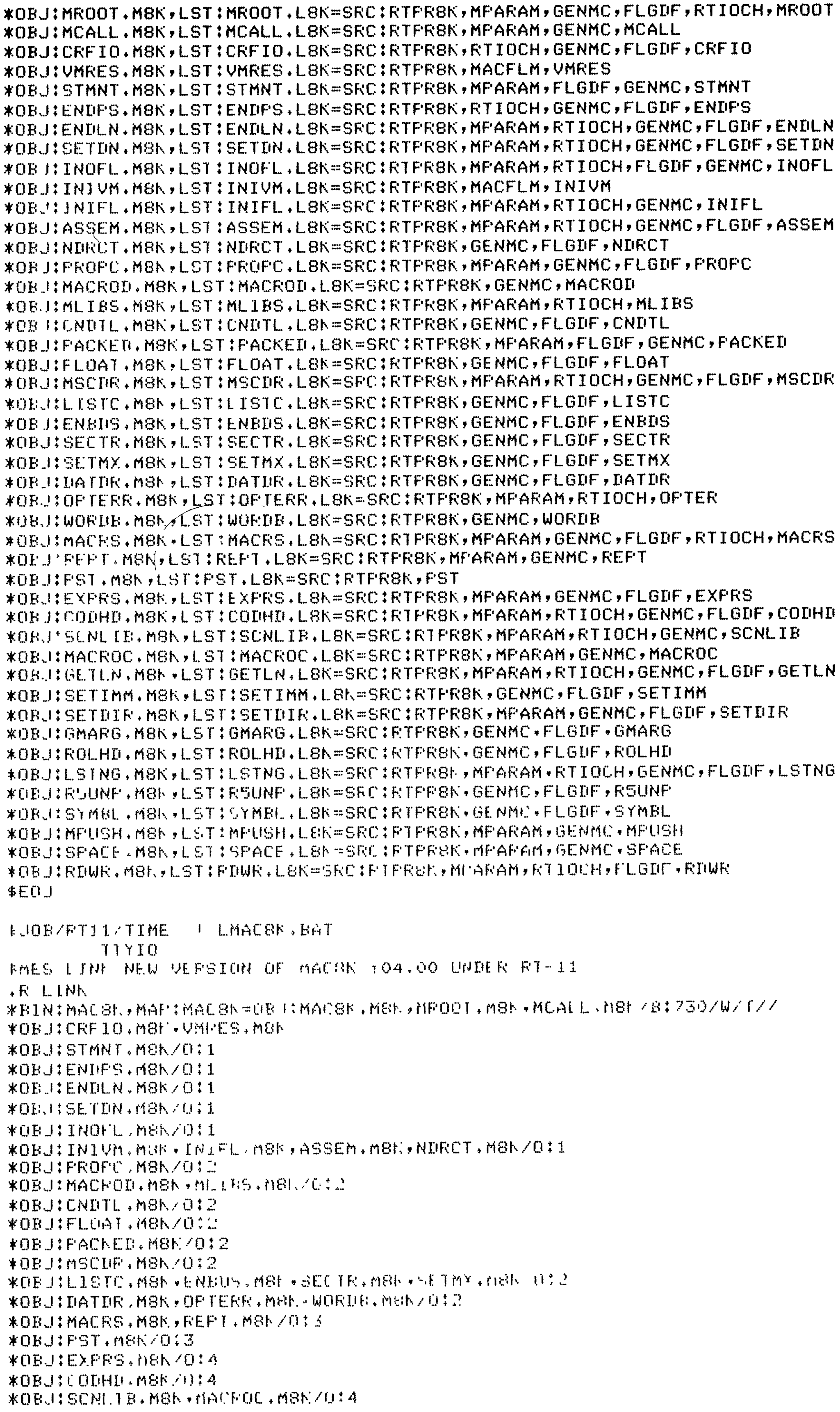
C.14 MACRO and MAC8K

```
$JOB/RT11/TIME   ! AMACRO.BAT
        TTYIO
$MES ASSEMBLE NEW VERSION OF MACRO-11 Y04.00 UNDER RT-11
.R MACRO
*OBJ:MACRO,LST:MACRO/C=SRC:RTPAR3,MACRO
*OBJ:MROOT,LST:MROOT/C=SRC:RTPAR3,MPARAM,GENMC,FLGDF,RTIOCH,MROOT
*OBJ:MCALL,LST:MCALL/C=SRC:RTPAR3,MPARAM,GENMC,MCALL
*OBJ:CRFIO,LST:CRFIO/C=SRC:RTPAR3,RTIOCH,GENMC,FLGDF,CRFIO
*OBJ:VMRES,LST:VMRES/C=SRC:RTPAR3,MACFLM,VMRES
*OBJ:STMNT,LST:STMNT/C=SRC:RTPAR3,MPARAM,FLGDF,GENMC,STMNT
*OBJ:ENDPS,LST:ENDPS/C=SRC:RTPAR3,RTIOCH,GENMC,FLGDF,ENDPS
*OBJ:ENDLN,LST:ENDLN/C=SRC:RTPAR3,MPARAM,RTIOCH,GENMC,FLGDF,ENDLN
```

```
*OBJ:SETDN,LST:SETDN/C=SRC:RTPAR3,MPARAM,RTIOCH,GENMC,FLGDF,SETDN
*OBJ:INOFL,LST:INOFL/C=SRC:RTPAR3,MPARAM,RTIOCH,FLGDF,GENMC,INOFL
*OBJ:INIVM,LST:INIVM/C=SRC:RTPAR3,MACFLM,INIVM
*OBJ:INIFL,LST:INIFL/C=SRC:RTPAR3,MPARAM,RTIOCH,GENMC,INIFL
*OBJ:ASSEM,LST:ASSEM/C=SRC:RTPAR3,MPARAM,RTIOCH,GENMC,FLGDF,ASSEM
*OBJ:NDRCT,LST:NDRCT/C=SRC:RTPAR3,GENMC,FLGDF,NDRCT
*OBJ:PROPC,LST:PROPC/C=SRC:RTPAR3,MPARAM,GENMC,FLGDF,PROPC
*OBJ:MACROD.LST:MACROD/C=SRC:RTPAR3,GENMC,MACROD
*OBJ:MLIBS,LST:MLIBS/C=SRC:RTPAR3,MPARAM,RTIOCH,MLIBS
*OBJ:CNDTL,LST:CNDTL/C=SRC:RTPAR3,GENMC,FLGDF,CNDTL
*OBJ:PACKED,LST:PACKED/C=SRC:RTPAR3,MPARAM,GENMC,FLGDF,PACKED
*OBJ:FLOAT.LST:FLOAT/C=SRC:RTPAR3,GENMC,FLGDF,FLOAT
*OBJ:MSCDR,LST:MSCDR/C=SRC:RTPAR3,MPARAM,RTIOCH.GENMC,FLGDF,MSCDR
*OBJ:LISTC,LST:LISTC/C=SRC:RTPAR3,GENMC,FLGDF,LISTC
*OBJ:ENBDS,LST:ENBDS/C=SRC:RTPAR3,GENMC,FLGDF,ENBDS
*OBJ:SECTR,LST:SECTR/C=SRC:RTPAR3.GENMC,FLGDF,SECTR
*OBJ:SETMX,LST:SETMX/C=SRC:RTPAR3,GENMC,FLGDF,SETMX
*OBJ:DATDR,LST:DATDR/C=SRC:RTPAR3,GENMC,FLGDF,DATDR
*OBJ:OPTERR,LST:OPTERR/C=SRC:RTPAR3,MPARAM,RTIOCH,OPTER
*OBJ:WORDB,LST:WORDB/C=SRC:RTPAR3,GENMC,WORDB
*OBJ:MACRS,LST:MACRS/C=SRC:RTPAR3,MPARAM,GENMC,FLGDF,RTIOCH,MACRS
*OBJ:REPT,LST:REPT/C=SRC:RTPAR3,MPARAM,GENMC,REPT
*OBJ:PST,LST:PST/C=SRC:RTPAR3,PST
*OBJ:EXPRS,LST:EXPRS/C=SRC:RTPAR3,MPARAM,GENMC,FLGDF,EXPRS
*OBJ:CODHD,LST:CODHD/C=SRC:RTPAR3,MPARAM,RTIOCH,GENMC,FLGDF,CODHD
*OBJ:SCNLIB,LST:SCNLIB/C=SRC:RTPAR3,MPARAM,RTIOCH,GENMC,SCNLIB
*OBJ:MACROC,LST:MACROC/C=SRC:RTPAR3,MPARAM,GENMC,MACROC
*OBJ:GETLN,LST:GETLN/C=SRC:RTPAR3,MPARAM,RTIOCH,GENMC,FLGDF,GETLN
*OBJ:SETIMM,LST:SETIMM/C=SRC:RTPAR3.GENMC,FLGDF,SETIMM
*OBJ:SETDIR,LST:SETDIR/C=SRC:RTPAR3,MPARAM,GENMC,FLGDF,SETDIR
*OBJ:GMARG.LST:GMARG/C=SRC:RTPAR3,GENMC,FLGDF,GMARG
*OBJ:ROLHD,LST:ROLHD/C=SRC:RTPAR3,GENMC,FLGDF,ROLHD
*OBJ:LSTNG,LST:LSTNG/C=SRC:RTPAR3.MPARAM,RTIOCH,GENMC,FLGDF,LSTNG
*OBJ:R5UNP.LST:R5UNP/C=SRC:RTPAR3,GENMC,FLGDF,R5UNP
*OBJ:SYMBL,LST:SYMBL/C=SRC:RTPAR3,GENMC.FLGDF,SYMBL
*OBJ:MPUSH,LST:MPUSH/C=SRC:RTPAR3,MPARAM,GENMC,MPUSH
*OBJ:SPACE,LST:SPACE/C=SRC:RTPAR3,MPARAM,GENMC,SPACE
*OBJ:RDWR,LST:RDWR/C=SRC:RTPAR3,MPARAM,RTIOCH,FLGDF,RDWR
$EOJ

$JOB/RT11/TIME   ! LMACRO.BAT
        TTYIO
$MES LINK NEW VERSION OF MACRO-11 Y04.00 UNDER RT-11
.R LINK
*BIN:MACRO.MAP:MACRO/W=OBJ:MACRO//
*OBJ:MROOT
*OBJ:SPACE,SETMX,MPUSH,MACRS,CRFIO
*OBJ:R5UNP,VMRES,MACPOC
*OBJ:SCNLIB
*OBJ:MCALL,CNDTL,MSCDR
*OBJ:REPT,WORDB,SECTR
*OBJ:DATDR,CODHD
*OBJ:EXPRS,SETIMM,SETDIR,GMARG
*OBJ:ROLHD,SYMBL
*OBJ:INIFL,INOFL/O:1
*OBJ:ASSEM,SETDN/O:1
*OBJ:INIVM,ENDPS/O:1
*OBJ:MACROD,STMNT,ENDLN,NDRCT/O:1
*OBJ:PST/O:2
*OBJ:FLOAT/O:2
*OBJ:PACKED/O:2
*OBJ:GETLN,RDWR,OPTERR,PROPC/O:2
*OBJ:MLIBS/O:3
*OBJ:LSTNG,ENBDS,LISTC/O:3
*//
$EOJ

$JOB/RT11/TIME   ! AMAC8K.BAT
        TTYIO
$MES ASSEMBLE NEW VERSION OF MAC8K Y04.00 UNDER RT-11
.R MACRO
*OBJ:MAC8K.M8K,LST:MAC8K.L8K=SRC:RTPR8K,MAC8K
```



```
*OBJ:MROOT.M8K,LST:MROOT.L8K=SRC:RTPR8K,MPARAM,GENMC,FLGDF,RTIOCH,MROOT
*OBJ:MCALL.M8K,LST:MCALL.L8K=SRC:RTPR8K,MPARAM,GENMC,MCALL
*OBJ:CRFIO.M8K,LST:CRFIO.L8K=SRC:RTPR8K,RTIOCH,GENMC,FLGDF,CRFIO
*OBJ:VMRES.M8K,LST:VMRES.L8K=SRC:RTPR8K,MACFLM,VMRES
*OBJ:STMNT.M8K,LST:STMNT.L8K=SRC:RTPR8K,MPARAM,FLGDF,GENMC,STMNT
*OBJ:ENDPS.M8K,LST:ENDPS.L8K=SRC:RTPR8K,RTIOCH,GENMC,FLGDF,ENDPS
*OBJ:ENDLN.M8K,LST:ENDLN.L8K=SRC:RTPR8K,MPARAM,RTIOCH,GENMC,FLGDF,ENDLN
*OBJ:SETDN.M8K,LST:SETDN.L8K=SRC:RTPR8K,MPARAM,RTIOCH,GENMC,FLGDF,SETDN
*OBJ:INOFL.M8K,LST:INOFL.L8K=SRC:RTPR8K,MPARAM,RTIOCH,FLGDF,GENMC,INOFL
*OBJ:INIVM.M8K,LST:INIVM.L8K=SRC:RTPR8K,MACFLM,INIVM
*OBJ:INIFL.M8K,LST:INIFL.L8K=SRC:RTPR8K,MPARAM,RTIOCH,GENMC,INIFL
*OBJ:ASSEM.M8K,LST:ASSEM.L8K=SRC:RTPR8K,MPARAM,RTIOCH,GENMC,FLGDF,ASSEM
*OBJ:NDRCT.M8K,LST:NDRCT.L8K=SRC:RTPR8K,GENMC,FLGDF,NDRCT
*OBJ:PROPC.M8K,LST:PROPC.L8K=SRC:RTPR8K,MPARAM,GENMC,FLGDF,PROPC
*OBJ:MACROD.M8K,LST:MACROD.L8K=SRC:RTPR8K,GENMC,MACROD
*OBJ:MLIBS.M8K,LST:MLIBS.L8K=SRC:RTPR8K,MPARAM,RTIOCH,MLIBS
*OBJ:CNDTL.M8K,LST:CNDTL.L8K=SRC:RTPR8K,GENMC,FLGDF,CNDTL
*OBJ:PACKED.M8K,LST:PACKED.L8K=SRC:RTPR8K,MPARAM,FLGDF,GENMC,PACKED
*OBJ:FLOAT.M8K,LST:FLOAT.L8K=SRC:RTPR8K,GENMC,FLGDF,FLOAT
*OBJ:MSCDR.M8K,LST:MSCDR.L8K=SRC:RTPR8K,MPARAM,RTIOCH,GENMC,FLGDF,MSCDR
*OBJ:LISTC.M8K,LST:LISTC.L8K=SRC:RTPR8K,GENMC,FLGDF,LISTC
*OBJ:ENBDS.M8K,LST:ENBDS.L8K=SRC:RTPR8K,GENMC,FLGDF,ENBDS
*OBJ:SECTR.M8K,LST:SECTR.L8K=SRC:RTPR8K,GENMC,FLGDF,SECTR
*OBJ:SETMX.M8K,LST:SETMX.L8K=SRC:RTPR8K,GENMC,FLGDF,SETMX
*OBJ:DATDR.M8K,LST:DATDR.L8K=SRC:RTPR8K,GENMC,FLGDF,DATDR
*OBJ:OPTERR.M8K,LST:OPTERR.L8K=SRC:RTPR8K,MPARAM,RTIOCH,OPTER
*OBJ:WORDB.M8K,LST:WORDB.L8K=SRC:RTPR8K,GENMC,WORDB
*OBJ:MACRS.M8K,LST:MACRS.L8K=SRC:RTPR8K,MPARAM,GENMC,FLGDF,RTIOCH,MACRS
*OBJ:REPT.M8K,LST:REPT.L8K=SRC:RTPR8K,MPARAM,GENMC,REPT
*OBJ:PST.M8K,LST:PST.L8K=SRC:RTPR8K,PST
*OBJ:EXPRS.M8K,LST:EXPRS.L8K=SRC:RTPR8K,MPARAM,GENMC,FLGDF,EXPRS
*OBJ:CODHD.M8K,LST:CODHD.L8K=SRC:RTPR8K,MPARAM,RTIOCH,GENMC,FLGDF,CODHD
*OBJ:SCNLIB.M8K,LST:SCNLIB.L8K=SRC:RTPR8K,MPARAM,RTIOCH,GENMC,SCNLIB
*OBJ:MACROC.M8K,LST:MACROC.L8K=SRC:RTPR8K,MPARAM,GENMC,MACROC
*OBJ:GETLN.M8K,LST:GETLN.L8K=SRC:RTPR8K,MPARAM,RTIOCH,GENMC,FLGDF,GETLN
*OBJ:SETIMM.M8K,LST:SETIMM.L8K=SRC:RTPR8K,GENMC,FLGDF,SETIMM
*OBJ:SETDIR.M8K,LST:SETDIR.L8K=SRC:RTPR8K,MPARAM,GENMC,FLGDF,SETDIR
*OBJ:GMARG.M8K,LST:GMARG.L8K=SRC:RTPR8K,GENMC,FLGDF,GMARG
*OBJ:ROLHD.M8K,LST:ROLHD.L8K=SRC:RTPR8K,GENMC,FLGDF,ROLHD
*OBJ:LSTNG.M8K,LST:LSTNG.L8K=SRC:RTPR8K,MPARAM,RTIOCH,GENMC,FLGDF,LSTNG
*OBJ:R5UNP.M8K,LST:R5UNP.L8K=SRC:RTPR8K,GENMC,FLGDF,R5UNP
*OBJ:SYMBL.M8K,LST:SYMBL.L8K=SRC:RTPR8K,GENMC,FLGDF,SYMBL
*OBJ:MPUSH.M8K,LST:MPUSH.L8K=SRC:RTPR8K,MPARAM,GENMC,MPUSH
*OBJ:SPACE.M8K,LST:SPACE.L8K=SRC:RTPR8K,MPARAM,GENMC,SPACE
*OBJ:RDWR.M8K,LST:RDWR.L8K=SRC:RTPR8K,MPARAM,RTIOCH,FLGDF,RDWR
$EOJ

$JOB/RT11/TIME    ! LMAC8K.BAT
        T1YIO
$MES LINK NEW VERSION OF MAC8K V04.00 UNDER RT-11
.R LINK
*BIN:MAC8K,MAP:MAC8K=OBJ:MAC8K.M8K,MROOT.M8K,MCALL.M8K/B:730/W/T//
*OBJ:CRFIO.M8K,VMRES.M8K
*OBJ:STMNT.M8K/O:1
*OBJ:ENDPS.M8K/O:1
*OBJ:ENDLN.M8K/O:1
*OBJ:SETDN.M8K/O:1
*OBJ:INOFL.M8K/O:1
*OBJ:INIVM.M8K,INIFL.M8K,ASSEM.M8K,NDRCT.M8K/O:1
*OBJ:PROPC.M8K/O:2
*OBJ:MACROD.M8K,MLIBS.M8K/O:2
*OBJ:CNDTL.M8K/O:2
*OBJ:FLOAT.M8K/O:2
*OBJ:PACKED.M8K/O:2
*OBJ:MSCDR.M8K/O:2
*OBJ:LISTC.M8K,ENBDS.M8K,SECTR.M8K,SETMX.M8K/O:2
*OBJ:DATDR.M8K,OPTERR.M8K,WORDB.M8K/O:2
*OBJ:MACRS.M8K,REPT.M8K/O:3
*OBJ:PST.M8K/O:3
*OBJ:EXPRS.M8K/O:4
*OBJ:CODHD.M8K/O:4
*OBJ:SCNLIB.M8K,MACROC.M8K/O:4
```

```
*OBJ:GETLN.M8K/O:5
*OBJ:SETIMM.M8K,SETDIR.LOK,GMARG.M8K/O:5
*OBJ:ROLHD.M8K/O:5
*OBJ:LSTNG.M8K,RSLNK.M8K/O:5
*OBJ:SYMBL.M8K/O:6
*OBJ:MPUSH.M8K,SPACE.M8K/O:6
*OBJ:RDWR.M8K/O:6
*//
*START
$EOJ
```

C.15 MDUP, MBOOT, and MSBOOT

```
$JOB/RT11/TIME   ! AMDUP.BAT
        TTYIO
$MES ASSEMBLE MDUP
.R MACRO
*OBJ:MDPROT.LST:MDPROT/C=SRC:DUPPRE,MDUP,DUPROT
*OBJ:MDPIN1,LST:MDPIN1/C=SRC:DUPPRE,MDUP,DUPIN1
*OBJ:MDPIN2,LST:MDPIN2/C=SRC:DUPPRE,MDUP,DUPIN2
*OBJ:MDPCRE,LST:MDPCRE/C=SRC:DUPPRE,MDUP,DUPCRE
*OBJ:MDPSCN,LST:MDPSCN/C=SRC:DUPPRE,MDUP,DUPSCN,DUPMRG
*OBJ:MDPBOT,LST:MDPBOT/C=SRC:DUPPRE,MDUP,DUPBOT
*OBJ:MDPWBT.LST:MDPWBT/C=SRC:DUPPRE,MDUP,DUPWBT
*OBJ:MDPZRO,LST:MDPZRO/C=SRC:DUPPRE,MDUP,DUPZRO
$EOJ
```

```
$JOB/RT11/TIME   ! LMDUP.BAT
        TTYIO
$MES LINK MDUP
.R LINK
*BIN:MDUP,MAP:MDUP/W=OBJ:MDPROT//
*OBJ:MDPIN1
*OBJ:MDPIN2
*OBJ:MDPCRE
*OBJ:MDPSCN
*OBJ:MDPBOT
*OBJ:MDPWBT
*OBJ:MDPZRO
*BIN:ULBLIB
*//
$EOJ
```

This procedure builds the program MDUP.SAV. MDUP.SAV builds the magtape build programs, MDUP.MM for TJU16 magtape, MDUP.MS for TS11 magtape, and MDUP.MT for TM11 magtape. The magtape build program runs from the booted magtape, initializes the system disk, and copies a minimal system from the magtape to the disk.

The Version 4 MDUP.MM, MDUP.MS, and MDUP.MT programs support the following devices:

RK05	RL01	RP02	RS03	RF11
RK06	RL02	RP03	RS04	
RK07				

If you need to create MDUP for a disk or magtape not supported by RT–11 (for which you have written your own handler), use MDUP.SAV as follows.

Use the hardware version of the magtape handler and the standard (distributed) single-job monitor. Your configuration must include at least 28K words of memory. Apply the following patch to the monitor. In the patch, xx is the name of the device on which the monitor resides.

```
.R SIPP<RET>
*xxn:RT11SJ.SYS<RET>
Base?      0<RET>
Offset?    1030<RET>

     Base          Offset        Old           New?
     000000        001030        000407        240<RET>
     000000        001032        013704        12704<RET>
     000000        001034        177570        60000<RET>
     000000        001036        042704        <CTRL/Y><RET>
*<CTRL/C>
.
```

Then copy the bootstrap:

```
.COPY/BOOT xxn:RT11SJ.SYS xxn:<RET>
.
```

This procedure causes the system to use only 12K words of memory when booted.

Next apply the following patch to every handler to be supported by the version of MDUP you are building. In the patch, xx is the name of the device on which the handler resides, and yy is the name of the handler.

```
.R SIPP<RET>
*xxn:yy.SYS<RET>
Base?      0<RET>
Offset?    176<RET>

     Base          Offset        Old           New?
     000000        000176        ??????        0<RET>
     000000        000200        ??????        0<RET>
     000000        000202        ??????        <CTRL/Y><RET>
*<CTRL/C>
.
```

Now boot the monitor you have patched. Set the USR NOSWAP. Load all the handlers to be supported. Run MDUP.SAV and issue the following command. In the command, xx is the physical device name and filnam.typ is your version of MDUP (MDUP.??).

```
.R MDUP.SAV<RET>
*xx:filnam.typ=/H<RET>
```

MDUP.SAV creates the file filnam.typ and exits.

If you make a typing error in the command line, use CTRL/C to abort MDUP.SAV and run MDUP.SAV again. If you get an insufficient memory error message, you cannot build MDUP with all the handlers loaded. Unload one of the handlers, and run MDUP.SAV again.

```
$JOB/RT11/TIME   ! AMBOOT.BAT
        TTYIO
$MES ASSEMBLE MBOOT
.R MACRO
*OBJ:MBOOT,LST:MBOOT/C=SRC:MBOOT
$EOJ

$JOB/RT11/TIME   ! LMBOOT.BAT
        TTYIO
$MES LINK MBOOT
.R LINK
*BIN:MBOOT.BOT,MAP:MBOOT/W=OBJ:MBOOT
$EOJ

$JOB/RT11/TIME   ! AMSBOO.BAT
        TTYIO
$MES ASSEMBLE MSBOOT
.R MACRO
*OBJ:MSBOOT,LST:MSBOOT/C=SRC:MSBOOT
$EOJ

$JOB/RT11/TIME   ! LMSBOO.BAT
        TTYIO
$MES LINK MSBOOT
.R LINK
*BIN:MSBOOT.BOT,MAP:MSBOOT/W=OBJ:MSBOOT
$EOJ
```

C.16 ODT

NOTE

When you assemble ODT, an assembly error occurs (a Z error). Ignore this error.

```
$JOB/RT11/TIME   ! AODT.BAT
        TTYIO
$MES ASSEMBLE ODT
.R MACRO
*BIN:ODT,LST:ODT/C=SRC:ODT
*BIN:VDT,LST:VDT/C=SRC:VDT,O
$EOJ
```

C.17 PAT

```
$JOB/RT11/TIME   ! APAT.BAT
        TTYIO
$MES ASSEMBLE PAT
.R MACRO
*OBJ:PAT,LST:PAT/C=SRC:PAT
$EOJ

$JOB/RT11/TIME   ! LPAT.BAT
        TTYIO
$MES LINK PAT
.R LINK
*BIN:PAT,MAP:PAT/W=OBJ:PAT
$EOJ
```

C.18 PATCH

```
$JOB/RT11/TIME   ! APATCH.BAT
        TTYIO
$MES ASSEMBLE PATCH
.R MACRO
*OBJ:PATCH,LST:PATCH/L:MEB/C=SRC:PATCH
$EOJ

$JOB/RT11/TIME   ! LPATCH.BAT
        TTYIO
$MES LINK PATCH
.R LINK
*BIN:PATCH,MAP:PATCH/W=OBJ:PATCH
$EOJ
```

C.19 PIP

```
$JOB/RT11/TIME   ! APIP.BAT
        TTYIO
$MES ASSEMBLE PIP
.R MACRO
*OBJ:PIPROT,LST:PIPROT/C=SRC:PIPROT
*OBJ:PIPINI,LST:PIPINI/C=SRC:PIPPRE,PIPINI
*OBJ:PIPEXE,LST:PIPEXE/C=SRC:PIPPRE,PIPEXE
*OBJ:PIPDK,LST:PIPDK/C=SRC:PIPPRE,PIPDK
*OBJ:PIPMT,LST:PIPMT/C=SRC:PIPPRE,PIPMT
*OBJ:PIPCT,LST:PIPCT/C=SRC:PIPPRE,PIPCT
$EOJ

$JOB/RT11/TIME   ! LPIP.BAT
        TTYIO
$MES LINK PIP
.R LINK
*BIN:PIP,MAP:PIP/W=OBJ:PIPROT,BIN:ULBLIB//
*OBJ:PIPINI/O:1
*OBJ:PIPEXE/O:1
*OBJ:PIPDK/O:2
*OBJ:PIPMT/O:2
*OBJ:PIPCT/O:2
*//
$EOJ
```

C.20 QUEUE

```
$JOB/RT11/TIME   ! AQUEUE.BAT
        TTYIO
$MES ASSEMBLE QUEUE PACKAGE
.R MACRO
*OBJ:QUEMAN,LST:QUEMAN/C=SRC:ULBMAC,QUEMAN
*OBJ:QUEUE,LST:QUEUE/C=SRC:QUEUE
$EOJ

$JOB/RT11/TIME   ! LQUEUE.BAT
        TTYIO
$MES LINK QUEUE PACKAGE
.R LINK
*BIN:QUEMAN,MAP:QUEMAN/W=OBJ:QUEMAN,BIN:ULBLIB
*BIN:QUEUE/R,MAP:QUEUE/W=OBJ:QUEUE
$EOJ
```

C.21 RESORC

```
$JOB/RT11/TIME   ! ARESOR.BAT
        TTYIO
$MES ASSEMBLE RESORC
.R MACRO
*OBJ:RESORC,LST:RESORC/C=SRC:ULBMAC,RESORC
$EOJ

$JOB/RT11/TIME   ! LRESOR.BAT
        TTYIO
$MES LINK RESORC
.R LINK
*BIN:RESORC,MAP:RESORC/W=OBJ:RESORC,BIN:UL
$EOJ
```

C.22 SIPP

```
$JOB/RT11/TIME   ! ASIPP.BAT
        TTYIO
$MES ASSEMBLE SIPP
.R MACRO
*OBJ:SIPP,LST:SIPP/L:MEB/C=SRC:ULBMAC,SIPP
$EOJ

$JOB/RT11/TIME   ! LSIPP.BAT
        TTYIO
$MES LINK SIPP
.R LINK
*BIN:SIPP,MAP:SIPP/W=OBJ:SIPP,BIN:ULBLIB
$EOJ
```

C.23 SLP

```
$JOB/RT11/TIME   ! ASLP.BAT
        TTYIO
$MES ASSEMBLE SLP
.R MACRO
*OBJ:SLP,LST:SLP/C=SRC:SLPPRE,SLP
*OBJ:SLPERR,LST:SLPERR/C=SRC:SLPPRE,SLPERR
*OBJ:SLPIO,LST:SLPIO/C=SRC:SLPPRE,SLPIO
*OBJ:SLPSUB,LST:SLPSUB/C=SRC:SLPPRE,SLPSUB
*OBJ:POSIT,LST:POSIT/C=SRC:SLPPRE,TPMAC,POSIT
*OBJ:TPARS,LST:TPARS/C=SRC:TPARS
$EOJ

$JOB/RT11/TIME   ! LSLP.BAT
        TTYIO
$MES LINK SLP
.R LINK
*BIN:SLP,MAP:SLP/W=OBJ:SLP//
*OBJ:SLPERR
*OBJ:SLPIO
*OBJ:SLPSUB
*OBJ:POSIT
*OBJ:TPARS
*//
$EOJ
```

C.24 SRCCOM

```
$JOB/RT11/TIME   ! ASRCCO.BAT
        TTYIO
$MES ASSEMBLE SRCCOM
.R MACRO
*OBJ:SRCCOM,LST:SRCCOM/C=SRC:ULBMAC,SRCCOM
$EOJ

$JOB/RT11/TIME   ! LSRCCO.BAT
        TTYIO
$MES LINK SRCCOM
.R LINK
*BIN:SRCCOM,MAP:SRCCOM/W=OBJ:SRCCOM,BIN:ULBLIB
$EOJ
```

C.25 SWAP

```
$JOB/RT11/TIME   ! ASWAP.BAT
        TTYIO
$MES ASSEMBLE SWAP.SYS
.R MACRO
*OBJ:SWAP,LST:SWAP/C=SRC:SWAP
$EOJ

$JOB/RT11/TIME   ! LSWAP.BAT
        TTYIO
$MES LINK SWAP.SYS
.R LINK
*BIN:SWAP.SYS=OBJ:SWAP
$EOJ
```

C.26 SYSGEN

```
$JOB/RT11/TIME   ! CSYSGE.BAT
        TTYIO
$MES COMPILE SYSGEN
.R FORTRA
*OBJ:SYSGEN,LST:SYSGEN=SRC:SYSGEN/I:THR/E/B/W/N:6/V/O/S
$EOJ

$JOB/RT11/TIME   ! LSYSGE.BAT
        TTYIO
$MES LINK SYSGEN
.R LINK
*BIN:SYSGEN,MAP:SYSGEN/W=OBJ:SYSGEN,SY:FORLIB
$EOJ
```

C.27 SYSLIB

```
$JOB/RT11/TIME   ! SYSLIB.BAT
        TTYIO
$MES ASSEMBLE SYSLIB
.R MACRO
*OBJ:OHANDL,LST:OHANDL/C=SRC:OHANDL
*OBJ:VHANDL,LST:VHANDL/C=SRC:VHANDL
*OBJ:CHAIN,LST:CHAIN/C=SRC:CHAIN
*OBJ:CLOSEC,LST:CLOSEC/C=SRC:CLOSEC
*OBJ:CMPLT,LST:CMPLT/C=SRC:CMPLT
*OBJ:CONCAT,LST:CONCAT/C=SRC:CONCAT
*OBJ:CVTTIM,LST:CVTTIM/C=SRC:CVTTIM
*OBJ:DEVICE,LST:DEVICE/C=SRC:DEVICE
*OBJ:DIV60,LST:DIV60/C=SRC:DIV60
```

```
*OBJ:GTIM,LST:GTIM/C=SRC:GTIM
*OBJ:GTJB,LST:GTJB/C=SRC:GTJB
*OBJ:GTLIN,LST:GTLIN/C=SRC:GTLIN
*OBJ:IADDR,LST:IADDR/C=SRC:IADDR
*OBJ:IASIGN,LST:IASIGN/C=SRC:IASIGN
*OBJ:ICDFN,LST:ICDFN/C=SRC:ICDFN
*OBJ:ICHCPY,LST:ICHCPY/C=SRC:ICHCPY
*OBJ:ICMKT,LST:ICMKT/C=SRC:ICMKT
*OBJ:ICSI,LST:ICSI/C=SRC:ICSI
*OBJ:ICSTAT,LST:ICSTAT/C=SRC:ICSTAT
*OBJ:IDELET,LST:IDELET/C=SRC:IDELET
*OBJ:IDSTAT,LST:IDSTAT/C=SRC:IDSTAT
*OBJ:IENTER,LST:IENTER/C=SRC:IENTER
*OBJ:IFETCH,LST:IFETCH/C=SRC:IFETCH
*OBJ:IFREEC,LST:IFREEC/C=SRC:IFREEC
*OBJ:IGETC,LST:IGETC/C=SRC:IGETC
*OBJ:IGETSP,LST:IGETSP/C=SRC:IGETSP
*OBJ:IJCVT,LST:IJCVT/C=SRC:IJCVT
*OBJ:ILUN,LST:ILUN/C=SRC:ILUN
*OBJ:INDEX,LST:INDEX/C=SRC:INDEX
*OBJ:INSERT,LST:INSERT/C=SRC:INSERT
*OBJ:INTSET,LST:INTSET/C=SRC:INTSET
*OBJ:IPEEK,LST:IPEEK/C=SRC:IPEEK
*OBJ:IPEEKB,LST:IPEEKB/C=SRC:IPEEKB
*OBJ:IQSET,LST:IQSET/C=SRC:IQSET
*OBJ:IRAD50,LST:IRAD50/C=SRC:IRAD50
*OBJ:IRCVD,LST:IRCVD/C=SRC:IRCVD
*OBJ:IRCVDF,LST:IRCVDF/C=SRC:IRCVDF
*OBJ:IREAD,LST:IREAD/C=SRC:IREAD
*OBJ:IREADF,LST:IREADF/C=SRC:IREADF
*OBJ:IRENAM,LST:IRENAM/C=SRC:IRENAM
*OBJ:IREOPN,LST:IREOPN/C=SRC:IREOPN
*OBJ:ISAVES,LST:ISAVES/C=SRC:ISAVES
*OBJ:ISCHED,LST:ISCHED/C=SRC:ISCHED
*OBJ:ISDAT,LST:ISDAT/C=SRC:ISDAT
*OBJ:ISDATF,LST:ISDATF/C=SRC:ISDATF
*OBJ:ISLEEP,LST:ISLEEP/C=SRC:ISLEEP
*OBJ:ISPFN,LST:ISPFN/C=SRC:ISPFN
*OBJ:ISPFNF,LST:ISPFNF/C=SRC:ISPFNF
*OBJ:ISPY,LST:ISPY/C=SRC:ISPY
*OBJ:ITLOCK,LST:ITLOCK/C=SRC:ITLOCK
*OBJ:ITTINR,LST:ITTINR/C=SRC:ITTINR
*OBJ:ITTOUR,LST:ITTOUR/C=SRC:ITTOUR
*OBJ:ITWAIT,LST:ITWAIT/C=SRC:ITWAIT
*OBJ:IUNTIL,LST:IUNTIL/C=SRC:IUNTIL
*OBJ:IWAIT,LST:IWAIT/C=SRC:IWAIT
*OBJ:IWRITE,LST:IWRITE/C=SRC:IWRITE
*OBJ:IWRITF,LST:IWRITF/C=SRC:IWRITF
*OBJ:JADD,LST:JADD/C=SRC:JADD
*OBJ:JCMP,LST:JCMP/C=SRC:JCMP
*OBJ:JDIV,LST:JDIV/C=SRC:JDIV
*OBJ:JFIX,LST:JFIX/C=SRC:JFIX
*OBJ:JFLT,LST:JFLT/C=SRC:JFLT
*OBJ:JICVT,LST:JICVT/C=SRC:JICVT
*OBJ:JJCVT,LST:JJCVT/C=SRC:JJCVT
*OBJ:JMOV,LST:JMOV/C=SRC:JMOV
*OBJ:JMUL,LST:JMUL/C=SRC:JMUL
*OBJ:JSUB,LST:JSUB/C=SRC:JSUB
*OBJ:JTIME,LST:JTIME/C=SRC:JTIME
*OBJ:LEN,LST:LEN/C=SRC:LEN
*OBJ:LOCK,LST:LOCK/C=SRC:LOCK
*OBJ:LOOKUP,LST:LOOKUP/C=SRC:LOOKUP
*OBJ:MRKT,LST:MRKT/C=SRC:MRKT
*OBJ:MTGET,LST:MTGET/C=SRC:MTGET
*OBJ:MTSET,LST:MTSET/C=SRC:MTSET
*OBJ:MWAIT,LST:MWAIT/C=SRC:MWAIT
*OBJ:PRINT,LST:PRINT/C=SRC:PRINT
*OBJ:PURGE,LST:PURGE/C=SRC:PURGE
*OBJ:R50ASC,LST:R50ASC/C=SRC:R50ASC
*OBJ:RCHAIN,LST:RCHAIN/C=SRC:RCHAIN
*OBJ:RCTRLO,LST:RCTRLO/C=SRC:RCTRLO
*OBJ:REPEAT,LST:REPEAT/C=SRC:REPEAT
*OBJ:RESUME,LST:RESUME/C=SRC:RESUME
```

```
*OBJ:SCCA,LST:SCCA/C=SRC:SCCA
*OBJ:SCOMP,LST:SCOMP/C=SRC:SCOMP
*OBJ:SCOPY,LST:SCOPY/C=SRC:SCOPY
*OBJ:SECNDS,LST:SECNDS/C=SRC:SECNDS
*OBJ:SETCMD,LST:SETCMD/C=SRC:SETCMD
*OBJ:STRPAD,LST:STRPAD/C=SRC:STRPAD
*OBJ:SUBSTR,LST:SUBSTR/C=SRC:SUBSTR
*OBJ:SUSPND,LST:SUSPND/C=SRC:SUSPND
*OBJ:SYSLBV,LST:SYSLBV/C=SRC:SYSLBV
*OBJ:TIMASC,LST:TIMASC/C=SRC:TIMASC
*OBJ:TIME,LST:TIME/C=SRC:TIME
*OBJ:TIME1,LST:TIME1/C=SRC:TIME1
*OBJ:TIMSUB,LST:TIMSUB/C=SRC:TIMSUB
*OBJ:TRANSL,LST:TRANSL/C=SRC:TRANSL
*OBJ:TRIM,LST:TRIM/C=SRC:TRIM
*OBJ:UNLOCK,LST:UNLOCK/C=SRC:UNLOCK
*OBJ:VERIFY,LST:VERIFY/C=SRC:VERIFY
$EOJ

$JOB/RT11/TIME   ! CSYSLI.BAT
        TTYIO
$MES    COMPILE SYSLIB FORTRAN PARTS
.R FORTRA
*OBJ:PUTSTR,LST:PUTSTR=SRC:PUTSTR/I:THR/S
*OBJ:GETSTR,LST:GETSTR=SRC:GETSTR/I:THR/S
$EOJ

$JOB/RT11/TIME   ! BSYSL1.BAT
        TTYIO
$MES BUILD SYSLIB
.R LIBR
*BIN:SYSLIB,MAP:SYSLIB/W=OBJ:VHANDL,OHANDL/G//
*OBJ:CHAIN,CLOSEC,CMPLT,CONCAT,CVTTIM
*OBJ:DEVICE,DIV60,GTIM,GTJB,GTLIN
*OBJ:IADDR,IASIGN,ICDFN,ICHCPY,ICMKT
*OBJ:ICSI,ICSTAT,IDELET,IDSTAT,IENTER
*OBJ:IFETCH,IFREEC,IGETC,IGETSP,IJCVT
*OBJ:ILUN,INDEX,INSERT,INTSET,IPEEK
*OBJ:IPEEKB,IQSET,IRAD50,IRCVD,IRCVDF
*OBJ:IREAD,IREADF,IRENAM,IREOPN,ISAVES
*OBJ:ISCHED,ISDAT,ISDATF,ISLEEP,ISPFN
*OBJ:ISPFNF,ISPY,ITLOCK,ITTINR,ITTOUR
*OBJ:ITWAIT,IUNTIL,IWAIT,IWRITE,IWRITF
*OBJ:JADD,JCMP,JDIV,JFIX,JFLT
*OBJ:JICVT,JJCVT,JMOV,JMUL,JSUB
*OBJ:JTIME,LEN,LOCK,LOOKUP,MRKT
*OBJ:MTGET,MTSET,MWAIT,PRINT,PURGE,R50ASC
*OBJ:RCHAIN,RCTRLO,REPEAT,RESUME,SCCA
*OBJ:SCOMP,SCOPY,SECNDS,SETCMD,STRPAD
*OBJ:SUBSTR,SUSPND,SYSLBV,TIMASC,TIME
*OBJ:TIME1,TIMSUB,TRANSL,TRIM,UNLOCK
*OBJ:VERIFY,PUTSTR,GETSTR
*//
*$OVRH
*
$EOJ
```

C.28 SYSMAC

```
$JOB/RT11/TIME   ! BSYSMA.BAT
        TTYIO
$MES BUILD SYSMAC.SML
.R LIBR
*BIN:SYSMAC.SML=SRC:SYSMAC.MAC/M
$EOJ
```

C.29 TECO

```
$JOB/RT11/TIME   ! ATECO.BAT
        TTYIO
$MES ASSEMBLE TECO
.R MACRO
*OBJ:TECOIO,LST:TECOIO/C=SRC:TIOPRE,TECOIO
*OBJ:TIOFET,LST:TIOFET/C=SRC:TIOPRE,TIOFET
*OBJ:TIOEIO,LST:TIOEIO/C=SRC:TIOPRE,TIOEIO
*OBJ:TIOIAS,LST:TIOIAS/C=SRC:TIOPRE,TIOIAS
*OBJ:CRTRUB,LST:CRTRUB/C=SRC:CRTPRE,CRTRUB
*OBJ:TIOENC,LST:TIOENC/C=SRC:TIOPRE,TIOENC
*OBJ:TIOINI,LST:TIOINI/C=SRC:TIOPRE,TIOINI
*OBJ:TIORFS,LST:TIORFS/C=SRC:TIOPRE,TIORFS
*OBJ:TIODCD,LST:TIODCD/C=SRC:TIOPRE,TIODCD
$EOJ
```

```
$JOB/RT11/TIME   ! LTECO.BAT
        TTYIO
$MES LINK TECO
.R LINK
*BIN:TECO,MAP:TECO/W=SRC:TECO,OBJ:TECOIO,TIOFET//
*OBJ:TIOEIO/O:1
*OBJ:TIOIAS/O:1,SRC:SCREEN,SCROLL,OBJ:CRTRUB
*OBJ:TIOENC/O:1
*OBJ:TIOINI/O:1
*OBJ:TIORFS/O:1
*OBJ:TIODCD/O:2
*//
*BIN:TECO/R:400,MAP:TECOFG/W=SRC:TECO,OBJ:TECOIO/E:20000//
*OBJ:TIOEIO/O:1
*OBJ:TIOIAS/O:1,SRC:SCREEN,OBJ:CRTRUB
*OBJ:TIOENC/O:1
*OBJ:TIOINI/O:1
*OBJ:TIORFS/O:1
*OBJ:TIODCD/O:2
*//
*MORBUF
$EOJ
```

C.30 Utilities Library

```
$JOB/RT11/TIME   ! AULBLI.BAT
        TTYIO
$MES ASSEMBLE THE UTILITIES LIBRARY
.R MACRO
*OBJ:ULB001,LST:ULB001/C=SRC:ULB001
*OBJ:ULB002,LST:ULB002/C=SRC:ULB002
*OBJ:ULB003,LST:ULB003/C=SRC:ULB003
*OBJ:ULB004,LST:ULB004/C=SRC:ULB004
*OBJ:ULB005,LST:ULB005/C=SRC:ULB005
*OBJ:ULB006,LST:ULB006/C=SRC:ULB006
*OBJ:ULB007,LST:ULB007/C=SRC:ULB007
*OBJ:ULB008,LST:ULB008/C=SRC:ULB008
*OBJ:ULB009,LST:ULB009/C=SRC:ULB009
*OBJ:ULB010,LST:ULB010/C=SRC:ULB010
*OBJ:ULB011,LST:ULB011/C=SRC:ULB011
*OBJ:ULB012,LST:ULB012/C=SRC:ULB012
*OBJ:ULB013,LST:ULB013/C=SRC:ULBMAC,ULB013
*OBJ:ULB014,LST:ULB014/C=SRC:ULB014
*OBJ:ULB015,LST:ULB015/C=SRC:ULB015
*OBJ:ULB016,LST:ULB016/C=SRC:ULB016
*OBJ:ULB017,LST:ULB017/C=SRC:ULB017
*OBJ:ULB018,LST:ULB018/C=SRC:ULB018
*OBJ:ULB019,LST:ULB019/C=SRC:ULB019
*OBJ:ULB020,LST:ULB020/C=SRC:ULB020
*OBJ:ULB021,LST:ULB021/C=SRC:ULB021
*OBJ:ULB022,LST:ULB022/C=SRC:ULB022
```

```
*OBJ:ULB023,LST:ULB023/C=SRC:ULB023
*OBJ:ULB024,LST:ULB024/C=SRC:ULB024
*OBJ:ULB025,LST:ULB025/C=SRC:ULB025
*OBJ:ULB026,LST:ULB026/C=SRC:ULB026
*OBJ:ULB027,LST:ULB027/C=SRC:ULB027
*OBJ:ULB028,LST:ULB028/C=SRC:ULBMAC,ULB028
*OBJ:ULB029,LST:ULB029/C=SRC:ULB029
*OBJ:ULB030,LST:ULB030/C=SRC:ULBMAC,ULB030
*OBJ:ULB031,LST:ULB031/C=SRC:ULBMAC,ULB031
*OBJ:ULB032,LST:ULB032/C=SRC:ULBMAC,ULB032
*OBJ:ULB033,LST:ULB033/C=SRC:ULBMAC,ULB033
*OBJ:ULB034,LST:ULB034/C=SRC:ULB034
*OBJ:ULB035,LST:ULB035/C=SRC:ULB035
*OBJ:ULB036,LST:ULB036/C=SRC:ULB036
*OBJ:ULB037,LST:ULB037/C=SRC:ULB037
*OBJ:ULB038,LST:ULB038/C=SRC:ULB038
*OBJ:ULB039,LST:ULB039/C=SRC:ULB039
*OBJ:ULB040,LST:ULB040/C=SRC:ULB040
*OBJ:ULB041,LST:ULB041/C=SRC:ULB041
*OBJ:ULB042,LST:ULB042/C=SRC:ULB042
*OBJ:ULB043,LST:ULB043/C=SRC:ULB043
*OBJ:ULB044,LST:ULB044/C=SRC:ULB044
$EOJ

$JOB/RT11/TIME   ! BULBLI.BAT
        TTYIO
$MES BUILD THE UTILITIES LIBRARY
.R LIBR
*BIN:ULBLIB/A,MAP:ULBLIB/W=//
*OBJ:ULB001,ULB002,ULB003,ULB004,ULB005,ULB006
*OBJ:ULB007,ULB008,ULB009,ULB010,ULB011,ULB012
*OBJ:ULB013,ULB014,ULB015,ULB016,ULB017,ULB018
*OBJ:ULB019,ULB020,ULB021,ULB022,ULB023,ULB024
*OBJ:ULB025,ULB026,ULB027,ULB028,ULB029,ULB030
*OBJ:ULB031,ULB032,ULB033,ULB034,ULB035,ULB036


*OBJ:ULB037,ULB038,ULB039,ULB040,ULB041,ULB042
*OBJ:ULB043,ULB044
*//
$EOJ
```

C.31 VT11 Library and PLOT55

```
$JOB/RT11/TIME   ! AVTLIB.BAT
        TTYIO
$MES ASSEMBLE VTLIB AND PLOT55
.R MACRO
*OBJ:VTCAL1,LST:VTCAL1/C=SRC:VTCAL1
*OBJ:VTCAL2,LST:VTCAL2/C=SRC:VTCAL2
*OBJ:VTCAL3,LST:VTCAL3/C=SRC:VTCAL3
*OBJ:VTCAL4,LST:VTCAL4/C=SRC:VTCAL4
*OBJ:VTBASE,LST:VTBASE/C=SRC:VTBASE
*BIN:PLOT55,LST:PLOT55/C=SRC:PLOT55
$EOJ

$JOB/RT11/TIME   ! BVTLIB.BAT
        TTYIO
$MES BUILD VTLIB
.R LIBR
*BIN:VTHDLR,MAP:VTHDLR=OBJ:VTCAL1.OBJ,VTCAL2.OBJ,VTCAL3.OBJ,VTCAL4.OBJ/C
*OBJ:VTBASE.OBJ
$EOJ
```

Appendix D
SYSGEN Script Language

The program SYSGEN is a FORTRAN program that uses a script file containing commands written in the SYSGEN script language. The Script file drives the terminal dialogue and sends output information to command and conditional files. Although it is possible to alter the SYSGEN script file for your own application generations, DIGITAL supports system generation only when you use the standard script file distributed with the system.

The script file must be named SYSGEN.CND and it must reside on device DK:. It is an ASCII input file that consists of SYSGEN command lines interspersed with the actual text lines that SYSGEN.SAV outputs to the output files.

Directed by commands in the script file, the SYSGEN program interacts with you at the terminal. SYSGEN.SAV manipulates internal variables (script variables), processes text lines included in the script, and outputs these text lines to the specified output file. Normally, the text lines processed in a script file are either assembly language statements for a conditional assembly file or command lines for an indirect command file. However, they can be any kind of text line for any purpose.

Each line in a script file must be less than or equal to 80. characters. If a line begins with a semicolon character, SYSGEN ignores it, so you can use semicolons to indicate comment lines in the script file.

SYSGEN interprets any line beginning with TAB# as a command line. However, if you use the #ABBR command to set abbreviation mode, SYSGEN interprets any line beginning with a # as a command. Refer to the description of #ABBR (Section D.1) before you attempt to use abbreviation mode.

Terminate the command name with a second TAB or, if there are no arguments, RETURN and LINE FEED. If the command does not match a legal command name, SYSGEN reports an error and treats the line as a text line.

SYSGEN sends any other lines to the currently open output file. Do not begin an output line with a # if you intend to use abbreviation mode (see Section D.1). Before SYSGEN processes any line, including command lines, it first makes any substitutions in effect (see #SUBS command). If you need to output a line that begins with a semicolon, precede the semicolon with a space or tab to differentiate it from a script comment.

The following examples illustrate these syntax rules. Example a results in the current output file containing the lines in example b after COMMAND1 and COMMAND2 execute:

Example a

```
;THIS IS A SCRIPT EXAMPLE
TAB#command1TABarg1
TAB;CONDITIONAL FILE COMMENT
TAB#command2TABarg1
COND1=1         ;CONDITIONAL
COND2=2         ;CONDITIONAL
```

Example b

```
TAB;CONDITIONAL FILE COMMENT
COND1=1         ;CONDITIONAL
COND2=2         ;CONDITIONAL
```

The script commands, summarized in the following list, are described in Sections D.1 through D.16.

#ABBR
#ASK
#CALL
#DECR
#ENDC
#ENDS
#EXIT
#FILE
#IF, #IFN, #IFGT
#IFT, #IFF, #IFTF
#INCR
#NAME
#PRINT
#ROPRT
#SET
#SUBS

Script commands manipulate script variables, which have names of from one to six alphanumeric characters. To establish the value of a script variable, initially undefined, use a #SET command or the variable in an #ASK command. Each script variable has two values associated with it: a character value and a numeric value. The character value is equal to the actual characters used in a #SET or #ASK command. The numeric value is equal to the numeric value of the characters with which the variable was associated during an #ASK command. #IF, #IFN, and #SUBS commands use the character value. #IFGT, #INCR, and #DECR commands use the numeric value. #ASK, #INCR, and #DECR commands set both the character and numeric values of a variable, while #SET alters only the character value. The number of different variables for each execution of the SYSGEN program is limited to 100. Once you define script variables, you can redefine them with subsequent #SET or #ASK commands, but redefinition within a substitute block does not take effect until the next #SUBS statement.

SYSGEN detects errors in the script file. Table D–1 lists and explains the errors. SYSGEN script file errors print in the format:

error message in segment name
script line

In the message, segment name is the name assigned by the most recent #NAME command, and script line is an image of the script input line in which the error was detected.

D.1 #ABBR Command

The #ABBR command sets abbreviation mode, which allows you to use an abbreviated form for commands so that you can omit the leading TAB from commands. If abbreviation mode is not set, SYSGEN interprets as a command only a line with a leading TAB# sequence.

normal mode command format:

TAB#command

If you issue an #ABBR command to set abbreviation mode, SYSGEN interprets as commands those lines that begin with pound signs (#).

abbreviation mode command format:

#command

or

TAB#command

SYSGEN uses the abbreviation mode format until you issue another #ABBR command, at which point SYSGEN reverts to using only normal mode command format.

NOTE

> If you do use the #ABBR command to eliminate the need for leading TABs, any line that begins with the pound sign character (#) is interpreted as a command. You cannot send lines with leading TABs to the output file, because SYSGEN would interpret those lines as commands.

D.2 #ASK Command

The #ASK command prints dialogue on the terminal and obtains an acceptable user response. #ASK then sets the specified script variable's character value and numeric value according to the user's response.

#ASK prints on the terminal the question indicated between the TAB and the question mark character (?) and waits for the user to respond.

The #ASK command always resets CTRL/O. If the user running SYSGEN typed CTRL/O to inhibit the previous text, the reset allows the next string (the question) to print on the terminal. Refer to Section D.14 for more information about CTRL/O.

When you use #ASK, the text must contain the question, question mark, variable name, slash character, type letter (A, D, O, or Y), equal sign, and default value, as shown in the following format.

Command Format:

$$\text{TAB\#ASKTAB[-]question?variable/}\left\{\begin{array}{c}\text{A}\\\text{D}\\\text{O}\\\text{Y}\end{array}\right\}\text{=default[,lowlimit[,highlimit]]}$$

SYSGEN sets a variable's character value to the characters in the user response and sets the variables's numeric value to the value of the numeric response, as appropriate. It then proceeds to the next command.

SYSGEN checks the terminal responses for validity according to the type letter code (A, D, O, or Y) you specified in the #ASK command. The type letter codes establish validity checking for the following data types:

/A The answer must be an alphanumeric device mnemonic of the form xxRET or xxnRET, where x is a letter and n is a digit in the range 0–7.

/D The answer must be a decimal number.

/O The answer must be an octal number.

/Y The answer must be either yesRET or noRET.

Optionally, you can use a hyphen as a continuation character. When a hyphen is the first character following TAB#ASKTAB, SYSGEN assumes the question portion of the command to be the second line of a two-line question. Use #PRINT or # to display the first line of the question. If you use the #PRINT/#ASK combination instead of hyphen for two lines, you may need to substitute #ROPRT for #PRINT. #ROPRT resets CTRL/O as well as prints, ensuring that the first line prints even after a CTRL/O has been used to inhibit previous text.

The low- and high-limit values are optional and apply to only numeric (/D or /O) responses. These limits establish a range for acceptable responses. SYSGEN checks numeric responses against the low and high limits, and prints an error message if the response is not valid (see Table 9–1). SYSGEN then repeats the question.

If the response to a question is RETURN, SYSGEN uses the default response. (Note that SYSGEN does not check the default response for validity.)

Examples:

```
TAB#ASK HOW OLD ARE YOU? AGE/D=21,18,65

TAB#PRINTTABTHIS IS THE FIRST LINE
TAB#ASKTAB-OF A TWO LINE QUESTION? ANS/Y=N
```

The PRINT/ASK combination in this example prints in the dialogue as follows:

```
THIS IS THE FIRST LINE
OF A TWO LINE QUESTION?
```

D.3 #CALL Command

The #CALL command causes SYSGEN to use the file you specify (filnam.typ) as the temporary source for script input. When the system encounters the end-of-file for this temporary input file, it resumes input from the previous file (SYSGEN.CND).

You cannot nest #CALL commands. Only one #CALL command can be active at a time.

You must explicitly enter the file type, since there is no default file type.

Command Format:

TAB#CALLTABfilnam.typ

Example:

```
TAB#CALLTABSYSTBL.CND
```

D.4 #DECR Command

The #DECR command decrements the value associated with the decimal variable you specify. When you use the #SET or #ASK command, SYSGEN sets the value of the variable equal to the numeric value of the user response to a dialogue question. #DECR alters both the numeric value and the answer (a decimal number).

You can use #DECR with #IFGT to repeat a command or a group of commands n times, where n is the value of a response to a dialogue question. For example, SYSGEN.CND uses this combination to ask for n DL11 vectors, where n is the number of DL11s specified by the user in response to the dialogue question.

Command Format:

TAB#DECRTABvariable name

D.5 #ENDC Command

The #ENDC command terminates the current #IF, #IFN, or #IFGT conditional block. The number of #ENDC commands must match the number of #IF, #IFN, and #IFGT commands.

Optionally, you can follow the #ENDC command with a script variable name to mark which block you terminate in nesting situations.

Command Format:

(TAB)#ENDC(TAB)[variable name]

Example:

```
(TAB)#IF(TAB)COND1
     .
     .
     .
(TAB)#ENDC(TAB)COND1
(TAB)#IFN(TAB)COND2
     .
     .
     .
(TAB)#IF(TAB)COND3
     .
     .
     .
(TAB)#IFF
     .
     .
     .
(TAB)#IFT
     .
     .
     .
(TAB)#ENDC(TAB)COND3
(TAB)#ENDC(TAB)COND2
```

D.6 #ENDS Command

The #ENDS command terminates the substitution activities of all active #SUBS commands. You need use only one #ENDS command, regardless of the number of #SUBS statements that are active.

Command Format:

(TAB)#ENDS

D.7 #EXIT Command

The #EXIT command terminates the SYSGEN run without closing the active output file.

Command Format:

(TAB)#EXIT

D.8 #FILE Command

The #FILE command opens the file you name as the current output file. SYSGEN writes all text lines in the output stream to this file until you issue another #FILE command. There is no default file type; you must explicitly enter the file type. SYSGEN closes all open output files before it opens the file you specify.

Command format:

(TAB)#FILE(TAB)filnam.typ

Example:

```
(TAB)#FILE(TAB)SYCND.MAC
(TAB)#FILE(TAB)SY1.COM
```

D.9 #IF, #IFN, #IFGT Commands

The #IFx commands test the character value or the numeric value of a script variable. If the condition is satisfied, SYSGEN includes the commands between the #IFx command and the matching #ENDC command in the script file processing. If the condition is not met, SYSGEN ignores lines between this command and the #ENDC command.

#IF tests the character value of a script variable for yes. The #IFN command tests the character value of a script variable for no. The #IFGT command tests the numeric value of a script variable. If the numeric value is greater than 0, the condition is true. If the value is less than or equal to 0, the condition is false. You can use #IFGT with #DECR to repeat a question n times, where n is the value of a response to a dialogue question.

You can nest #IF, #IFN, and #IFGT commands up to 10 levels, but you must pair each #IF, #IFN, or #IFGT with a corresponding #ENDC command.

Command format:

(TAB)#IF(TAB)variable
(TAB)#IFN(TAB)variable
(TAB)#IFGT(TAB)variable

Example:

```
TAB#SETTABA=Y
TAB#SETTABB=N
A IS YES
TAB#ENDC
TAB#IFNTABA
A IS NO
TAB#ENDC
TAB#IFTABB
B IS YES
TAB#ENDC
TAB#IFNTABB
B IS NO
TAB#ENDC
```

This example writes the following lines in the output file:

```
A IS YES
B IS NO
```

D.10 #IFT, #IFF, #IFTF Commands

You can use the IFT, IFF, and IFTF commands within an existing #IF, #IFN, or #IFGT conditional block to:

1. Include an alternative body of text and command lines when the condition of the block tests false.
2. Include a noncontiguous body of text and command lines within the block when the condition of the block tests true.
3. Include a body of text and command lines unconditionally within a conditional block.

#IFT, #IFF, and #IFTF do not require a matching #ENDC; they are part of the #IF, #IFN, or #IFGT block to which they belong.

#IFT includes the subsequent lines if the conditional that heads the current conditional block is true.

#IFF includes the subsequent lines if the conditional that heads the current conditional block is false.

#IFTF includes the subsequent lines in either case.

Command format:

```
TAB#IFT
TAB#IFF
TAB#IFTF
```

Thus, you could change the example in Section D.9 as follows:

```
TAB#SETTABA=Y
TAB#SETTABB=N
TAB#IFTABA
A IS YES
TAB#IFF
A IS NO
TAB#ENDC
TAB#IFNTABB
TAB#IFF
B IS YES
TAB#IFT
B IS NO
TAB#ENDC
```

D.11 #INCR Command

The #INCR command increments the value associated with the decimal variable you specify. When you use the #SET or #ASK command, SYSGEN sets the numeric value of the variable equal to the numeric value of the user response to a dialogue question. #INCR alters both the numeric value and the answer (a decimal number).

Command Format:

TAB#INCRTABvariable name

D.12 #NAME Command

The #NAME command identifies a particular segment of the script file. The segment name prints as part of the error message when SYSGEN encounters an error condition. You can use this command to logically divide the script file into discrete sections. The segment name consists of from one to six alphanumeric characters.

Command format:

TAB#NAMETABsegnam

Example:

```
TAB#NAMETABSYCND
TAB#NAMETABPART 1
```

D.13 #PRINT Command

Use the #PRINT command to print the text you specify on the terminal. When SYSGEN prints the text, it precedes it with a LINE FEED and follows it with a RETURN.

Command Format:

TAB#PRINTTABtext

Example:

```
(TAB)#PRINT(TAB)MESSAGE1
(TAB)#PRINT
(TAB)#PRINT(TAB)MESSAGE2
```

This example prints the following lines on the terminal:

```
MESSAGE1

MESSAGE2
```

SYSGEN also offers an alternative method of printing text on the terminal. You can use the pound sign (#) alone instead of #PRINT:

Command Format:

(TAB)#(TAB)text

Example:

```
(TAB)#(TAB)MESSAGE1
(TAB)#
(TAB)#(TAB)MESSAGE2
```

This example prints the following lines on the terminal:

```
MESSAGE1

MESSAGE2
```

D.14 #ROPRT Command

The #ROPRT command resets CTRL/O and prints text on the terminal. If a user running SYSGEN types CTRL/O to inhibit text (that is, keep text from printing on the terminal), CTRL/O must be reset to enable text to print again.

#ROPRT is commonly used for two-line questions. You can use #ROPRT to print the first line of a question (even if the user running SYSGEN has typed CTRL/O to inhibit previous text). Then in combination with the #ROPRT command, you can use #ASK to print the second line of the question and obtain a response.

Command Format:

(TAB)#ROPRT(TAB)MESSAGE

Example:

```
(TAB)#ROPRT(TAB)RESET CTRL/O TO PRINT THIS LINE.  IS
(TAB)#ASK(TAB)THIS A QUESTION? ANS/Y=Y
```

D.15 #SET Command

The #SET command makes the character value of the variable you specify equal to the characters in the answer.

Command Format:

TAB#SETTABvariable name = answer

Examples:

```
TAB#SETTABA=1
TAB#SETTABCOND1=N
TAB#SETTABCOND2=Y
```

D.16 #SUBS Command

The #SUBS command substitutes the character value of the script variable you specify for each occurrence of the pattern in all script lines between the #SUBS command and the next #ENDS command.

SYSGEN prepares the text substitution at the time it interprets the SUBS statement. Therefore, the value of the variable within a SUBS block does not change until SYSGEN encounters the next SUBS statement that uses that variable.

Command Format:

TAB#SUBSTAB<delimiter>pattern<delimiter>variable name

Example:

```
TAB#SETTABCOND1=3
TAB#SUBSTAB$XXX$COND1
TAB#PRINTTABTHE VALUE OF COND1 IS $XXX$
TABCOND1=$XXX$
TAB#ENDS
```

This example prints the following line on the terminal:

```
THE VALUE OF COND1 IS 3
```

The following line goes to the output file:

```
COND1=3
```

Table D–1: SYSGEN Script File Errors

Message	Meaning
?SCRIPT-W-Bad #ASK in	You left off the ? delimiter in an #ASK command.
?SCRIPT-W-Bad #SET in	You left off the = delimiter in a #SET command.
?SCRIPT-W-Bad substitute pattern in	The delimiters in a #SUBS command do not match.
?SCRIPT-W-Missing variable in	You issued an #IF, #IFN, or #IFGT command without a script variable.
?SCRIPT-W-Missing file name in	You issued a #FILE command without a file name.
?SCRIPT-W-Nested #CALL in	You issued a #CALL command from a script file other than the original.
?SCRIPT-W-Too few #ENDCs in	The number of #IF, #IFN, and #IFGT statements exceeds the number of #ENDC commands.
?SCRIPT-W-Too many variables in	You defined more than 100 script variables.
?SCRIPT-W-Too many #ENDCs in	The number of #ENDC commands exceeds the number of #IF, #IFN, and #IFGT commands.
?SCRIPT-W-Too many nested #IFs or #SUBSs in	The number of nested #IF, #IFN, and #IFGT commands exceeds 10. The number of active #SUBS commands exceeds 5.
?SCRIPT-W-Undefined command in	You began the script line with TAB# but did not define a command or set abbreviation mode.
?SCRIPT-W-Undefined variable in	You did not define a variable name you used in an #IF, #IFN, #IFGT, or #SUBS command.

Appendix E
Loading Software Bootstraps

If your hardware configuration does not include a hardware bootstrap, use the following instructions for performing a software bootstrap. Find the section for bootstrapping your device, and use the switch register to deposit the bootstrap in memory. If your hardware configuration includes a push-button console emulator instead of a switch register, follow the instructions in your hardware manual to load the appropriate bootstrap loader (listed in Tables E–1 through E–8).

E.1 Loading the RK11 (RK05) DECpack Bootstrap

Deposit the basic RK11 disk bootstrap loader in memory as follows:

1. Set the ENABLE/HALT switch to HALT.
2. Set the first address, 001000, in the switch register (see Table E–1).
3. Press the LOAD ADDR switch.
4. Set the contents for the first address (from Table E–1) in the switch register.
5. Lift the DEP switch. The computer automatically advances to the next address.
6. Set the contents for the next address (from Table E–1) in the switch register.
7. Lift the DEP switch.
8. Repeat steps 6 and 7 until you have deposited all the instructions.

Now verify that you deposited the bootstrap program properly.

1. Set the first address, 001000, in the switch register.
2. Press the LOAD ADDR switch.
3. Press the EXAM switch to display the contents of that address in the data register.
4. Compare the number in the data register with the value for that address in Table E–1.
5. If the values are the same, press EXAM again to display the contents of the next address. If the values are not the same, repeat the entire procedure for depositing the bootstrap. Verify the contents of all the addresses in this way. If any instruction is incorrect, repeat the entire deposit procedure.

Once you have correctly deposited the bootstrap in memory, start the computer as follows:

1. Set the starting address, 001000, in the switch register.
2. Press the LOAD ADDR switch.
3. Set the ENABLE/HALT switch to ENABLE.
4. Press the START switch.

E.2 Loading the TC11 DECtape Bootstrap

Deposit the basic DECtape bootstrap loader in memory as follows:

1. Set the ENABLE/HALT switch to HALT.
2. Set the first address, 001000, in the switch register (see Table E–2).
3. Press the LOAD ADDR switch.
4. Set the contents for the first address (from Table E–2) in the switch register.
5. Lift the DEP switch. The computer automatically advances to the next address.
6. Set the contents for the next address (from Table E–2) in the switch register.
7. Lift the DEP switch.
8. Repeat steps 6 and 7 until you have deposited all the instructions.

Now verify that you deposited the bootstrap program properly.

1. Set the first address, 001000, in the switch register.
2. Press the LOAD ADDR switch.

3. Press the EXAM switch to display the contents of that address in the data register.
4. Compare the number in the data register with the value for that address in Table E–2.
5. If the values are the same, press EXAM again to display the contents of the next address. If the values are not the same, repeat the entire procedure for depositing the bootstrap. Verify the contents of all the addresses in this way. If any instruction is incorrect, repeat the entire deposit procedure.

Once you have correctly deposited the bootstrap in memory, start the computer as follows:

1. Set the starting address, 001000, in the switch register.
2. Press the LOAD ADDR switch.
3. Set the ENABLE/HALT switch to ENABLE.
4. Press the START switch.

E.3 Loading the RX11 Bootstrap

The procedure to deposit the RX11 disk bootstrap loader in memory depends on the type of processor you have. If your computer is a PDP–11V/03, PDP–11/03, or LSI–11, see the PDP–11/03 user's manual for instructions.

1. Set the ENABLE/HALT switch to HALT.
2. Set the first address, 010000, in the switch register (see Table E–3).
3. Press the LOAD ADDR switch.
4. Set the contents for the first address (from Table E–3) in the switch register.
5. Lift the DEP switch. The computer automatically advances to the next address.
6. Set the contents for the next address (from Table E–3) in the switch register.
7. Lift the DEP switch.
8. Repeat steps 6 and 7 until you have deposited all the instructions.

Now verify that you deposited the bootstrap program properly.

1. Set the first address, 001000, in the switch register.
2. Press the LOAD ADDR switch.
3. Press the EXAM switch to display the contents of that address in the data register.

4. Compare the number in the data register with the value for that address in Table E–3.

5. If the values are the same, press EXAM again to display the contents of the next address. If the values are not the same, repeat the entire procedure for depositing the bootstrap. Verify the contents of all the addresses in this way. If any instruction is incorrect, repeat the entire deposit procedure.

Once you have correctly deposited the bootstrap in memory, start the computer as follows:

1. Set the starting address, 001000, in the switch register.
2. Press the LOAD ADDR switch.
3. Set the ENABLE/HALT switch to ENABLE.
4. Press the START switch.

E.4 Loading the Magtape Bootstrap

Deposit the basic magtape bootstrap loader in memory as follows:

1. Set the ENABLE/HALT switch to HALT.
2. Set the first address, 010000, in the switch register (see Table E–4 or E–5).
3. Press the LOAD ADDR switch.
4. Set the contents for the first address (from Table E–4 for TJU16 magtape or Table E–5 for TM11 magtape) in the switch register.
5. Lift the DEP switch. The computer automatically advances to the next address.
6. Set the contents for the next address (from Table E–4 or E–5) in the switch register.
7. Lift the DEP switch.
8. Repeat steps 6 and 7 until you have deposited all the instructions.

Now verify that you deposited the bootstrap program properly.

1. Set the first address, 010000, in the switch register.
2. Press the LOAD ADDR switch.
3. Press the EXAM switch to display the contents of that address in the data register.
4. Compare the number in the data register with the value for that address in Table E–4 or E–5.

5. If the values are the same, press EXAM again to display the contents of the next address. If the values are not the same, repeat the entire procedure for depositing the bootstrap. Verify the contents of all the addresses in this way. If any instruction is incorrect, repeat the entire deposit procedure.

Once you have correctly deposited the bootstrap in memory, start the computer as follows:

1. If the magtape is not positioned at the load point, rewind the magtape manually.
2. Set the starting address, 010000, in the switch register.
3. Press the LOAD ADDR switch.
4. Set the ENABLE/HALT switch to ENABLE.
5. Press the START switch.

E.5 Loading the RK06 DECpack Bootstrap

Deposit the basic RK06 disk bootstrap loader in memory as follows:

1. Set the ENABLE/HALT switch to HALT.
2. Set the first address, 001000, in the switch register (see Table E–6).
3. Press the LOAD ADDR switch.
4. Set the contents for the first address (from Table E–6) in the switch register.
5. Lift the DEP switch. The computer automatically advances to the next address.
6. Set the contents for the next address (from Table E–6) in the switch register.
7. Lift the DEP switch.
8. Repeat steps 6 and 7 until you have deposited all the instructions.

Now verify that you deposited the bootstrap program properly.

1. Set the first address, 001000, in the switch register.
2. Press the LOAD ADDR switch.
3. Press the EXAM switch to display the contents of that address in the data register.
4. Compare the number in the data register with the value for that address in Table E–6.

5. If the values are the same, press EXAM again to display the contents of the next address. If the values are not the same, repeat the entire procedure for depositing the bootstrap. Verify the contents of all the addresses in this way. If any instruction is incorrect, repeat the entire deposit procedure.

Once you have correctly deposited the bootstrap in memory, start the computer as follows:

1. Set the starting address, 001000, in the switch register.
2. Press the LOAD ADDR switch.
3. Set the ENABLE/HALT switch to ENABLE.
4. Press the START switch.

E.6 Loading the RL01/RL02 Disk Bootstrap

Deposit the basic RL01/RL02 disk bootstrap loader in memory as follows:

1. Set the ENABLE/HALT switch to HALT.
2. Set the first address, 001000, in the switch register (see Table E–7).
3. Press the LOAD ADDR switch.
4. Set the contents for the first address (from Table E–7) in the switch register.
5. Lift the DEP switch. The computer automatically advances to the next address.
6. Set the contents for the next address (from Table E–7) in the switch register.
7. Lift the DEP switch.
8. Repeat steps 6 and 7 until you have deposited all the instructions.

Now verify that you deposited the bootstrap program properly.

1. Set the first address, 001000, in the switch register.
2. Press the LOAD ADDR switch.
3. Press the EXAM switch to display the contents of that address in the data register.
4. Compare the number in the data register with the value for that address in Table E–7.
5. If the values are the same, press EXAM again to display the contents of the next address. If the values are not the same, repeat the entire

procedure for depositing the bootstrap. Verify the contents of all the addresses in this way. If any instruction is incorrect, repeat the entire deposit procedure.

Once you have correctly deposited the bootstrap in memory, start the computer as follows:

1. Set the starting address, 001000, in the switch register.
2. Press the LOAD ADDR switch.
3. Set the ENABLE/HALT switch to ENABLE.
4. Press the START switch.

E.7 Loading the RX211 Bootstrap

Deposit the basic RX211 bootstrap loader in memory as follows:

1. Set the ENABLE/HALT switch to HALT.
2. Set the first address, 002000, in the switch register (see Table E–8).
3. Press the LOAD ADDR switch.
4. Set the contents for the first address (from Table E–8) in the switch register.
5. Lift the DEP switch. The computer automatically advances to the next address.
6. Set the contents for the next address (from Table E–8) in the switch register.
7. Lift the DEP switch.
8. Repeat steps 6 and 7 until you have deposited all the instructions.

Now verify that you deposited the bootstrap program properly.

1. Set the first address, 002000, in the switch register.
2. Press the LOAD ADDR switch.
3. Press the EXAM switch to display the contents of that address in the data register.
4. Compare the number in the data register with the value for that address in Table E–8.
5. If the values are the same, press EXAM again to display the contents of the next address. If the values are not the same, repeat the entire procedure for depositing the bootstrap. Verify the contents of all the addresses in this way. If any instruction is incorrect, repeat the entire deposit procedure.

Once you have correctly deposited the bootstrap in memory, start the computer as follows:

1. Set the starting address, 002000, in the switch register.
2. Press the LOAD ADDR switch.
3. Set the ENABLE/HALT switch to ENABLE.
4. Press the START switch.

Table E–1: RK11 Bootstrap Loader

Location	Contents	Location	Contents
001000	012700	001012	000005
001002	177406	001014	105710
001004	012710	001016	100376
001006	177400	001020	005007
001010	012740		

Table E–2: TC11 Bootstrap Loader

Location	Contents	Location	Contents
001000	012700	001022	000003
001002	177344	001024	105710
001004	012710	001026	100376
001006	177400	001030	012710
001010	012740	001032	000005
001012	004002	001034	105710
001014	005710	001036	100376
001016	100376	001040	005007
001020	012710		

Table E–3: RX11 Bootstrap Loader

Location	Contents	Location	Contents
001000	005000	001022	100405
001002	012701	001024	105711
001004	177170	001026	100004
001006	105711	001030	116120
001010	001776	001032	000002
001012	012711	001034	000770
001014	000003	001036	000000
001016	005711	001040	005000
001020	001776	001042	000110

Table E–4: TJU16 Bootstrap Loader

Location	Contents	Location	Contents
001000	012700	001040	177000
001002	172440	001042	012740
001004	012710	001044	000071
001006	000021	001046	032710
001010	012760	001050	100200
001012	001300	001052	001775
001014	000032	001054	100007
001016	012760	001056	022760
001020	177777	001060	001000
001022	000006	001062	000014
001024	012720	001064	001403
001026	000031	001066	000005
001030	105760	001070	000167
001032	000010	001072	177704
001034	100375	001074	005007
001036	012710		

Table E–5: TM11 Bootstrap Loader

Location	Contents	Location	Contents
010000	012700	010020	100767
010002	172524	010022	012710
010004	005310	010024	060003
010006	012740	010026	105710
010010	060011	010030	100376
010012	105710	010032	005710
010014	100376	010034	100777
010016	005710	010036	005007

Table E–6: RK06 Bootstrap Loader

Location	Contents	Location	Contents
001000	012701	001020	177400
001002	177440	001022	000002
001004	012711	001024	012711
001006	000003	001026	000021
001010	032711	001030	032711
001012	100200	001032	100200
001014	001775	001034	001775
001016	012761	001036	005007

Table E–7: RL01/RL02 Bootstrap Loader

Location	Contents	Location	Contents
001000	012701	001026	005061
001002	174400	001030	000004
001004	012761	001032	012761
001006	000013	001034	177400
001010	000004	001036	000006
001012	012711	001040	012711
001014	000004	001042	000014
001016	105711	001044	105711
001020	100376	001046	100376
001022	005061	001050	005007
001024	000002		

Table E–8: RX211 Bootstrap Loader

Location	Contents	Location	Contents
002000	012701	002060	000304
002002	177170	002062	030011
002004	012700	002064	001776
002006	100240	002066	100422
002010	005002	002070	012711
002012	012705	002072	000403
002014	000200	002074	030011
002016	012704	002076	001776
002020	000401	002100	100415
002022	012703	002102	010513
002024	177172	002104	030011
002026	030011	002106	001776
002030	001776	002110	100411
002032	100440	002112	010213
002034	012711	002114	060502
002036	000407	002116	060502
002040	030011	002120	122424
002042	001776	002122	120427
002044	100433	002124	000007
002046	110413	002126	003737
002050	000304	002130	005000
002052	030011	002132	005007
002054	001776	002134	000000
002056	110413		

Appendix F
RT–11 Conditionals

The following listing contains the conditionals that the RT–11 system uses with a brief explanation of each conditional. In this listing, the symbols have the following meanings:

symbol	meaning
nnnnnn	CSR address
nnn	vector address
n	number
x	number of local DL interfaces
y	number of remote DL11–E lines

F.1 System Conditionals

```
BATC$H    = 1
                    ;BATCH SUPPORT

CLOCK     = 50.     ;POWER LINE FREQUENCY (50 CYCLE)
CLOCK     = 60.     ;POWER LINE FREQUENCY (60 CYCLE)

DD$CSR    = nnnnnn  ;STATUS REGISTER FOR FIRST DECTAPE II CONTROLLER
DD$VEC    = nnn     ;VECTOR FOR FIRST DECTAPE II CONTROLLER
DDT$O     = 1       ;SECOND DECTAPE II CONTROLLER IS PRESENT
DD$CS2    = nnnnnn  ;SECOND DECTAPE II CONTROLLER STATUS REGISTER
DD$VC2    = nnn     ;VECTOR FOR SECOND DECTAPE II CONTROLLER
DL$UN     = n       ;NUMBER OF RL01 UNITS
DL11$L    = x       ;NUMBER OF LOCAL DL11 INTERFACES
DL11$M    = y       ;NUMBER OF REMOTE DL11–E LINES
DL11$N    = x + y   ;TOTAL NUMBER OF DL11 LINES
DLC$0     = nnnnnn  ;CSR OF FIRST DL11 (CONSOLE)
DLV$0     = nnn     ;VECTOR OF FIRST DL11 (CONSOLE)
DLC$1     = nnnnnn  ;CSR OF SECOND DL11 (LOCAL OR REMOTE)
DLV$1     = nnn     ;VECTOR OF SECOND DL11 (LOCAL OR REMOTE)
DLC$2     = nnnnnn  ;CSR OF THIRD DL11 (LOCAL OR REMOTE)
DLV$2     = nnn     ;VECTOR OF THIRD DL11 (LOCAL OR REMOTE)
```

```
DLC$3     = nnnnnn  ;CSR OF FOURTH DL11 (LOCAL OR REMOTE)
DLV$3     = nnn     ;VECTOR OF FOURTH DL11 (LOCAL OR REMOTE)
DLC$4     = nnnnnn  ;CSR OF FIFTH DL11 (LOCAL OR REMOTE)
DLV$4     = nnn     ;VECTOR OF FIFTH DL11 (LOCAL OR REMOTE)
DLC$5     = nnnnnn  ;CSR OF SIXTH DL11 (LOCAL OR REMOTE)
DLV$5     = nnn     ;VECTOR OF SIXTH DL11 (LOCAL OR REMOTE)
DLC$6     = nnnnnn  ;CSR OF SEVENTH DL11(LOCAL OR REMOTE)
DLV$6     = nnn     ;VECTOR OF SEVENTH DL11 (LOCAL OR REMOTE)
DLC$7     = nnnnnn  ;CSR OF EIGHTH DL11 (LOCAL OR REMOTE)
DLV$7     = nnn     ;VECTOR OF EIGHTH DL11 (LOCAL OR REMOTE)
DX$CSR    = nnnnnn  ;STATUS REGISTER FOR FIRST RX11
DX$VEC    = nnn     ;VECTOR FOR FIRST RX11
DXT$O     = 1       ;SECOND RX11 CONTROLLER IS PRESENT
DX$CS2    = nnnnnn  ;STATUS REGISTER FOR SECOND RX01
DX$VC2    = nnn     ;VECTOR FOR SECOND RX11
DY$DD     = 1       ;RX02 DOUBLE DENSITY ONLY
DY$CSR    = nnnnnn  ;STATUS REGISTER FOR FIRST RX02
DY$VEC    = nnn     ;VECTOR FOR FIRST RX02
DYT$O     = 1       ;SECOND RX02 CONTROLLER
DY$CS2    = nnnnnn  ;STATUS REGISTER FOR SECOND RX02
DY$VC2    = nnn     ;VECTOR FOR SECOND RX02
DZ11$N    = 0       ;NO DZ11 OR DZV11 SUPPORT
DZ11$N    = n       ;NUMBER OF DZ11 OR DZV11 MULTIPLEXORS SUPPORTED
DZV$11    = 0       ;DZ11 MULTIPLEXOR SUPPORT
DZV$11    = 1       ;DZV11 MULTIPLEXOR SUPPORT
DZCS$0    = nnnnnn  ;CSR FOR FIRST DZ11
DZVC$0    = nnn     ;VECTOR FOR FIRST DZ11
DZCS$1    = nnnnnn  ;CSR FOR SECOND DZ11
DZVC$1    = nnn     ;VECTOR FOR SECOND DZ11
DZCS$2    = nnnnnn  ;CSR FOR THIRD DZV11
DZVC$2    = nnn     ;VECTOR FOR THIRD DZV11
DZCS$3    = nnnnnn  ;CSR FOR FOURTH DZV11
DZVC$3    = nnn     ;VECTOR FOR FOURTH DZV11
DZ11$L    = n       ;NUMBER OF LOCAL LINES
DZ11$M    = n       ;NUMBER OF REMOTE LINES
DZSP$D    = n       ;BAUD RATE
DZST$P    = n       ;STOP UNIT

ERL$G     = 1       ;ERROR LOG SUPPORT
ERL$U     = n       ;NUMBER OF UNITS TO BE LOGGED

FPU$11    = 1       ;FLOATING POINT SUPPORT

HSR$B     = 1       ;HIGH SPEED RING BUFFER

KW11$P    = 1       ;USE KW11$P CLOCK AS SYSTEM CLOCK

LIGH$T    = 1       ;IDLE LOOP LIGHT PATTERN
L$ANG     = 0       ;NO LANGUAGE COMMANDS
L$ANG     = 1       ;LANGUAGE COMMANDS
LP$CSR    = nnnnnn  ;NON–STANDARD LINE PRINTER CSR
```

```
LP$VEC     = nnn      ;NON–STANDARD LINE PRINTER VECTOR
LS$CSR     = nnnnnn   ;CSR FOR SERIAL LINEPRINTER
LS$VEC     = nnn      ;VECTOR FOR SERIAL LINEPRINTER

MPT$Y      = 1        ;MEMORY PARITY
MTT$Y      = 1        ;MULTI–TERMINAL SUPPORT
MAT$S      = 1        ;ASYNCHRONOUS TERMINAL STATUS
MTI$M      = 1        ;TERMINAL TIME OUT
M$INI      = 0        ;NO MINIMAL COMMANDS
M$INI      = 1        ;MINIMAL COMMANDS
MT$FSM     = 1        ;TM11 MAGTAPE FILE–STRUCTURE SUPPORT
MT$UN      = n        ;NUMBER OF TM11 MAGTAPE UNITS
MM$FSM     = 1        ;TJU16 MAGTAPE FILE–STRUCTURE SUPPORT
MM$UN      = n        ;NUMBER OF TJU16 MAGTAPE UNITS
MS$FSM     = 1        ;TS11 MAGTAPE FILE–STRUCTURED SUPPORT
MS$UN      = n        ;NUMBER OF TS11 MAGTAPE UNITS
MS$CSR     = nnnnnn   ;STARTING STATUS WORD OF FIRST TS11 UNIT
MS$VEC     = nnn      ;VECTOR OF FIRST TS11 UNIT
MS$CS1     = nnnnnn   ;STARTING STATUS WORD OF SECOND TS11 UNIT
MS$VC1     = nnn      ;VECTOR OF SECOND TS11 UNIT
MS$CS2     = nnnnnn   ;STARTING STATUS WORD OF THIRD TS11 UNIT
MS$VC2     = nnn      ;VECTOR OF THIRD TS11 UNIT
MS$CS3     = nnnnnn   ;STARTING STATUS WORD OF FOURTH TS11 UNIT
MS$VC3     = nnn      ;VECTOR OF FOURTH TS11 UNIT
MS$CS4     = nnnnnn   ;STARTING STATUS WORD OF FIFTH TS11 UNIT
MS$VC4     = nnn      ;VECTOR OF FIFTH TS11 UNIT
MS$CS5     = nnnnnn   ;STARTING STATUS WORD OF SIXTH TS11 UNIT
MS$VC5     = nnn      ;VECTOR OF SIXTH TS11 UNIT
MS$CS6     = nnnnnn   ;STARTING STATUS WORD OF SEVENTH TS11 UNIT
MS$VC6     = nnn      ;VECTOR OF SEVENTH TS11 UNIT
MS$CS7     = nnnnnn   ;STARTING STATUS WORD OF EIGHTH TS11 UNIT
MS$VC7     = nnn      ;VECTOR OF EIGHTH TS11 UNIT

PR11$X     = 0        ;PC11 HIGH–SPEED READER/PUNCH
PR11$X     = 1        ;PR11 HIGH–SPEED READER
PWF$L      = 1        ;POWER FAIL

$RFNUM     = n        ;NUMBER OF RF11 PLATTERS
RDF$L      = 0        ;NO SYSTEM I/O ERROR MESSAGE
RDF$L      = 1        ;ADD SYSTEM I/O ERROR MESSAGE
ROL$OV     = 1        ;MONTH ROLLOVER SUPPORT
RJS0$3     = 0        ;RJS DISK IS RJS04
RJS0$3     = 1        ;RJS DISK IS RJS03
RP0$3      = 0        ;RP11 DISK IS RP02/RPR02
RP0$3      = 1        ;RP11 DISK IS RP03

SPC$PS     = 1        ;SAVE/SET MAIN – LINE PC AND PS SUPPORT
STAR$T     = 1        ;STARTUP COMMAND FILE
SYT$K      = 1        ;SYSTEM JOB SUPPORT
SYSG$N     = 1        ;INDICATE SYSGENED MONITOR
```

```
TIME$R      = 1        ;ADD TIMER SUPPORT
TIM$IT      = 1        ;ADD DEVICE TIMER SUPPORT
TIM$IT      = 1        ;DEVICE TIME OUT
TTYOUT      = nnn      ;SIZE OF OUTPUT RING BUFFERS
TTYIN       = nnn      ;SIZE OF INPUT RING BUFFERS

U.K.        = 1        ;UK MODEM SUPPORT
U$TIL       = 0        ;NO UTILITY COMMANDS
U$TIL       = 1        ;UTILITY COMMANDS

VS60$       = 1        ;VS60 DISPLAY SUPPORT
VT11$       = 1        ;VT11 DISPLAY SUPPORT
VT.CSR      = nnnnnn   ;CSR FOR VT11/VS60
VT.VEC      = nnn      ;VECTOR FOR VT11/VS60
```

F.2 Individual Keyboard Monitor Command Conditionals

To include or exclude individual keyboard commands, insert the particular conditional in SYCND.MAC and substitute 0 (to exclude the command) or 1 (to include the command) for n.

Utility Program Subset

```
U$TIL       = n        ;IF N IS SET EQUAL TO 0, ALL
                       ;UTILITY COMMANDS DEFAULT = 0
                       ;IF N IS SET EQUAL TO 1, ALL
                       ;UTILITY COMMANDS DEFAULT = 1

COPY$$      = n        ;COPY
DIRE$$      = n        ;DIRECTORY
EDIT$$      = n        ;EDIT
SQUE$$      = n        ;SQUEEZE
INIT$$      = n        ;INITIALIZE
FORM$$      = n        ;FORMAT
DUMP$$      = n        ;DUMP
DIFF$$      = n        ;DIFFERENCES
TYPE$$      = n        ;TYPE
PRIN$$      = n        ;PRINT
RENA$$      = n        ;RENAME
DELE$$      = n        ;DELETE
BOOT$$      = n        ;BOOT
SHOW$$      = n        ;SHOW
MAKE$$      = n        ;MAKE
TECO$$      = n        ;TECO
MUNG$$      = n        ;MUNG
CREA$$      = n        ;CREATE
```

Language Subset

```
L$ANG       = n       ;IF N IS SET EQUAL TO 0, ALL
                      ;LANGUAGE COMMANDS DEFAULT = 0
                      ;IF N IS SET EQUAL TO 1, ALL
                      ;LANGUAGE COMMANDS DEFAULT = 1

APL$$       = n       ;APL
BASI$$      = n       ;BASIC
FOCA$$      = n       ;FOCAL
FORT$$      = n       ;FORTRAN
MACR$$      = n       ;MACRO
DIBO$$      = n       ;DIBOL
COMP$$      = n       ;COMPILE
LINK$$      = n       ;LINK
LIBR$$      = n       ;LIBRARY
EXEC$$      = n       ;EXECUTE
PASC$$      = n       ;PASCAL
```

Minimal Subset

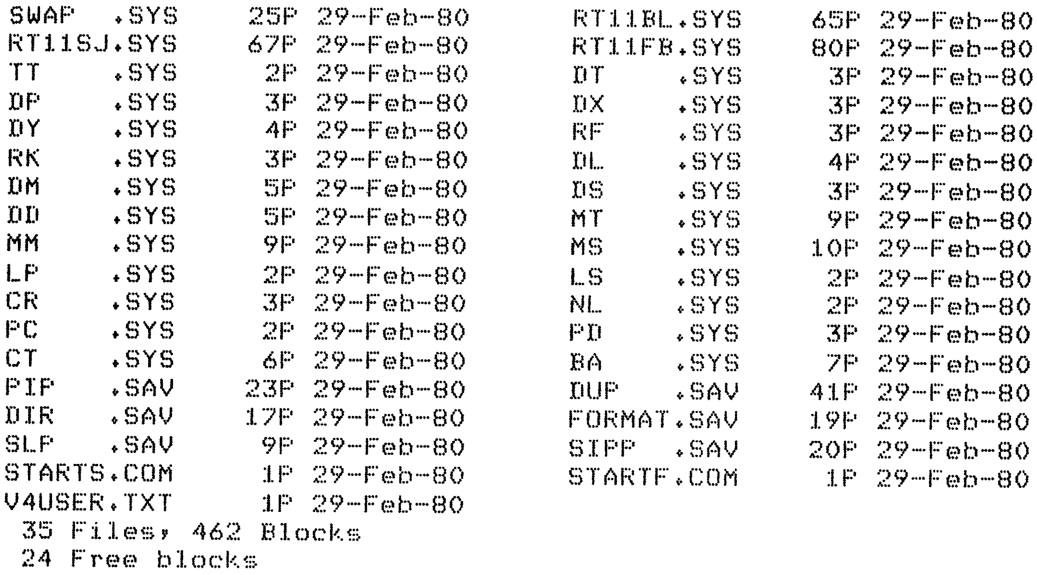
```
M$INI       = n       ;IF N IS SET EQUAL TO 0, ALL
                      ;MINIMAL COMMANDS DEFAULT = 0
                      ;IF N IS SET EQUAL TO 1, ALL
                      ;MINIMAL COMMANDS DEFAULT = 1

E$$         = n       ;EXAMINE
B$$         = n       ;BASE
D$$         = n       ;DEPOSIT
DATE$$      = n       ;DATE
TIME$$      = n       ;TIME
CLOS$$      = n       ;CLOSE
GT$$        = n       ;GT ON/OFF
STAR$$      = n       ;START
FRUN$$      = n       ;FRUN
SRUN$$      = n       ;SRUN
INST$$      = n       ;INSTALL
REMO$$      = n       ;REMOVE
LOAD$$      = n       ;LOAD
UNLO$$      = n       ;UNLOAD
ASSI$$      = n       ;ASSIGN
DEAS$$      = n       ;DEASSIGN
REEN$$      = n       ;REENTER
SUSP$$      = n       ;SUSPEND
RESU$$      = n       ;RESUME
RESE$$      = n       ;RESET
SET$$       = n       ;SET
HELP$$      = n       ;HELP
GET$$       = n       ;GET
SAVE$$      = n       ;SAVE
```

Appendix G
Software Kit Maps

G.1 Single–Density Diskette Kit Directories

Diskette 1



```
SWAP  .SYS     25P 29-Feb-80        RT11BL.SYS     65P 29-Feb-80
RT11SJ.SYS     67P 29-Feb-80        RT11FB.SYS     80P 29-Feb-80
TT    .SYS      2P 29-Feb-80        DT    .SYS      3P 29-Feb-80
DP    .SYS      3P 29-Feb-80        DX    .SYS      3P 29-Feb-80
DY    .SYS      4P 29-Feb-80        RF    .SYS      3P 29-Feb-80
RK    .SYS      3P 29-Feb-80        DL    .SYS      4P 29-Feb-80
DM    .SYS      5P 29-Feb-80        DS    .SYS      3P 29-Feb-80
DD    .SYS      5P 29-Feb-80        MT    .SYS      9P 29-Feb-80
MM    .SYS      9P 29-Feb-80        MS    .SYS     10P 29-Feb-80
LP    .SYS      2P 29-Feb-80        LS    .SYS      2P 29-Feb-80
CR    .SYS      3P 29-Feb-80        NL    .SYS      2P 29-Feb-80
PC    .SYS      2P 29-Feb-80        PD    .SYS      3P 29-Feb-80
CT    .SYS      6P 29-Feb-80        BA    .SYS      7P 29-Feb-80
PIP   .SAV     23P 29-Feb-80        DUP   .SAV     41P 29-Feb-80
DIR   .SAV     17P 29-Feb-80        FORMAT.SAV     19P 29-Feb-80
SLP   .SAV      9P 29-Feb-80        SIPP  .SAV     20P 29-Feb-80
STARTS.COM      1P 29-Feb-80        STARTF.COM      1P 29-Feb-80
V4USER.TXT      1P 29-Feb-80
 35 Files, 462 Blocks
 24 Free blocks
```

Diskette 1–T

```
SWAP  .SYS     25P 29-Feb-80    RT11BL.SYS     65P 29-Feb-80
RT11SJ.SYS     67P 29-Feb-80    RT11FB.SYS     80P 29-Feb-80
TT    .SYS      2P 29-Feb-80    DT    .SYS      3P 29-Feb-80
DP    .SYS      3P 29-Feb-80    DX    .SYS      3P 29-Feb-80
DY    .SYS      4P 29-Feb-80    RF    .SYS      3P 29-Feb-80
RK    .SYS      3P 29-Feb-80    DL    .SYS      4P 29-Feb-80
DM    .SYS      5P 29-Feb-80    DS    .SYS      3P 29-Feb-80
DD    .SYS      5P 29-Feb-80    MT    .SYS      9P 29-Feb-80
MM    .SYS      9P 29-Feb-80    MS    .SYS     10P 29-Feb-80
LP    .SYS      2P 29-Feb-80    LS    .SYS      2P 29-Feb-80
CR    .SYS      3P 29-Feb-80    NL    .SYS      2P 29-Feb-80
PC    .SYS      2P 29-Feb-80    PD    .SYS      3P 29-Feb-80
CT    .SYS      6P 29-Feb-80    BA    .SYS      7P 29-Feb-80
PIP   .SAV     23P 29-Feb-80    DUP   .SAV     41P 29-Feb-80
DIR   .SAV     17P 29-Feb-80    FORMAT.SAV     19P 29-Feb-80
SLP   .SAV      9P 29-Feb-80    SIPP  .SAV     20P 29-Feb-80
STARTS.COM      1P 29-Feb-80    STARTF.COM      1P 29-Feb-80
V4USER.TXT      1P 29-Feb-80
 35 Files, 462 Blocks
 24 Free blocks
```

Diskette 2

```
MTHD  .SYS      4P 29-Feb-80    MMHD  .SYS      4P 29-Feb-80
MSHD  .SYS      4P 29-Feb-80    RESORC.SAV     15P 29-Feb-80
SYSMAC.SML     42P 29-Feb-80    EDIT  .SAV     19P 29-Feb-80
KED   .SAV     60P 29-Feb-80    K52   .SAV     55P 29-Feb-80
TECO  .SAV     50P 29-Feb-80    MACRO .SAV     51P 29-Feb-80
MAC8K .SAV     56P 29-Feb-80    CREF  .SAV      6P 29-Feb-80
LINK  .SAV     41P 29-Feb-80    LIBR  .SAV     22P 29-Feb-80
FILEX .SAV     18P 29-Feb-80    SRCCOM.SAV     13P 29-Feb-80
BINCOM.SAV     10P 29-Feb-80    DUMP  .SAV      8P 29-Feb-80
 18 Files, 478 Blocks
 8 Free blocks
```

Diskette 3

```
PATCH .SAV     10P 29-Feb-80    PAT   .SAV      8P 29-Feb-80
HELP  .SAV    107P 29-Feb-80    HELP  .EXE      7P 29-Feb-80
HELP  .MLB     98P 29-Feb-80    ERROUT.SAV     17P 29-Feb-80
SYSGEN.SAV     39P 29-Feb-80    SYSGEN.CND    134P 29-Feb-80
SYSTBL.CND     23P 29-Feb-80    SYSMAC.MAC     41P 29-Feb-80
 10 Files, 484 Blocks
 2 Free blocks
```

Diskette 4

```
BATCH .SAV    26P 29-Feb-80      QUEMAN.SAV    13P 29-Feb-80
QUEUE .REL    14P 29-Feb-80      SYSLIB.OBJ    47P 29-Feb-80
MDUP  .SAV    18P 29-Feb-80      MDUP  .MM     48P 29-Feb-80
MDUP  .MT     48P 29-Feb-80      MDUP  .MS     48P 29-Feb-80
MBOT16.BOT     1P 29-Feb-80      MBOOT .BOT     1P 29-Feb-80
MSBOOT.BOT     3P 29-Feb-80      DEMOBG.MAC     2P 29-Feb-80
DEMOFG.MAC     3P 29-Feb-80      DEMOX1.MAC     3P 29-Feb-80
DEMOF1.FOR     2P 29-Feb-80      DEMOED.TXT     1P 29-Feb-80
README.TXT    36P 29-Feb-80      VT52  .TEC     5P 29-Feb-80
VEG   .TEC     4P 29-Feb-80      EDIT  .TEC     1P 29-Feb-80
INSERT.TEC     2P 29-Feb-80      LOCAL .TEC     3P 29-Feb-80
SORT  .TEC     3P 29-Feb-80      VTEDIT.TEC    32P 29-Feb-80
TECO  .TC     23P 29-Feb-80      SEARCH.TEC     3P 29-Feb-80
SQU   .TEC    13P 29-Feb-80      TYPE  .TEC    12P 29-Feb-80
TECO  .INI    15P 29-Feb-80      ODT   .OBJ     9P 29-Feb-80
VDT   .OBJ     9P 29-Feb-80      VTMAC .MAC     7P 29-Feb-80
VTHDLR.OBJ     9P 29-Feb-80      PLOT55.OBJ     3P 29-Feb-80
TEST55.FOR     5P 29-Feb-80      GETSTR.FOR     2P 29-Feb-80
PUTSTR.FOR     2P 29-Feb-80
 37 Files, 476 Blocks
 10 Free blocks
```

Diskette 5

```
SJ    .MAC     1P 29-Feb-80      FB    .MAC     1P 29-Feb-80
XM    .MAC     1P 29-Feb-80      USR   .MAC    69P 29-Feb-80
KMON  .MAC   119P 29-Feb-80      RMONSJ.MAC    76P 29-Feb-80
RMONFB.MAC   138P 29-Feb-80      XMSUBS.MAC    27P 29-Feb-80
EDTGBL.MAC    26P 29-Feb-80
 9 Files, 458 Blocks
 28 Free blocks`
```

Diskette 6

```
BSTRAP.MAC    48P 29-Feb-80      TT    .MAC     9P 29-Feb-80
BA    .MAC    20P 29-Feb-80      KMOVLY.MAC   185P 29-Feb-80
MTTEMT.MAC    18P 29-Feb-80      MTTINT.MAC    45P 29-Feb-80
LP    .MAC     9P 29-Feb-80      LS    .MAC    10P 29-Feb-80
NL    .MAC     3P 29-Feb-80      DD    .MAC    22P 29-Feb-80
DL    .MAC    21P 29-Feb-80      RK    .MAC     9P 29-Feb-80
DS    .MAC     9P 29-Feb-80      DX    .MAC    15P 29-Feb-80
DY    .MAC    18P 29-Feb-80      PC    .MAC     5P 29-Feb-80
PD    .MAC     9P 29-Feb-80      DM    .MAC    20P 29-Feb-80
 18 Files, 475 Blocks
 11 Free blocks
```

Diskette 7

```
DP     .MAC     10P 29-Feb-80        DT     .MAC      8P 29-Feb-80
RF     .MAC      7P 29-Feb-80        CR     .MAC     14P 29-Feb-80
CT     .MAC     33P 29-Feb-80        TM     .MAC     25P 29-Feb-80
TJ     .MAC     30P 29-Feb-80        TS     .MAC     29P 29-Feb-80
FSM    .MAC     31P 29-Feb-80        SPEED  .SAV      4P 29-Feb-80
ELCOPY.MAC       8P 29-Feb-80        ELINIT.MAC      15P 29-Feb-80
ELTASK.MAC      15P 29-Feb-80        ERROUT.MAC      49P 29-Feb-80
ERRTXT.MAC       9P 29-Feb-80        SYCND  .BL       3P 29-Feb-80
SYSTBL.BL        4P 29-Feb-80        SYCND  .DIS      5P 29-Feb-80
SYCND  .HD       5P 29-Feb-80        SYSTBL.DIS       4P 29-Feb-80
RTBL   .MAP     16P 29-Feb-80        RTSJ   .MAP     16P 29-Feb-80
RTFB   .MAP     23P 29-Feb-80        DISMT1.COM       9P 29-Feb-80
DISMT2.COM       8P 29-Feb-80
 25 Files, 380 Blocks
 106 Free blocks
```

G.2 DECtape II Cartridge Kit Directories

Cartridge 1

```
SWAP  .SYS    25P 29-Feb-80       RT11BL.SYS    65P 29-Feb-80
TT    .SYS     2P 29-Feb-80       DT    .SYS     3P 29-Feb-80
DP    .SYS     3P 29-Feb-80       DX    .SYS     3P 29-Feb-80
DY    .SYS     4P 29-Feb-80       RF    .SYS     3P 29-Feb-80
RK    .SYS     3P 29-Feb-80       DL    .SYS     4P 29-Feb-80
DM    .SYS     5P 29-Feb-80       FILE1 .BAD     1
RT11SJ.SYS    67P 29-Feb-80       DS    .SYS     3P 29-Feb-80
DD    .SYS     5P 29-Feb-80       MT    .SYS     9P 29-Feb-80
MM    .SYS     9P 29-Feb-80       MS    .SYS    10P 29-Feb-80
LP    .SYS     2P 29-Feb-80       LS    .SYS     2P 29-Feb-80
CR    .SYS     3P 29-Feb-80       NL    .SYS     2P 29-Feb-80
PC    .SYS     2P 29-Feb-80       PD    .SYS     3P 29-Feb-80
CT    .SYS     6P 29-Feb-80       FILE2 .BAD     1
RT11FB.SYS    80P 29-Feb-80       BA    .SYS     7P 29-Feb-80
PIP   .SAV    23P 29-Feb-80       DIR   .SAV    17P 29-Feb-80
FILE3 .BAD     1                  DUP   .SAV    41P 29-Feb-80
FORMAT.SAV    19P 29-Feb-80       SLP   .SAV     9P 29-Feb-80
SIPP  .SAV    20P 29-Feb-80
 35 Files, 462 Blocks
 42 Free blocks
```

Cartridge 1–T

```
SWAP  .SYS    25P 29-Feb-80       RT11BL.SYS    65P 29-Feb-80
TT    .SYS     2P 29-Feb-80       DT    .SYS     3P 29-Feb-80
DP    .SYS     3P 29-Feb-80       DX    .SYS     3P 29-Feb-80
DY    .SYS     4P 29-Feb-80       RF    .SYS     3P 29-Feb-80
RK    .SYS     3P 29-Feb-80       DL    .SYS     4P 29-Feb-80
DM    .SYS     5P 29-Feb-80       FILE1 .BAD     1
RT11SJ.SYS    67P 29-Feb-80       DS    .SYS     3P 29-Feb-80
DD    .SYS     5P 29-Feb-80       MT    .SYS     9P 29-Feb-80
MM    .SYS     9P 29-Feb-80       MS    .SYS    10P 29-Feb-80
LP    .SYS     2P 29-Feb-80       LS    .SYS     2P 29-Feb-80
CR    .SYS     3P 29-Feb-80       NL    .SYS     2P 29-Feb-80
PC    .SYS     2P 29-Feb-80       PD    .SYS     3P 29-Feb-80
CT    .SYS     6P 29-Feb-80       FILE2 .BAD     1
RT11FB.SYS    80P 29-Feb-80       BA    .SYS     7P 29-Feb-80
PIP   .SAV    23P 29-Feb-80       DIR   .SAV    17P 29-Feb-80
FILE3 .BAD     1                  DUP   .SAV    41P 29-Feb-80
FORMAT.SAV    19P 29-Feb-80       SLP   .SAV     9P 29-Feb-80
SIPP  .SAV    20P 29-Feb-80
 35 Files, 462 Blocks
 42 Free blocks
```

Cartridge 2

```
MTHD  .SYS     4P 04-Mar-80    MMHD  .SYS     4P 04-Mar-80
MSHD  .SYS     4P 04-Mar-80    RESORC.SAV    15P 04-Mar-80
SYSMAC.SML    42P 04-Mar-80    EDIT  .SAV    19P 04-Mar-80
KED   .SAV    60P 04-Mar-80    K52   .SAV    55P 04-Mar-80
TECO  .SAV    50P 04-Mar-80    MACRO .SAV    51P 04-Mar-80
MAC8K .SAV    56P 04-Mar-80    CREF  .SAV     6P 04-Mar-80
LINK  .SAV    41P 04-Mar-80    LIBR  .SAV    22P 04-Mar-80
FILEX .SAV    18P 04-Mar-80    SRCCOM.SAV    13P 04-Mar-80
BINCOM.SAV    10P 04-Mar-80    DUMP  .SAV     8P 04-Mar-80
 18 Files, 478 Blocks
 26 Free blocks
```

Cartridge 3

```
PATCH .SAV    10P 04-Mar-80    PAT   .SAV     8P 04-Mar-80
HELP  .SAV   107P 04-Mar-80    HELP  .EXE     7P 04-Mar-80
HELP  .MLB    98P 04-Mar-80    ERROUT.SAV    17P 04-Mar-80
SYSGEN.SAV    39P 04-Mar-80    SYSGEN.CND   134P 04-Mar-80
SYSTBL.CND    23P 04-Mar-80    SYSMAC.MAC    41  29-Feb-80
 10 Files, 484 Blocks
 20 Free blocks
```

Cartridge 4

```
BATCH .SAV    26P 04-Mar-80    QUEMAN.SAV    13P 04-Mar-80
QUEUE .REL    14P 04-Mar-80    SYSLIB.OBJ    47P 04-Mar-80
MDUP  .SAV    18P 04-Mar-80    MDUP  .MM     48P 04-Mar-80
MDUP  .MT     48P 04-Mar-80    MDUP  .MS     48P 04-Mar-80
MBOOT .BOT     1P 04-Mar-80    MBOT16.BOT     1P 04-Mar-80
MSBOOT.BOT     3P 04-Mar-80    DEMOBG.MAC     2P 04-Mar-80
DEMOFG.MAC     3P 04-Mar-80    DEMOX1.MAC     3P 04-Mar-80
DEMOF1.FOR     2P 04-Mar-80    DEMOED.TXT     1P 04-Mar-80
README.TXT    36P 04-Mar-80    VT52  .TEC     5P 04-Mar-80
VEG   .TEC     4P 04-Mar-80    EDIT  .TEC     1P 04-Mar-80
INSERT.TEC     2P 04-Mar-80    LOCAL .TEC     3P 04-Mar-80
SORT  .TEC     3P 04-Mar-80    VTEDIT.TEC    32P 04-Mar-80
TECO  .TC     23P 04-Mar-80    SEARCH.TEC     3P 04-Mar-80
SQU   .TEC    113P 04-Mar-80     TYPE  .TEC    12P 04-Mar-80
TECO  .INI    15P 04-Mar-80    ODT   .OBJ     9P 04-Mar-80
VDT   .OBJ     9P 04-Mar-80    VTMAC .MAC     7P 04-Mar-80
VTHDLR.OBJ     9P 04-Mar-80    PLOT55.OBJ     3P 04-Mar-80
TEST55.FOR     5P 04-Mar-80    GETSTR.FOR     2P 04-Mar-80
PUTSTR.FOR     2P 04-Mar-80
 37 Files, 476 Blocks
 28 Free blocks
```

Cartridge 5

```
SJ     .MAC      1P 04-Mar-80     FB     .MAC      1P 04-Mar-80
XM     .MAC      1P 04-Mar-80     USR    .MAC     69P 04-Mar-80
KMON   .MAC    119P 04-Mar-80     RMONSJ.MAC      76P 04-Mar-80
RMONFB.MAC     138P 04-Mar-80     XMSUBS.MAC      27P 04-Mar-80
EDTGBL.MAC      26P 04-Mar-80
 9 Files, 458 Blocks
 46 Free blocks
```

Cartridge 6

```
BSTRAP.MAC      48P 04-Mar-80     TT     .MAC      9P 04-Mar-80
BA     .MAC     20P 04-Mar-80     KMOVLY.MAC     185P 04-Mar-80
MTTEMT.MAC      18P 04-Mar-80     MTTINT.MAC      45P 04-Mar-80
LP     .MAC      9P 04-Mar-80     LS     .MAC     10P 04-Mar-80
NL     .MAC      3P 04-Mar-80     DD     .MAC     22P 04-Mar-80
DL     .MAC     21P 04-Mar-80     RK     .MAC      9P 04-Mar-80
DS     .MAC      9P 04-Mar-80     DX     .MAC     15P 04-Mar-80
DY     .MAC     18P 04-Mar-80     PC     .MAC      5P 04-Mar-80
PD     .MAC      9P 04-Mar-80     DM     .MAC     20P 04-Mar-80
 18 Files, 475 Blocks
 29 Free blocks
```

Cartridge 7

```
DP     .MAC     10P 04-Mar-80     DT     .MAC      8P 04-Mar-80
RF     .MAC      7P 04-Mar-80     CR     .MAC     14P 04-Mar-80
CT     .MAC     33P 04-Mar-80     TM     .MAC     25P 04-Mar-80
TJ     .MAC     30P 04-Mar-80     TS     .MAC     29P 04-Mar-80
FSM    .MAC     31P 04-Mar-80     SPEED .SAV       4P 04-Mar-80
ELCOPY.MAC       8P 04-Mar-80     ELINIT.MAC      15P 04-Mar-80
ELTASK.MAC      15P 04-Mar-80     ERROUT.MAC      49P 04-Mar-80
ERRTXT.MAC       9P 04-Mar-80     SYCND .BL        3P 04-Mar-80
SYSTBL.BL        4P 04-Mar-80     SYCND .DIS       5P 04-Mar-80
SYCND .HD        5P 04-Mar-80     SYSTBL.DIS       4P 04-Mar-80
RTBL   .MAP     16P 04-Mar-80     RTSJ   .MAP     16P 04-Mar-80
RTFB   .MAP     23P 04-Mar-80     DISMT1.COM       9P 04-Mar-80
DISMT2.COM       8P 04-Mar-80
 25 Files, 380 Blocks
 124 Free blocks
```

G.3 Hard Disk Kit Directory

RL01 Disk

```
SWAP  .SYS    25P 21-Feb-80     RT11SJ.SYS    67P 21-Feb-80
RT11FB.SYS    80P 21-Feb-80     RT11BL.SYS    65P 21-Feb-80
TT    .SYS     2P 21-Feb-80     DT    .SYS     3P 21-Feb-80
DP    .SYS     3P 21-Feb-80     DX    .SYS     3P 21-Feb-80
DY    .SYS     4P 21-Feb-80     RF    .SYS     3P 21-Feb-80
RK    .SYS     3P 21-Feb-80     DL    .SYS     4P 21-Feb-80
DM    .SYS     5P 21-Feb-80     DS    .SYS     3P 21-Feb-80
DD    .SYS     5P 21-Feb-80     LP    .SYS     2P 21-Feb-80
LS    .SYS     2P 21-Feb-80     CR    .SYS     3P 21-Feb-80
MT    .SYS     9P 21-Feb-80     MTHD  .SYS     4P 21-Feb-80
MM    .SYS     9P 21-Feb-80     MMHD  .SYS     4P 21-Feb-80
MS    .SYS    10P 21-Feb-80     MSHD  .SYS     4P 21-Feb-80
NL    .SYS     2P 21-Feb-80     PC    .SYS     2P 21-Feb-80
PD    .SYS     3P 21-Feb-80     CT    .SYS     6P 21-Feb-80
BA    .SYS     7P 21-Feb-80     PIP   .SAV    23P 21-Feb-80
DUP   .SAV    41P 21-Feb-80     FORMAT.SAV    19P 21-Feb-80
RESORC.SAV    15P 21-Feb-80     DIR   .SAV    17P 21-Feb-80
SYSMAC.SML    42P 21-Feb-80     EDIT  .SAV    19P 21-Feb-80
KED   .SAV    60P 21-Feb-80     K52   .SAV    55P 21-Feb-80
TECO  .SAV    50P 21-Feb-80     MACRO .SAV    51P 21-Feb-80
MAC8K .SAV    56P 21-Feb-80     CREF  .SAV     6P 21-Feb-80
LINK  .SAV    41P 21-Feb-80     LIBR  .SAV    22P 21-Feb-80
FILEX .SAV    18P 21-Feb-80     SRCCOM.SAV    13P 21-Feb-80
BINCOM.SAV    10P 21-Feb-80     SLP   .SAV     9P 21-Feb-80
DUMP  .SAV     8P 21-Feb-80     SIPP  .SAV    20P 21-Feb-80
PATCH .SAV    10P 21-Feb-80     PAT   .SAV     8P 21-Feb-80
HELP  .SAV   107P 21-Feb-80     HELP  .EXE     7P 21-Feb-80
HELP  .MLB    98P 21-Feb-80     BATCH .SAV    26P 21-Feb-80
ERROUT.SAV    17P 21-Feb-80     SYSGEN.SAV    39P 21-Feb-80
SYSGEN.CND   134P 21-Feb-80     SYSTBL.CND    23P 21-Feb-80
QUEMAN.SAV    13P 21-Feb-80     QUEUE .REL    14P 21-Feb-80
README.TXT    36P 21-Feb-80     VT52  .TEC     5P 21-Feb-80
VEG   .TEC     4P 21-Feb-80     EDIT  .TEC     1P 21-Feb-80
INSERT.TEC     2P 21-Feb-80     LOCAL .TEC     3P 21-Feb-80
SORT  .TEC     3P 21-Feb-80     VTEDIT.TEC    32P 21-Feb-80
TECO  .TC     23P 21-Feb-80     SEARCH.TEC     3P 21-Feb-80
SQU   .TEC    13P 21-Feb-80     TYPE  .TEC    12P 21-Feb-80
TECO  .INI    15P 21-Feb-80     ODT   .OBJ     9P 21-Feb-80
VDT   .OBJ     9P 21-Feb-80     VTMAC .MAC     7P 21-Feb-80
VTHDLR.OBJ     9P 21-Feb-80     PLOT55.OBJ     3P 21-Feb-80
TEST55.FOR     5P 21-Feb-80     SYSLIB.OBJ    47P 21-Feb-80
GETSTR.FOR     2P 21-Feb-80     PUTSTR.FOR     2P 21-Feb-80
SYSMAC.MAC    41P 21-Feb-80     MDUP  .SAV    18P 21-Feb-80
MBOOT .BOT     1P 21-Feb-80     MBOT16.BOT     1P 21-Feb-80
MSBOOT.BOT     3P 21-Feb-80     SPEED .SAV     4P 21-Feb-80
DEMOBG.MAC     2P 21-Feb-80     DEMOFG.MAC     3P 21-Feb-80
DEMOX1.MAC     3P 21-Feb-80     DEMOF1.FOR     2P 21-Feb-80
DEMOED.TXT     1P 21-Feb-80     STARTS.COM     1P 21-Feb-80
STARTF.COM     1P 21-Feb-80     V4USER.TXT     1P 21-Feb-80
SJ    .MAC     1P 21-Feb-80     FB    .MAC     1P 21-Feb-80
XM    .MAC     1P 21-Feb-80     KMON  .MAC   119P 21-Feb-80
USR   .MAC    69P 21-Feb-80     RMONSJ.MAC    76P 21-Feb-80
RMONFB.MAC   138P 21-Feb-80     XMSUBS.MAC    27P 21-Feb-80
KMOVLY.MAC   185P 21-Feb-80     EDTGBL.MAC    26P 21-Feb-80
BSTRAP.MAC    48P 21-Feb-80     MTTEMT.MAC    18P 21-Feb-80
MTTINT.MAC    45P 21-Feb-80     TT    .MAC     9P 21-Feb-80
BA    .MAC    20P 21-Feb-80     LP    .MAC     9P 21-Feb-80
LS    .MAC    10P 21-Feb-80     DD    .MAC    22P 21-Feb-80
DL    .MAC    21P 21-Feb-80     RK    .MAC     9P 21-Feb-80
DT    .MAC     8P 21-Feb-80     DS    .MAC     9P 21-Feb-80
PC    .MAC     5P 21-Feb-80     PD    .MAC     9P 21-Feb-80
RF    .MAC     7P 21-Feb-80     DM    .MAC    20P 21-Feb-80
DX    .MAC    15P 21-Feb-80     DY    .MAC    18P 21-Feb-80
CR    .MAC    14P 21-Feb-80     DP    .MAC    10P 21-Feb-80
CT    .MAC    33P 21-Feb-80     TM    .MAC    25P 21-Feb-80
TJ    .MAC    30P 21-Feb-80     TS    .MAC    29P 21-Feb-80
NL    .MAC     3P 21-Feb-80     FSM   .MAC    31P 21-Feb-80
ELCOPY.MAC     8P 21-Feb-80     ELINIT.MAC    15P 21-Feb-80
ELTASK.MAC    15P 21-Feb-80     ERROUT.MAC    49P 21-Feb-80
ERRTXT.MAC     9P 21-Feb-80     DISMT1.COM     9P 21-Feb-80
DISMT2.COM     8P 21-Feb-80     SYCND .BL      3P 21-Feb-80
SYSTBL.BL      4P 21-Feb-80     SYCND .DIS     5P 21-Feb-80
SYSTBL.DIS     4P 21-Feb-80     SYCND .HD      5P 21-Feb-80
RTBL  .MAP    16P 21-Feb-80     RTSJ  .MAP    16P 21-Feb-80
RTFB  .MAP    23P 21-Feb-80     MDUP  .MT     48P 21-Feb-80
MDUP  .MM     48P 21-Feb-80     MDUP  .MS     48P 21-Feb-80
 152 Files, 3213 Blocks
 6959 Free blocks
```

RL02 Disk

```
SWAP  .SYS    25P 21-Feb-80     RT11SJ.SYS    67P 21-Feb-80
RT11FB.SYS    80P 21-Feb-80     RT11BL.SYS    65P 21-Feb-80
TT    .SYS     2P 21-Feb-80     DT    .SYS     3P 21-Feb-80
DP    .SYS     3P 21-Feb-80     DX    .SYS     3P 21-Feb-80
DY    .SYS     4P 21-Feb-80     RF    .SYS     3P 21-Feb-80
RK    .SYS     3P 21-Feb-80     DL    .SYS     4P 21-Feb-80
DM    .SYS     5P 21-Feb-80     DS    .SYS     3P 21-Feb-80
DD    .SYS     5P 21-Feb-80     LP    .SYS     2P 21-Feb-80
LS    .SYS     2P 21-Feb-80     CR    .SYS     3P 21-Feb-80
MT    .SYS     9P 21-Feb-80     MTHD  .SYS     4P 21-Feb-80
MM    .SYS     9P 21-Feb-80     MMHD  .SYS     4P 21-Feb-80
MS    .SYS    10P 21-Feb-80     MSHD  .SYS     4P 21-Feb-80
NL    .SYS     2P 21-Feb-80     PC    .SYS     2P 21-Feb-80
PD    .SYS     3P 21-Feb-80     CT    .SYS     6P 21-Feb-80
BA    .SYS     7P 21-Feb-80     PIP   .SAV    23P 21-Feb-80
DUP   .SAV    41P 21-Feb-80     FORMAT.SAV    19P 21-Feb-80
RESORC.SAV    15P 21-Feb-80     DIR   .SAV    17P 21-Feb-80
SYSMAC.SML    42P 21-Feb-80     EDIT  .SAV    19P 21-Feb-80
KED   .SAV    60P 21-Feb-80     K52   .SAV    55P 21-Feb-80
TECO  .SAV    50P 21-Feb-80     MACRO .SAV    51P 21-Feb-80
MAC8K .SAV    56P 21-Feb-80     CREF  .SAV     6P 21-Feb-80
LINK  .SAV    41P 21-Feb-80     LIBR  .SAV    22P 21-Feb-80
FILEX .SAV    18P 21-Feb-80     SRCCOM.SAV    13P 21-Feb-80
BINCOM.SAV    10P 21-Feb-80     SLP   .SAV     9P 21-Feb-80
DUMP  .SAV     8P 21-Feb-80     SIPP  .SAV    20P 21-Feb-80
PATCH .SAV    10P 21-Feb-80     PAT   .SAV     8P 21-Feb-80
HELP  .SAV   107P 21-Feb-80     HELP  .EXE     7P 21-Feb-80
HELP  .MLB    98P 21-Feb-80     BATCH .SAV    26P 21-Feb-80
ERROUT.SAV    17P 21-Feb-80     SYSGEN.SAV    39P 21-Feb-80
SYSGEN.CND   134P 21-Feb-80     SYSTBL.CND    23P 21-Feb-80
QUEMAN.SAV    13P 21-Feb-80     QUEUE .REL    14P 21-Feb-80
README.TXT    36P 21-Feb-80     VT52  .TEC     5P 21-Feb-80
VEG   .TEC     4P 21-Feb-80     EDIT  .TEC     1P 21-Feb-80
INSERT.TEC     2P 21-Feb-80     LOCAL .TEC     3P 21-Feb-80
SORT  .TEC     3P 21-Feb-80     VTEDIT.TEC    32P 21-Feb-80
TECO  .TC     23P 21-Feb-80     SEARCH.TEC     3P 21-Feb-80
SQU   .TEC    13P 21-Feb-80     TYPE  .TEC    12P 21-Feb-80
TECO  .INI    15P 21-Feb-80     ODT   .OBJ     9P 21-Feb-80
VDT   .OBJ     9P 21-Feb-80     VTMAC .MAC     7P 21-Feb-80
VTHDLR.OBJ     9P 21-Feb-80     PLOT55.OBJ     3P 21-Feb-80
TEST55.FOR     5P 21-Feb-80     SYSLIB.OBJ    47P 21-Feb-80
GETSTR.FOR     2P 21-Feb-80     PUTSTR.FOR     2P 21-Feb-80
SYSMAC.MAC    41P 21-Feb-80     MDUP  .SAV    18P 21-Feb-80
MBOOT .BOT     1P 21-Feb-80     MBOT16.BOT     1P 21-Feb-80
MSBOOT.BOT     3P 21-Feb-80     SPEED .SAV     4P 21-Feb-80
DEMOBG.MAC     2P 21-Feb-80     DEMOFG.MAC     3P 21-Feb-80
DEMOX1.MAC     3P 21-Feb-80     DEMOF1.FOR     2P 21-Feb-80
DEMOED.TXT     1P 21-Feb-80     STARTS.COM     1P 21-Feb-80
STARTF.COM     1P 21-Feb-80     V4USER.TXT     1P 21-Feb-80
SJ    .MAC     1P 21-Feb-80     FB    .MAC     1P 21-Feb-80
XM    .MAC     1P 21-Feb-80     KMON  .MAC   119P 21-Feb-80
USR   .MAC    69P 21-Feb-80     RMONSJ.MAC    76P 21-Feb-80
RMONFB.MAC   138P 21-Feb-80     XMSUBS.MAC    27P 21-Feb-80
KMOVLY.MAC   185P 21-Feb-80     EDTGBL.MAC    26P 21-Feb-80
BSTRAP.MAC    48P 21-Feb-80     MTTEMT.MAC    18P 21-Feb-80
MTTINT.MAC    45P 21-Feb-80     TT    .MAC     9P 21-Feb-80
BA    .MAC    20P 21-Feb-80     LP    .MAC     9P 21-Feb-80
LS    .MAC    10P 21-Feb-80     DD    .MAC    22P 21-Feb-80
DL    .MAC    21P 21-Feb-80     RK    .MAC     9P 21-Feb-80
DT    .MAC     8P 21-Feb-80     DS    .MAC     9P 21-Feb-80
PC    .MAC     5P 21-Feb-80     PD    .MAC     9P 21-Feb-80
RF    .MAC     7P 21-Feb-80     DM    .MAC    20P 21-Feb-80
DX    .MAC    15P 21-Feb-80     DY    .MAC    18P 21-Feb-80
CR    .MAC    14P 21-Feb-80     DP    .MAC    10P 21-Feb-80
CT    .MAC    33P 21-Feb-80     TM    .MAC    25P 21-Feb-80
TJ    .MAC    30P 21-Feb-80     TS    .MAC    29P 21-Feb-80
NL    .MAC     3P 21-Feb-80     FSM   .MAC    31P 21-Feb-80
ELCOPY.MAC     8P 21-Feb-80     ELINIT.MAC    15P 21-Feb-80
ELTASK.MAC    15P 21-Feb-80     ERROUT.MAC    49P 21-Feb-80
ERRTXT.MAC     9P 21-Feb-80     DISMT1.COM     7P 21-Feb-80
DISMT2.COM     8P 21-Feb-80     SYCND .BL      3P 21-Feb-80
SYSTBL.BL      4P 21-Feb-80     SYCND .DIS     5P 21-Feb-80
SYSTBL.DIS     4P 21-Feb-80     SYCND .HD      5P 21-Feb-80
RTBL  .MAP    16P 21-Feb-80     RTSJ  .MAP    16P 21-Feb-80
RTFB  .MAP    23P 21-Feb-80     MDUP  .MT     48P 21-Feb-80
MDUP  .MM     48P 21-Feb-80     MDUP  .MS     48P 21-Feb-80
 152 Files, 3213 Blocks
 17169 Free blocks
```

RK05 Disk

```
SWAP   .SYS     25P 21-Feb-80      RT11SJ.SYS     67P 21-Feb-80
RT11FB.SYS     80P 21-Feb-80      RT11BL.SYS     65P 21-Feb-80
TT     .SYS      2P 21-Feb-80      DT     .SYS      3P 21-Feb-80
DP     .SYS      3P 21-Feb-80      DX     .SYS      3P 21-Feb-80
DY     .SYS      4P 21-Feb-80      RF     .SYS      3P 21-Feb-80
RK     .SYS      3P 21-Feb-80      DL     .SYS      4P 21-Feb-80
DM     .SYS      5P 21-Feb-80      DS     .SYS      3P 21-Feb-80
DD     .SYS      5P 21-Feb-80      LP     .SYS      2P 21-Feb-80
LS     .SYS      2P 21-Feb-80      CR     .SYS      3P 21-Feb-80
MT     .SYS      9P 21-Feb-80      MTHD   .SYS      4P 21-Feb-80
MM     .SYS      9P 21-Feb-80      MMHD   .SYS      4P 21-Feb-80
MS     .SYS     10P 21-Feb-80      MSHD   .SYS      4P 21-Feb-80
NL     .SYS      2P 21-Feb-80      PC     .SYS      2P 21-Feb-80
PD     .SYS      3P 21-Feb-80      CT     .SYS      6P 21-Feb-80
BA     .SYS      7P 21-Feb-80      PIP    .SAV     23P 21-Feb-80
DUP    .SAV     41P 21-Feb-80      FORMAT.SAV     19P 21-Feb-80
RESORC.SAV     15P 21-Feb-80      DIR    .SAV     17P 21-Feb-80
SYSMAC.SML     42P 21-Feb-80      EDIT   .SAV     19P 21-Feb-80
KED    .SAV     60P 21-Feb-80      K52    .SAV     55P 21-Feb-80
TECO   .SAV     50P 21-Feb-80      MACRO  .SAV     51P 21-Feb-80
MAC8K  .SAV     56P 21-Feb-80      CREF   .SAV      6P 21-Feb-80
LINK   .SAV     41P 21-Feb-80      LIBR   .SAV     22P 21-Feb-80
FILEX  .SAV     18P 21-Feb-80      SRCCOM.SAV     13P 21-Feb-80
BINCOM.SAV     10P 21-Feb-80      SLP    .SAV      9P 21-Feb-80
DUMP   .SAV      8P 21-Feb-80      SIPP   .SAV     20P 21-Feb-80
PATCH  .SAV     10P 21-Feb-80      PAT    .SAV      8P 21-Feb-80
HELP   .SAV    107P 21-Feb-80      HELP   .EXE      7P 21-Feb-80
HELP   .MLB     98P 21-Feb-80      BATCH  .SAV     26P 21-Feb-80
ERROUT.SAV     17P 21-Feb-80      SYSGEN.SAV     39P 21-Feb-80
SYSGEN.CND    134P 21-Feb-80      SYSTBL.CND     23P 21-Feb-80
QUEMAN.SAV     13P 21-Feb-80      QUEUE  .REL     14P 21-Feb-80
README.TXT     36P 21-Feb-80      VT52   .TEC      5P 21-Feb-80
VEG    .TEC      4P 21-Feb-80      EDIT   .TEC      1P 21-Feb-80
INSERT.TEC      2P 21-Feb-80      LOCAL  .TEC      3P 21-Feb-80
SORT   .TEC      3P 21-Feb-80      VTEDIT.TEC     32P 21-Feb-80
TECO   .TC      23P 21-Feb-80      SEARCH.TEC      3P 21-Feb-80
SQU    .TEC     13P 21-Feb-80      TYPE   .TEC     12P 21-Feb-80
TECO   .INI     15P 21-Feb-80      ODT    .OBJ      9P 21-Feb-80
VDT    .OBJ      9P 21-Feb-80      VTMAC  .MAC      7P 21-Feb-80
VTHDLR.OBJ      9P 21-Feb-80      PLOT55.OBJ      3P 21-Feb-80
TEST55.FOR      5P 21-Feb-80      SYSLIB.OBJ     47P 21-Feb-80
GETSTR.FOR      2P 21-Feb-80      PUTSTR.FOR      2P 21-Feb-80
SYSMAC.MAC     41P 21-Feb-80      MDUP   .SAV     18P 21-Feb-80
MBOOT  .BOT      1P 21-Feb-80      MBOT16.BOT      1P 21-Feb-80
MSBOOT.BOT      3P 21-Feb-80      SPEED  .SAV      4P 21-Feb-80
DEMOBG.MAC      2P 21-Feb-80      DEMOFG.MAC      3P 21-Feb-80
DEMOX1.MAC      3P 21-Feb-80      DEMOF1.FOR      2P 21-Feb-80
DEMOED.TXT      1P 21-Feb-80      STARTS.COM      1P 21-Feb-80
STARTF.COM      1P 21-Feb-80      V4USER.TXT      1P 21-Feb-80
SJ     .MAC      1P 21-Feb-80      FB     .MAC      1P 21-Feb-80
XM     .MAC      1P 21-Feb-80      KMON   .MAC    119P 21-Feb-80
USR    .MAC     69P 21-Feb-80      RMONSJ.MAC     76P 21-Feb-80
RMONFB.MAC    138P 21-Feb-80      XMSUBS.MAC     27P 21-Feb-80
KMOVLY.MAC    185P 21-Feb-80      EDTGBL.MAC     26P 21-Feb-80
BSTRAP.MAC     48P 21-Feb-80      MTTEMT.MAC     18P 21-Feb-80
MTTINT.MAC     45P 21-Feb-80      TT     .MAC      9P 21-Feb-80
BA     .MAC     20P 21-Feb-80      LP     .MAC      9P 21-Feb-80
LS     .MAC     10P 21-Feb-80      DD     .MAC     22P 21-Feb-80
DL     .MAC     21P 21-Feb-80      RK     .MAC      9P 21-Feb-80
DT     .MAC      8P 21-Feb-80      DS     .MAC      9P 21-Feb-80
PC     .MAC      5P 21-Feb-80      PD     .MAC      9P 21-Feb-80
RF     .MAC      7P 21-Feb-80      DM     .MAC     20P 21-Feb-80
DX     .MAC     15P 21-Feb-80      DY     .MAC     18P 21-Feb-80
CR     .MAC     14P 21-Feb-80      DP     .MAC     10P 21-Feb-80
CT     .MAC     33P 21-Feb-80      TM     .MAC     25P 21-Feb-80
TJ     .MAC     30P 21-Feb-80      TS     .MAC     29P 21-Feb-80
NL     .MAC      3P 21-Feb-80      FSM    .MAC     31P 21-Feb-80
ELCOPY.MAC      8P 21-Feb-80      ELINIT.MAC     15P 21-Feb-80
ELTASK.MAC     15P 21-Feb-80      ERROUT.MAC     49P 21-Feb-80
ERRTXT.MAC      9P 21-Feb-80      DISMT1.COM      9P 21-Feb-80
DISMT2.COM      8P 21-Feb-80      SYCND  .BL       3P 21-Feb-80
SYSTBL.BL       4P 21-Feb-80      SYCND  .DIS      5P 21-Feb-80
SYSTBL.DIS      4P 21-Feb-80      SYCND  .HD       5P 21-Feb-80
RTBL   .MAP     16P 21-Feb-80      RTSJ   .MAP     16P 21-Feb-80
RTFB   .MAP     23P 21-Feb-80      MDUP   .MT      48P 21-Feb-80
MDUP   .MM      48P 21-Feb-80      MDUP   .MS      48P 21-Feb-80
 152 Files, 3213 Blocks
 1549 Free blocks
```

G.4 Double-Density Diskette Kit Directories

Diskette 1

```
SWAP  .SYS    25P 01-Mar-80        RT11BL.SYS    65P 01-Mar-80
RT11SJ.SYS    67P 01-Mar-80        RT11FB.SYS    80P 01-Mar-80
TT    .SYS     2P 01-Mar-80        DT    .SYS     3P 01-Mar-80
DP    .SYS     3P 01-Mar-80        DX    .SYS     3P 01-Mar-80
DY    .SYS     4P 01-Mar-80        RF    .SYS     3P 01-Mar-80
RK    .SYS     3P 01-Mar-80        DL    .SYS     4P 01-Mar-80
DM    .SYS     5P 01-Mar-80        DS    .SYS     3P 01-Mar-80
DD    .SYS     5P 01-Mar-80        MT    .SYS     9P 01-Mar-80
MM    .SYS     9P 01-Mar-80        MS    .SYS    10P 01-Mar-80
LP    .SYS     2P 01-Mar-80        LS    .SYS     2P 01-Mar-80
CR    .SYS     3P 01-Mar-80        NL    .SYS     2P 01-Mar-80
PC    .SYS     2P 01-Mar-80        PD    .SYS     3P 01-Mar-80
CT    .SYS     6P 01-Mar-80        BA    .SYS     7P 01-Mar-80
PIP   .SAV    23P 01-Mar-80        DUP   .SAV    41P 01-Mar-80
DIR   .SAV    17P 01-Mar-80        FORMAT.SAV    19P 01-Mar-80
SLP   .SAV     9P 01-Mar-80        SIPP  .SAV    20P 01-Mar-80
MTHD  .SYS     4P 01-Mar-80        MMHD  .SYS     4P 01-Mar-80
MSHD  .SYS     4P 01-Mar-80        RESORC.SAV    15P 01-Mar-80
SYSMAC.SML    42P 01-Mar-80        EDIT  .SAV    19P 01-Mar-80
KED   .SAV    60P 01-Mar-80        K52   .SAV    55P 01-Mar-80
TECO  .SAV    50P 01-Mar-80        MACRO .SAV    51P 01-Mar-80
MAC8K .SAV    56P 01-Mar-80        CREF  .SAV     6P 01-Mar-80
LINK  .SAV    41P 01-Mar-80        LIBR  .SAV    22P 01-Mar-80
FILEX .SAV    18P 01-Mar-80        SRCCOM.SAV    13P 01-Mar-80
BINCOM.SAV    10P 01-Mar-80        DUMP  .SAV     8P 01-Mar-80
STARTS.COM     1P 01-Mar-80        STARTF.COM     1P 01-Mar-80
V4USER.TXT     1P 01-Mar-80
 53 Files, 940 Blocks
 34 Free blocks
```

Diskette 2

```
PATCH .SAV    10P 01-Mar-80        PAT   .SAV     8P 01-Mar-80
HELP  .SAV   107P 01-Mar-80        HELP  .EXE     7P 01-Mar-80
HELP  .MLB    98P 01-Mar-80        ERROUT.SAV    17P 01-Mar-80
SYSGEN.SAV    39P 01-Mar-80        SYSGEN.CND   134P 01-Mar-80
SYSTBL.CND    23P 01-Mar-80        BATCH .SAV    26P 01-Mar-80
QUEMAN.SAV    13P 01-Mar-80        QUEUE .REL    14P 01-Mar-80
SYSLIB.OBJ    47P 01-Mar-80        MDUP  .SAV    18P 01-Mar-80
MDUP  .MM     48P 01-Mar-80        MDUP  .MT     48P 01-Mar-80
MDUP  .MS     48P 01-Mar-80        MBOOT .BOT     1P 01-Mar-80
MSBOOT.BOT     3P 01-Mar-80        MBOT16.BOT     1P 01-Mar-80
DEMOBG.MAC     2P 01-Mar-80        DEMOFG.MAC     3P 01-Mar-80
DEMOX1.MAC     3P 01-Mar-80        DEMOF1.FOR     2P 01-Mar-80
DEMOED.TXT     1P 01-Mar-80        README.TXT    36P 01-Mar-80
VT52  .TEC     5P 01-Mar-80        VEG   .TEC     4P 01-Mar-80
EDIT  .TEC     1P 01-Mar-80        INSERT.TEC     2P 01-Mar-80
LOCAL .TEC     3P 01-Mar-80        SORT  .TEC     3P 01-Mar-80
VTEDIT.TEC    32P 01-Mar-80        TECO  .TC     23P 01-Mar-80
SEARCH.TEC     3P 01-Mar-80        SQU   .TEC    13P 01-Mar-80
TYPE  .TEC    12P 01-Mar-80        TECO  .INI    15P 01-Mar-80
ODT   .OBJ     9P 01-Mar-80        VDT   .OBJ     9P 01-Mar-80
VTMAC .MAC     7P 01-Mar-80        VTHDLR.OBJ     9P 01-Mar-80
PLOT55.OBJ     3P 01-Mar-80        TEST55.FOR     5P 01-Mar-80
GETSTR.FOR     2P 01-Mar-80        PUTSTR.FOR     2P 01-Mar-80
SYSMAC.MAC    41P 01-Mar-80
 47 Files, 960 Blocks
 14 Free blocks
```

Diskette 3

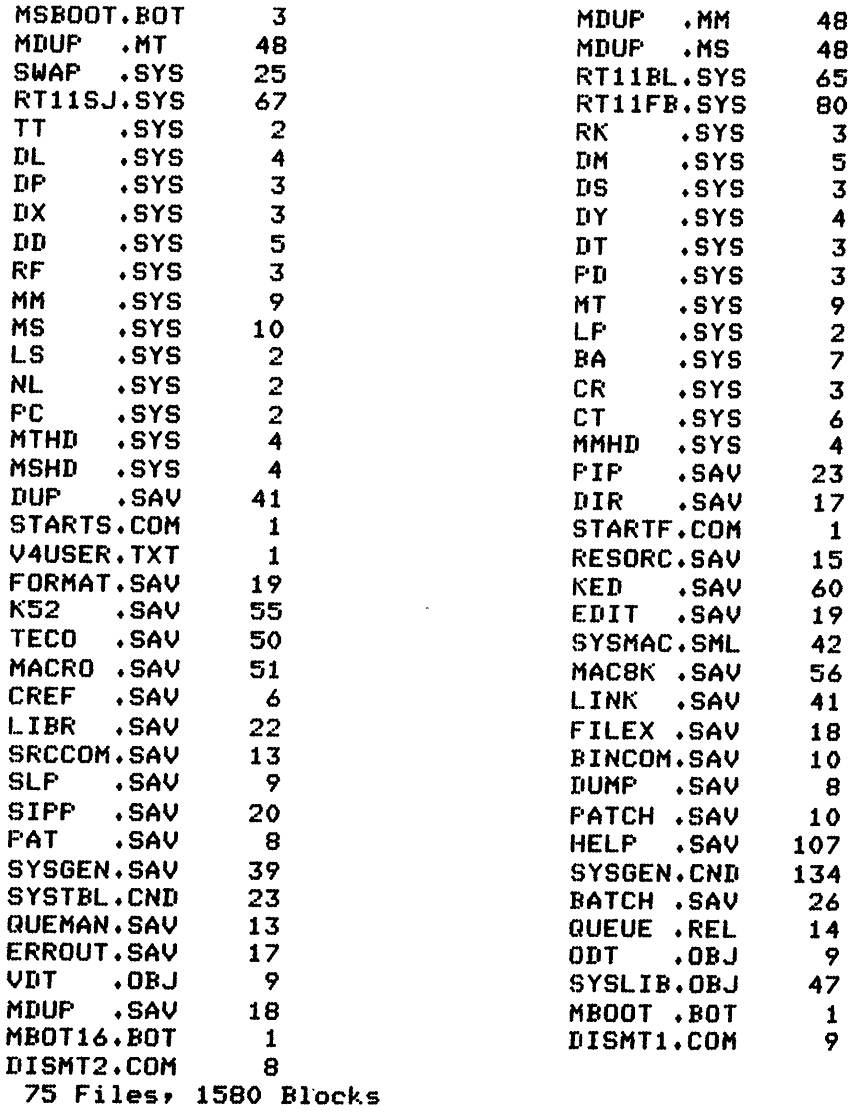
```
SJ    .MAC     1P 01-Mar-80        FB    .MAC     1P 01-Mar-80
XM    .MAC     1P 01-Mar-80        USR   .MAC    69P 01-Mar-80
KMON  .MAC   119P 01-Mar-80        RMONSJ.MAC    76P 01-Mar-80
RMONFB.MAC   138P 01-Mar-80        XMSUBS.MAC    27P 01-Mar-80
EDTGBL.MAC    26P 01-Mar-80        BSTRAP.MAC    48P 01-Mar-80
TT    .MAC     9P 01-Mar-80        BA    .MAC    20P 01-Mar-80
KMOVLY.MAC   185P 01-Mar-80        MTTEMT.MAC    18P 01-Mar-80
MTTINT.MAC    45P 01-Mar-80        LP    .MAC     9P 01-Mar-80
LS    .MAC    10P 01-Mar-80        NL    .MAC     3P 01-Mar-80
DD    .MAC    22P 01-Mar-80        DL    .MAC    21P 01-Mar-80
RK    .MAC     9P 01-Mar-80        DS    .MAC     9P 01-Mar-80
DX    .MAC    15P 01-Mar-80        DY    .MAC    18P 01-Mar-80
PC    .MAC     5P 01-Mar-80        PD    .MAC     9P 01-Mar-80
DM    .MAC    20P 01-Mar-80
 27 Files, 933 Blocks
 41 Free blocks
```

Diskette 4

```
DP    .MAC    10P 01-Mar-80        DT    .MAC     8P 01-Mar-80
RF    .MAC     7P 01-Mar-80        CR    .MAC    14P 01-Mar-80
CT    .MAC    33P 01-Mar-80        TM    .MAC    25P 01-Mar-80
TJ    .MAC    30P 01-Mar-80        TS    .MAC    29P 01-Mar-80
FSM   .MAC    31P 01-Mar-80        SPEED .SAV     4P 01-Mar-80
ELCOPY.MAC     8P 01-Mar-80        ELINIT.MAC    15P 01-Mar-80
ELTASK.MAC    15P 01-Mar-80        ERROUT.MAC    49P 01-Mar-80
ERRTXT.MAC     9P 01-Mar-80        SYCND .BL      3P 01-Mar-80
SYSTBL.BL      4P 01-Mar-80        SYCND .DIS     5P 01-Mar-80
SYCND .HD      5P 01-Mar-80        SYSTBL.DIS     4P 01-Mar-80
RTBL  .MAP    16P 01-Mar-80        RTSJ  .MAP    16P 01-Mar-80
RTFB  .MAP    23P 01-Mar-80        DISMT1.COM     9P 01-Mar-80
DISMT2.COM     8P 01-Mar-80
 25 Files, 380 Blocks
 594 Free blocks
```

G.5 Magtape Kit Directories

Magtape 1



```
MSBOOT.BOT     3          MDUP  .MM     48
MDUP  .MT     48          MDUP  .MS     48
SWAP  .SYS    25          RT11BL.SYS    65
RT11SJ.SYS    67          RT11FB.SYS    80
TT    .SYS     2          RK    .SYS     3
DL    .SYS     4          DM    .SYS     5
DP    .SYS     3          DS    .SYS     3
DX    .SYS     3          DY    .SYS     4
DD    .SYS     5          DT    .SYS     3
RF    .SYS     3          PD    .SYS     3
MM    .SYS     9          MT    .SYS     9
MS    .SYS    10          LP    .SYS     2
LS    .SYS     2          BA    .SYS     7
NL    .SYS     2          CR    .SYS     3
PC    .SYS     2          CT    .SYS     6
MTHD  .SYS     4          MMHD  .SYS     4
MSHD  .SYS     4          PIP   .SAV    23
DUP   .SAV    41          DIR   .SAV    17
STARTS.COM     1          STARTF.COM     1
V4USER.TXT     1          RESORC.SAV    15
FORMAT.SAV    19          KED   .SAV    60
K52   .SAV    55          EDIT  .SAV    19
TECO  .SAV    50          SYSMAC.SML    42
MACRO .SAV    51          MAC8K .SAV    56
CREF  .SAV     6          LINK  .SAV    41
LIBR  .SAV    22          FILEX .SAV    18
SRCCOM.SAV    13          BINCOM.SAV    10
SLP   .SAV     9          DUMP  .SAV     8
SIPP  .SAV    20          PATCH .SAV    10
PAT   .SAV     8          HELP  .SAV   107
SYSGEN.SAV    39          SYSGEN.CND   134
SYSTBL.CND    23          BATCH .SAV    26
QUEMAN.SAV    13          QUEUE .REL    14
ERROUT.SAV    17          ODT   .OBJ     9
VDT   .OBJ     9          SYSLIB.OBJ    47
MDUP  .SAV    18          MBOOT .BOT     1
MBOT16.BOT     1          DISMT1.COM     9
DISMT2.COM     8
 75 Files, 1580 Blocks
```

Magtape 2

```
SYSMAC.MAC    41            VTMAC .MAC     7
VTHDLR.OBJ     9            PLOT55.OBJ     3
TEST55.FOR     5            GETSTR.FOR     2
PUTSTR.FOR     2            SJ    .MAC     1
FB    .MAC     1            XM    .MAC     1
USR   .MAC    69            KMON  .MAC   119
RMONSJ.MAC    76            RMONFB.MAC   138
XMSUBS.MAC    27            EDTGBL.MAC    26
BSTRAP.MAC    48            TT    .MAC     9
BA    .MAC    20            KMOVLY.MAC   185
MTTEMT.MAC    18            MTTINT.MAC    45
LP    .MAC     9            LS    .MAC    10
DD    .MAC    22            DL    .MAC    21
RK    .MAC     9            DT    .MAC     8
DS    .MAC     9            PC    .MAC     5
PD    .MAC     9            RF    .MAC     7
DM    .MAC    20            DX    .MAC    15
DY    .MAC    18            CR    .MAC    14
DP    .MAC    10            NL    .MAC     3
CT    .MAC    33            TM    .MAC    25
TJ    .MAC    30            TS    .MAC    29
FSM   .MAC    31            SPEED .SAV     4
ELCOPY.MAC     8            ELINIT.MAC    15
ELTASK.MAC    15            ERROUT.MAC    49
ERRTXT.MAC     9            HELP  .EXE     7
HELP  .MLB    98            DEMOBG.MAC     2
DEMOFG.MAC     3            DEMOX1.MAC     3
DEMOF1.FOR     2            DEMOED.TXT     1
SYCND .BL      3            SYSTBL.BL      4
SYCND .DIS     5            SYCND .HD      5
SYSTBL.DIS     4            README.TXT    36
VT52  .TEC     5            VEG   .TEC     4
EDIT  .TEC     1            INSERT.TEC     2
LOCAL .TEC     3            SORT  .TEC     3
VTEDIT.TEC    32            TECO  .TC     23
SEARCH.TEC     3            SQU   .TEC    13
TYPE  .TEC    12            TECO  .INI    15
RTBL  .MAP    16            RTSJ  .MAP    16
RTFB  .MAP    23
 77 Files, 1633 Blocks
```

Appendix H
Link Maps for Standard (Distributed) Monitors

H.1 Link Map for Base-Line Single-Job Monitor

```
RT-11 LINK  V06.01       Load Map       Thu 21-Feb-80 01:55:11
RT11BL.SYS       Title:  BSTRAP  Ident:  V04.00

Section  Addr   Size    Global  Value   Global  Value   Global  Value   Global  Value   Global  Value   Global  Value

 . ABS.  000000 006000   (RW,I,GBL,ABS,OVR)
                        PATLVL  000000  F.BLNK  000000  C.CSW   000000  C.HOT   000000  TT.NUM  000000  HDERR$  000001
                        ELUSR   000001  $$RESD  000001  ELKMON  000001  C.LOT   000002  C.SBLK  000002  F.BADR  000002
                        ELEDTG  000002  Q.CSW   000002  $$PIP   000002  $$DIR   000003  C.LENG  000004  C.LINK  000004
                        Q.BLKN  000004  ELBSTR  000004  F.BR5   000004  ELKMOV  000004  $$DUP   000004  $$LINK  000005
                        C.JNUM  000006  C.USED  000006  F.BR4   000006  BOOTSZ  000006  Q.FUNC  000006  ELRMSJ  000006
                        $$BASI  000006  Q.JNUM  000007  Q.UNIT  000007  $$FORT  000007  Q.BUFF  000010  C.DEVQ  000010
                        C.SEQ   000010  USRSZ   000010  $$MACR  000010  C.UNIT  000011  $$DUMP  000011  $SLOT   000011
                        CHSIZ   000012  Q.WCNT  000012  C.SYS   000012  $$LIBR  000012  ..CSI   000013  $$SRCC  000013
                        C.COMP  000014  Q.COMP  000014  $$FILE  000014  $$F4P   000015  $$DICO  000016  $$FOCA  000017
                        KMONSZ  000017  $$APL   000020  ..GTIM  000021  $$FORM  000021  $$BINC  000022  $$ERRO  000023
                        $$QUEM  000024  $$KED   000025  $$K52   000026  $$PASC  000027  $$EDIT  000030  SWAPSZ  000031
                        $$TECO  000031  ..SDTM  000040  V.TKB   000060  V.TPS   000064  OVSZ10  000072  INDX$M  000076
                        RENAM$  000100  DWRIT$  000200  OVSZ12  000372  MAXOFF  000416  OVSZ32  000670  OVSZ2   000702
                        B$BOOT  001000  BHALT   001030  ..28KW  001060  OVSZ14  001076  FXDEVN  001100  CFIG    001126
                        ..NOVT  001176  CFIG2   001204  OVSZ4   001222  OVSZ0   001616  OVSZ30  001626  OVSZ24  001672
                        OVSZ6   001676  OVSZ26  001706  OVSZRS  001710  OVSZ16  001736  OVSZ22  001762  OVSZ20  001774
                        SPFUN$  002000  ..INSA  002606  ..NIND  003050  ..SLNT  003056  SILNT$  003056  ..TTQU  003106
                        BREAD1  003276  BCNFG2  003374  BCNFG   003376  TSLIST  003400  SWPSIZ  003426  SWPBLK  003430
                        AREA    003432  $USIZ$  003740  HNDLR$  004000  USRLEN  004000  B$DEVN  004716  B$DEVS  004720
                        B$DEVU  004722  B$FNAM  004724  B$READ  004730  SYHTOP  004732  DUPFLG  004734  $RMSIZ  004736
                        BSTRNG  004740  ..PLVL  004761  SUFFX   004774  ..SUFF  004774  RELLST  005000  VECLST  005076
                        LODREL  005134  ..TFIL  005162  STFIL   005210  HNDLST  005314  RELEND  005454  KMLEN   007400
                        SPECL$  010000  USRSIZ  010000  KMSIZE  017000  DBLK$M  017400  WONLY$  020000  EOF$    020000
                        RONLY$  040000  FILST$  100000  ACTIV$  100000  VT05$   100000  TKS     177560  TKB     177562
                        TPS     177564  TPB     177566
RT11     006000 017000   (RW,I,LCL,REL,CON)
                        RT11    006000  KMON    006000  ..USRL  006236  .KMLOC  006240  .$USRL  006242  .$KMLO  006244
                        ..SYSL  006246  .MONLC  006250  MEXIT2  006416  .$MONB  010160  .$SWPB  011006  ..ATFX  011230
                        ..DDEV  011470  .EXTFL  011736  .IFSVS  012032  .CORPT  012150  .IFMXN  012274  ..TTOB  012766
                        ..ZAP   012774  ..$CSW  013020  ..QSIZ  013074  ..TTKS  013120  ..CDEV  020264  KPATCH  022450
                        KMSTK   022470  KMSTKE  022660  KMSTAK  023000  OPTX1   023774
RMNUSR   025000 010000   (RW,I,LCL,REL,CON)
                        USRBUF  025000  RMNUSR  025000  LOCATE  025006  LOCAT1  025022  .MNLOC  025054  .USRLO  025070
                        CDFNWT  026032  RSTTTW  026132  .TTOBU  026134  .ZAP    026142  .$CSW   026166  RSTIOW  026240
                        .QSIZE  026242  .TTKS   026266  ..$MTP  026346  USRST   027000  .RECUR  027004  .MONOU  027242
                        .FNAME  027436  .MAXBL  027550  .USRTO  027650  .USRBO  027702  .$DATE  027762  .DFLG   030024
                        .M.USR  030036  .D.USR  030042  .$STAT  030252  .SPUSR  030662  .SPSIZ  030720  .$DVRE  031006
                        .SYSGE  031032  .FNAM6  031112  .$SYS   031126  .DRVTB  031204  .SYSLO  031226  .USRIO  031246
                        .CHKEY  031700  .BLKEY  031732  .USRBU  031744  .I.CNU  032052  .I.CSW  032100  LK4DEV  032302
                        .$JNAM  032310  .$PNAM  032334  .SYIND  032410  .$ENTR  032414  .STKSV  032436  .STATW  032524
                        .INBFP  032602  EMTMOD  033250  .FILDE  033256  .SVSTK  034012  .HANSP  034032  .DVSTS  034070
                        GETFD   034134  DEV1    034174  GETNAM  034226  OCTNUM  034362  DECNUM  034364  CVTNUM  034366
                        DEVSTS  034506
RTDATA   035000 000515   (RW,I,LCL,REL,CON)
                        RTDATA  035000  $RMON   035000  $CSW    035004  SB17    035234  $SYSCH  035244  BLKEY   035256
                        CHKEY   035260  $DATE   035262  DFLG    035264  $USRLC  035266  QCOMP   035270  SYUNIT  035274
                        CONFIG  035300  SCROLL  035302  TTKS    035304  TTKB    035306  TTPS    035310  TTPB    035312
                        $TIME   035320  SYNCH   035324  LOWMAP  035326  USRLOC  035352  GTVECT  035354  $MTPS   035360
                        $MFPS   035362  SYINDX  035364  STATWD  035366  CONFG2  035370  SYSGEN  035372  ERRLEV  035376
                        IFMXNS  035377  PNPTR   035404  MONAME  035406  HSUFFX  035412  DECNET  035414  PROGDF  035416
                        WILDEF  035420  GETPSW  035422  PUTPSW  035442  SYSLOW  035454  CORPTR  035456  CCB     035462
                        I.CSW   035504  IFSVST  035512  SYDVSZ  035514
OWNER$   035516 000000   (RW,I,LCL,REL,CON)
                        $OWNER  035515
UNAM1$   035516 000026   (RW,I,LCL,REL,CON)
                        $UNAM1  035516  DKASSG  035540  SYASSG  035542
UNAM2$   035544 000026   (RW,I,LCL,REL,CON)
                        $UNAM2  035544
PNAME$   035572 000022   (RW,I,LCL,REL,CON)
                        $PNAME  035572
ENTRY$   035614 000024   (RW,I,LCL,REL,CON)
                        TTENTR  035614  $ENTRY  035614
STAT$    035640 000022   (RW,I,LCL,REL,CON)
                        $STAT   035640
DVREC$   035662 000022   (RW,I,LCL,REL,CON)
                        $DVREC  035662
HSIZE$   035704 000022   (RW,I,LCL,REL,CON)
                        $HSIZE  035704
DVSIZ$   035726 000022   (RW,I,LCL,REL,CON)
                        $DVSIZ  035726
RMON     035750 005644   (RW,I,LCL,REL,CON)
                        TTCNFG  035750  EMTPRO  035752  TRAPLC  036454  TRAP10  036476  TRAP4   036476  TRAPER  036540
                        LST16   036564  T$TIN   037050  PTIBF6  037074  EMTDON  037166  KMLOC   037242  TTIINT  037420
                        PTIBF4  037426  PTTPRE  037450  ..BS    037556  SCOP$1  037614  TTIPRE  037742  I.SCCA  040022
                        EXTFLG  040044  SCOP$2  040160  SCOP$3  040164  ZAP     040220  PAVAIL  040306  SCNOP   040334
                        TTILCT  040372  TTIBUF  040374  TTOBUF  040406  TTOCLO  040416  TTORUB  040726  TTSCRL  040752
                        PTOBF2  040760  TTOINT  041024  PTOBF4  041030  $INTEN  041256  LKINT   041276  WAITDV  041742
                        $MONBL  042046  PMONLC  042072  MONCAL  042124  KMBLK   042176  MONLOC  042210  RDOVLY  042220
                        ENTRPG  042224  KMONIN  042232  ECHO    042236  P$SYSC  042400  SYENTR  042414  QSIZE   042440
                        AVAIL   042444  PQCNT   042470  QEWAIT  042472  WAITIO  042670  PI.CSW  043024  P$SWPB  043076
                        $SWPBL  043166  $KMLOC  043176  COMPLT  043204  PSWLST  043330  IOEHLT  043410  $FORK   043412
                        SPTR    043614
PATCH$   043614 001164   (RW,I,LCL,REL,CON)
                        $RMEND  043614
OVLY0    045000 002000   (RW,I,GBL,REL,OVR)
                        $RTEND  045000  CLOSE   045000  HELP    045004  AT2     045010  DATE    045014  TIME    045344
                        ..MNTR  045754  ..HELP  046456  ..ATDK  046550
OVLY2    047000 002000   (RW,I,GBL,REL,OVR)
                        UNLOAD  047006  LOAD    047042  OL1     047102
OVLY4    051000 002000   (RW,I,GBL,REL,OVP)
                        SET     051000
OVLY6    053000 002000   (RW,I,GBL,REL,OVR)
                        B       053000  D       053004  E       053010  INSTAL  053022  REMOVE  053632  ..EMON  054532
OVLY10   055000 000000   (RW,I,GBL,REL,OVR)
OVLYE    055000 002000   (RW,I,GBL,REL,OVR)
                        ERRCOD  055000
OVLY12   057000 000000   (RW,I,GBL,REL,OVR)
OVLYE2   057000 002000   (RW,I,GBL,REL,OVR)
                        RTERR   057000  BRTERR  057034
OVLY14   061000 002000   (RW,I,GBL,REL,OVR)
                        ASSIGN  061000  DEASSI  061004  SQUEEZ  061056  FORMAT  061176  BOOT    061266  CREATE  061362
                        ..ASSY  061550  OPTX2   061766
OVLY16   063000 002000   (RW,I,GBL,REL,OVR)
                        DIFFER  063000  DUMP    063120  TECO    063146  MAKE    063146  MUNG    063146  ..TECN  063212
                        EDIT    063310  ..EDDV  063366  ..EDIF  063654  ..TECF  063662  OPTX3   064434
OVLY20   065000 002000   (RW,I,GBL,REL,OVR)
                        DELETE  065000  RENAME  065140  PRINT   065332  TYPE    065602  INITIA  065734  ..QULP  066316
                        OPTX4   066536
OVLY22   067000 002000   (RW,I,GBL,REL,OVR)
                        FORTRA  067000  COMPIL  067004  BASIC   067010  FOCAL   067026  APL     067044  PASCAL  067062
                        MACRO   067232  DIBOL   067434  OPTX5   070432
OVLY24   071000 002000   (RW,I,GBL,REL,OVR)
                        EXECUT  071372  OPTX6   072264
OVLY26   073000 002000   (RW,I,GBL,REL,OVR)
                        REENTE  073000  START   073004  STRE    073010  LINK    073232  LIBRAR  073740  OPTX7   074356
OVLY30   075000 002000   (RW,I,GBL,REL,OVR)
                        GET1    075000  SAV1    075424  SHOW    075504  OPTX10  075576  OPTX11  076030
OVLY32   077000 002000   (RW,I,GBL,REL,OVR)
                        INDF1   077000  SCNO    077404  AT4     077506
OVLY34   101000 000000   (RW,I,GBL,REL,OVR)

Transfer address = 000001, High limit = 100776 = 16639. words
```

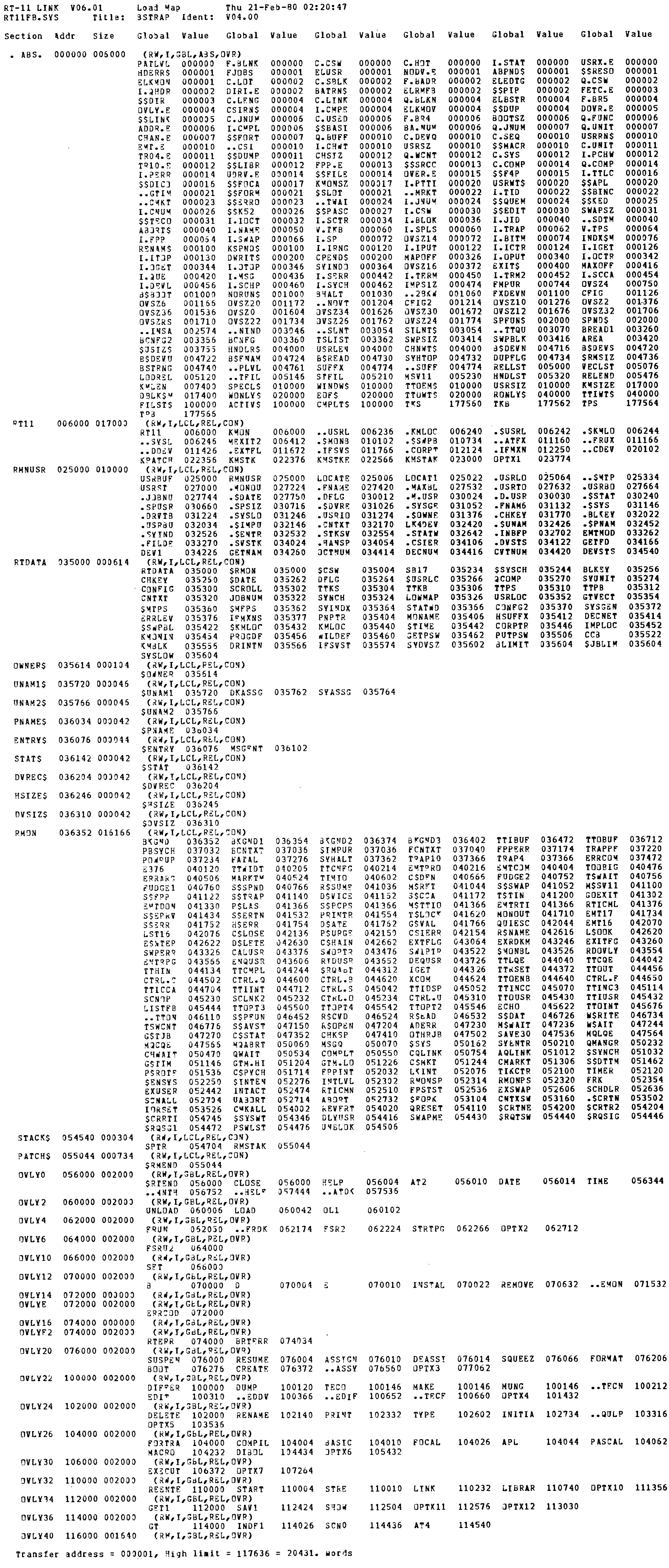
```
RT-11 LINK  V06.01      Load Map       Thu 21-Feb-80 01:54:59
RT11SJ.SYS       Title:  BSTRAP  Ident:  V04.00

Section  Addr   Size    Global  Value   Global  Value   Global  Value   Global  Value   Global  Value   Global  Value

 . ABS.  000000 005000   (RW,I,GBL,ABS,OVR)
                        PATLVL  000000  F.BLNK  000000  C.CSW   000000  C.HOT   000000  TT.NUM  000000  HDERR$  000001
                        ELUSR   000001  $$RFSO  000001  ELKMON  000001  C.LOT   000002  C.SBLK  000002  F.BADR  000002
                        ELEDTG  000002  Q.CSW   000002  $$PIP   000002  $$DIR   000003  C.LENG  000004  C.LINK  000004
                        Q.BLKN  000004  ELBSTR  000004  F.BR5   000004  ELKMOV  000004  $$DUP   000004  BA.NUM  000004
                        $$LINK  000005  C.JNUM  000006  C.USED  000006  F.BR4   000006  BOOTSZ  000006  Q.FUNC  000006
                        ELRMSJ  000006  $$BASI  000006  Q.JNUM  000007  Q.UNIT  000007  $$FORT  000007  Q.BUFF  000010
                        C.DEVQ  000010  C.SEQ   000010  USRSZ   000010  $$MACR  000010  C.UNIT  000011  $$DUMP  000011
                        CHSIZ   000012  Q.WCNT  000012  C.SYS   000012  $$LIBP  000012  ..CSI   000013  $$SRCC  000013
                        C.COMP  000014  Q.COMP  000014  $$FILE  000014  $$F4P   000015  $$DICO  000016  $$FOCA  000017
                        KMONSZ  000017  $$APL   000020  $SLOT   000020  ..GTIM  000021  $$FORM  000021  $$BINC  000022
                        $$ERRO  000023  $$QUEM  000024  $$KED   000025  $$K52   000026  $$PASC  000027  $$EDIT  000030
                        SWAPSZ  000031  $$TECO  000031  ..SDTM  000040  V.TKB   000060  V.TPS   000064  OVSZ10  000072
                        INDX$M  000076  RENAM$  000100  DWRIT$  000200  OVSZ12  000372  MAXOFF  000416  OVSZ2   000702
                        B$BOOT  001000  BHALT   001030  ..28KW  001060  OVSZ14  001076  FXDEVN  001100  CFIG    001126
                        ..NOVT  001204  CFIG2   001214  OVSZ4   001222  OVSZ32  001476  OVSZ0   001616  OVSZ30  001626
                        OVSZ24  001672  OVSZ6   001676  OVSZ26  001706  OVSZRS  001710  OVSZ16  001736  OVSZ22  001762
                        OVSZ20  001774  SPFUN$  002000  ..INSA  002616  ..NIND  003070  ..SLNT  003076  SILNT$  003076
                        ..ITQU  003126  BREAD1  003316  BCNFG2  003414  BCNFG   003416  TSLIST  003420  SWPSIZ  003452
                        SWPBLK  003454  AREA    003456  $USIZ$  003750  HNDLR$  004000  USRLEN  004000  B$DEVN  004716
                        B$DEVS  004720  B$DEVU  004722  B$FNAM  004724  B$READ  004730  SYHTOP  004732  DUPFLG  004734
                        $RMSIZ  004736  BSTRNG  004740  ..PLVL  004761  SUFFX   004774  ..SUFF  004774  RELLST  005000
                        VECLST  005104  LODREL  005125  ..TFIL  005154  STFIL   005216  HNDLST  005322  RELEND  005462
                        KMLEN   007400  SPECL$  010000  USRSIZ  010000  KMSIZE  017000  DBLK$M  017400  WONLY$  020000
                        EOF$    020000  RONLY$  040000  FILST$  100000  ACTIV$  100000  VT05$   100000  TKS     177560
                        TKB     177562  TPS     177564  TPB     177566
 RT11    006000 017000   (RW,I,LCL,REL,CON)
                        RT11    006000  KMON    006000  ..USRL  006236  .KMLOC  006240  .$USRL  006242  .$KMLO  006244
                        ..SYSL  006245  .MONLC  006250  MEXIT2  006415  .$MONB  010160  .$SWPB  011006  ..ATFX  011230
                        ..DDEV  011470  .EXTFL  011736  .IFSVS  012032  .CORPT  012170  .IFMXN  012314  ..TTOB  013006
                        ..ZAP   013014  ..$CSW  013040  ..QSIZ  013114  ..TTKS  013140  ..CDEV  020304  KPATCH  022470
                        KMSTK   022510  KMSTKE  022700  KMSTAK  023000  OPTX1   023774
 RMNUSR  025000 010000   (RW,I,LCL,REL,CON)
                        USRBUF  025000  RMNUSR  025000  LOCATE  025006  LOCAT1  025022  .MNLOC  025054  .USRLO  025070
                        CDFNWT  026044  RSTTTW  026144  .TTOBU  026146  .ZAP    026154  .$CSW   026200  RSTIOW  026252
                        .QSIZE  026254  .TTKS   026300  ..$MTP  026360  USRST   027000  .RECUR  027004  .MONOU  027242
                        .FNAME  027436  .MAXBL  027550  .USRTO  027650  .USRBO  027702  .$DATE  027762  .DFLG   030024
                        .M.USR  030036  .D.USR  030042  .$STAT  030252  .SPUSR  030662  .SPSIZ  030720  .$DVRE  031006
                        .SYSGE  031032  .FNAM6  031112  .$SYS   031126  .DRVTB  031204  .SYSLO  031226  .USRIO  031246
                        .CHKEY  031700  .BLKEY  031732  .USRBU  031744  .I.CNU  032052  .I.CSW  032100  LK4DEV  032302
                        .SUNAM  032310  .SPNAM  032334  .SYIND  032410  .SENTR  032414  .STKSV  032436  .STATW  032524
                        .INBFP  032602  EMTMOD  033270  .FILDS  033276  .SVSTK  034032  .HANSP  034052  .DVSTS  034110
                        GETFD   034154  DEV1    034214  GETNAM  034246  OCTNUM  034402  DECNUM  034404  CVTNUM  034406
                        DEVSTS  034526
 RTDATA  035000 000516   (RW,I,LCL,REL,CON)
                        RTDATA  035000  $RMON   035000  $CSW    035004  SB17    035234  $SYSCH  035244  BLKEY   035256
                        CHKEY   035260  $DATE   035262  DFLG    035264  $USRLC  035266  QCOMP   035270  SYUNIT  035274
                        CONFIG  035300  SCROLL  035302  TTKS    035304  TTKB    035306  TTPS    035310  TTPB    035312
                        $TIME   035320  SYNCH   035324  LOWMAP  035326  USRLOC  035352  GTVECT  035354  $MTPS   035360
                        $MFPS   035362  SYINDX  035364  STATWD  035366  CONFG2  035370  SYSGEN  035372  ERRLEV  035376
                        IFMXNS  035377  PNPTR   035404  MONAME  035406  HSUFFX  035412  DECNET  035414  PROGDF  035416
                        WILDEF  035420  GETPSW  035422  PUTPSW  035442  SYSLOW  035454  CORPTR  035456  CCB     035462
                        I.CSW   035504  IFSVST  035512  SYDVSZ  035514
 OWNER$  035516 000000   (RW,I,LCL,REL,CON)
                        $OWNER  035516
 UNAM1$  035516 000044   (RW,I,LCL,REL,CON)
                        $UNAM1  035516  DKASSG  035556  SYASSG  035560
 UNAM2$  035562 000044   (RW,I,LCL,REL,CON)
                        $UNAM2  035562
 PNAME$  035626 000040   (RW,I,LCL,REL,CON)
                        $PNAME  035626
 ENTRY$  035666 000042   (RW,I,LCL,REL,CON)
                        TTENTR  035666  $ENTRY  035666
 STAT$   035730 000040   (RW,I,LCL,REL,CON)
                        $STAT   035730
 DVREC$  035770 000040   (RW,I,LCL,REL,CON)
                        $DVREC  035770
 HSIZE$  036030 000040   (RW,I,LCL,REL,CON)
                        $HSIZE  036030
 DVSIZ$  036070 000040   (RW,I,LCL,REL,CON)
                        $DVSIZ  036070
 RMON    036130 006112   (RW,I,LCL,REL,CON)
                        TTCNFG  036130  EMTPRO  036132  TRAPLC  036634  TRAP10  036556  TRAP4   036656  TRAPER  036720
                        FPPADD  036746  FPPINT  036754  FPPIGN  037004  TRAPPF  037030  LST16   037124  T$TIN   037410
                        PTIBF6  037434  EMTDON  037526  KMLOC   037602  TTIINT  037760  PTIBF4  037766  PTTPRE  040014
                        ..BS    040122  SCOP$1  040160  SCLNK2  040306  TTIPRE  040310  I.SCCA  040370  EXTFLG  040424
                        SCOP$2  040540  SCOP$3  040544  ZAP     040600  PAVAIL  040666  SCNOP   040714  TTILCT  040752
                        TTIBUF  040754  TTOBUF  040766  TTOCLO  040776  TTORUB  041306  TTSCRL  041332  PTOBF2  041340
                        TTOINT  041404  PTOBF4  041410  $INTEN  041636  LKINT   041656  WAITDV  042322  $MONBL  042426
                        PMONLC  042452  MONCAL  042554  KMBLK   042626  MONLOC  042640  RDOVLY  042646  ENTRPG  042652
                        KMONIN  042660  ECHO    042664  P$SYSC  043026  SYENTR  043042  QSIZE   043066  AVAIL   043072
                        PQCNT   043116  QEWAIT  043120  WAITIO  043316  PI.CSW  043452  P$SWPB  043524  $SWPBL  043614
                        $KMLOC  043624  COMPLT  043632  PSWLST  043756  IOEHLT  044036  $FORK   044040  SPTR    044242
 PATCH$  044242 000536   (RW,I,LCL,REL,CON)
                        $RMEND  044242
 OVLY0   045000 002000   (RW,I,GBL,REL,OVR)
                        $RTEND  045000  CLOSE   045000  HELP    045004  AT2     045010  DATE    045014  TIME    045344
                        ..MNTH  045754  ..HELF  046456  ..ATDK  046550
 OVLY2   047000 002000   (RW,I,GBL,REL,OVR)
                        UNLOAD  047005  LOAD    047042  OL1     047102
 OVLY4   051000 002000   (RW,I,GBL,REL,OVR)
                        SET     051000
 OVLY6   053000 002000   (RW,I,GBL,REL,OVR)
                        B       053000  D       053004  E       053010  INSTAL  053022  REMOVE  053632  ..EMON  054532
 OVLY10  055000 000000   (RW,I,GBL,REL,OVR)
 OVLYE   055000 002000   (RW,I,GBL,REL,OVR)
                        ERRCOD  055000
 OVLY12  057000 000000   (RW,I,GBL,REL,OVR)
 OVLYE2  057000 002000   (RW,I,GBL,REL,OVR)
                        RTERR   057000  BRTERR  057034
 OVLY14  061000 002000   (RW,I,GBL,REL,OVR)
                        ASSIGN  061000  DEASSI  061004  SQUEEZ  061056  FORMAT  061176  BOOT    061266  CREATE  061362
                        ..ASSY  061550  OPTX2   061766
 OVLY16  063000 002000   (RW,I,GBL,REL,OVR)
                        DIFFER  063000  DUMP    063120  TECO    063146  MAKE    063146  MUNG    063146  ..TECN  063212
                        EDIT    063310  ..EDDV  063366  ..EDIF  063654  ..TECF  063662  OPTX3   064434
 OVLY20  065000 002000   (RW,I,GBL,REL,OVR)
                        DELETE  065000  RENAME  065140  PRINT   065332  TYPE    065602  INITIA  065734  ..QULP  066316
                        OPTX4   066536
 OVLY22  067000 002000   (RW,I,GBL,REL,OVR)
                        FORTRA  067000  COMPIL  067004  BASIC   067010  FOCAL   067026  APL     067044  PASCAL  067062
                        MACRO   067232  DIBOL   067434  OPTX5   070432
 OVLY24  071000 002000   (RW,I,GBL,REL,OVR)
                        EXECUT  071372  OPTX6   072264
 OVLY26  073000 002000   (RW,I,GBL,REL,OVR)
                        REENTE  073000  START   073004  STRE    073010  LINK    073232  LIBRAR  073740  OPTX7   074356
 OVLY30  075000 002000   (RW,I,GBL,REL,OVR)
                        GET1    075000  SAV1    075424  SHOW    075504  OPTX10  075576  OPTX11  076030
 OVLY32  077000 002000   (RW,I,GBL,REL,OVR)
                        GT      077000  INDF1   077025  SCNO    077436  AT4     077540
 OVLY34  101000 001756   (RW,I,GBL,REL,OVR)

Transfer address = 000001, High limit = 102754 = 17142. words
```

H.3 Link Map for Foreground/Background Monitor



```
RT-11 LINK  V06.01      Load Map       Thu 21-Feb-80 02:20:47
RT11FB.SYS       Title:  BSTRAP  Ident:  V04.00

Section  Addr   Size    Global  Value   Global  Value   Global  Value   Global  Value   Global  Value   Global  Value

 . ABS.  000000 006000   (RW,I,GBL,ABS,OVR)
                        PATLVL  000000  F.BLNK  000000  C.CSW   000000  C.HOT   000000  I.STAT  000000  USRX.E  000000
                        HDERR$  000001  FJOBS   000001  ELUSR   000001  NODV.E  000001  ABPND$  000001  $$RESO  000001
                        ELKMON  000001  C.LOT   000002  C.SBLK  000002  F.BADR  000002  ELEDTG  000002  Q.CSW   000002
                        I.QHDR  000002  DIRI.E  000002  BATRN$  000002  ELRMFB  000002  $$PIP   000002  FETC.E  000003
                        $$DIR   000003  C.LENG  000004  C.LINK  000004  Q.BLKN  000004  ELBSTR  000004  F.BR5   000004
                        OVLY.E  000004  CSIRN$  000004  I.CMPE  000004  ELKMOV  000004  $$DUP   000004  DOVR.E  000005
                        $$LINK  000005  C.JNUM  000006  C.USED  000006  F.BR4   000006  BOOTSZ  000006  Q.FUNC  000006
                        ADDR.E  000006  I.CMPL  000006  $$BASI  000006  BA.NUM  000006  Q.JNUM  000007  Q.UNIT  000007
                        CHAN.E  000007  $$FORT  000007  Q.BUFF  000010  C.DEVQ  000010  C.SEQ   000010  USRRN$  000010
                        EMT.E   000010  ..CSI   000010  I.CHWT  000010  USRSZ   000010  $$MACR  000010  C.UNIT  000011
                        TR04.E  000011  $$DUMP  000011  CHSIZ   000012  Q.WCNT  000012  C.SYS   000012  I.PCHW  000012
                        TR10.E  000012  $$LIBR  000012  FPP.E   000013  $$SRCC  000013  C.COMP  000014  Q.COMP  000014
                        I.PERR  000014  UDRV.E  000014  $$FILE  000014  OVER.E  000015  $$F4P   000015  I.TTLC  000016
                        $$DICO  000016  $$FOCA  000017  KMONSZ  000017  I.PTTI  000020  USRWT$  000020  $$APL   000020
                        ..GTIM  000021  $$FORM  000021  $SLOT   000021  ..MRKT  000022  I.TID   000022  $$BINC  000022
                        ..CMKT  000023  $$ERRO  000023  ..TWAI  000024  I.JNUM  000024  $$QUEM  000024  $$KED   000025
                        I.CNUM  000026  $$K52   000026  $$PASC  000027  I.CSW   000030  $$EDIT  000030  SWAPSZ  000031
                        $$TECO  000031  I.IOCT  000032  I.SCTR  000034  I.BLOK  000036  I.JID   000040  ..SDTM  000040
                        ABORT$  000040  I.NAME  000050  V.TKB   000060  I.SPLS  000060  I.TRAP  000062  V.TPS   000064
                        I.FPP   000064  I.SWAP  000066  I.SP    000072  OVSZ14  000072  I.BITM  000074  INDX$M  000076
                        RENAM$  000100  KSPND$  000100  I.IRNG  000120  I.IPUT  000122  I.ICTR  000124  I.IGET  000126
                        I.ITOP  000130  DWRIT$  000200  CPEND$  000200  MAPOFF  000326  I.OPUT  000340  I.OCTR  000342
                        I.OGET  000344  I.OTOP  000346  SYINDO  000364  OVSZ16  000372  EXIT$   000400  MAXOFF  000416
                        I.QUE   000420  I.MSG   000436  I.SERR  000442  I.TERM  000450  I.TRM2  000452  I.SCCA  000454
                        I.DEVL  000456  I.SCHP  000460  I.SYCH  000462  IMPSIZ  000474  FMPUR   000744  OVSZ4   000750
                        B$BOOT  001000  NORUN$  001000  BHALT   001030  ..28KW  001060  FXDEVN  001100  CFIG    001126
                        OVSZ6   001166  OVSZ20  001172  ..NOVT  001204  CFIG2   001214  OVSZ10  001276  OVSZ2   001376
                        OVSZ36  001536  OVSZ0   001604  OVSZ34  001626  OVSZ30  001672  OVSZ12  001676  OVSZ32  001706
                        OVSZRS  001710  OVSZ22  001734  OVSZ26  001762  OVSZ24  001774  SPFUN$  002000  SPND$   002000
                        ..INSA  002574  ..NIND  003046  ..SLNT  003054  SILNT$  003054  ..TTQU  003070  BREAD1  003260
                        BCNFG2  003356  BCNFG   003360  TSLIST  003362  SWPSIZ  003414  SWPBLK  003416  AREA    003420
                        $JSIZ$  003755  HNDLR$  004000  USRLEN  004000  CHNWT$  004000  B$DEVN  004716  B$DEVS  004720
                        B$DEVU  004722  B$FNAM  004724  B$READ  004730  SYHTOP  004732  DUPFLG  004734  $RMSIZ  004736
                        BSTRNG  004740  ..PLVL  004761  SUFFX   004774  ..SUFF  004774  RELLST  005000  VECLST  005076
                        LODREL  005120  ..TFIL  005146  STFIL   005210  MSV11   005230  HNDLST  005320  RELEND  005476
                        KMLEN   007400  SPECL$  010000  WINDW$  010000  TTOEM$  010000  USRSIZ  010000  KMSIZE  017000
                        DBLK$M  017400  WONLY$  020000  EOF$    020000  TTOWT$  020000  RONLY$  040000  TTIWT$  040000
                        FILST$  100000  ACTIV$  100000  CMPLT$  100000  TKS     177560  TKB     177562  TPS     177564
                        TPB     177566
RT11     006000 017000   (RW,I,LCL,REL,CON)
                        RT11    006000  KMON    006000  ..USRL  006236  .KMLOC  006240  .$USRL  006242  .$KMLO  006244
                        ..SYSL  006246  MEXIT2  006412  .$MONB  010102  .$SWPB  010734  ..ATFX  011160  ..FRUX  011166
                        ..DDEV  011426  .EXTFL  011672  .IFSVS  011766  .CORPT  012124  .IFMXN  012250  ..CDEV  020102
                        KPATCH  022356  KMSTK   022376  KMSTKE  022566  KMSTAK  023000  OPTX1   023774
RMNUSR   025000 010000   (RW,I,LCL,REL,CON)
                        USRBUF  025000  RMNUSR  025000  LOCATE  025006  LOCAT1  025022  .USRLO  025064  ..$MTP  025334
                        USRST   027000  .MONOU  027224  .FNAME  027420  .MAXBL  027532  .USRTO  027632  .USRBO  027664
                        .JOBNU  027744  .$DATE  027750  .DFLG   030012  .M.USR  030024  .D.USR  030030  .$STAT  030240
                        .SPUSR  030660  .SPSIZ  030716  .$DVRE  031026  .SYSGE  031052  .FNAM6  031132  .$SYS   031146
                        .DRVTB  031224  .SYSLO  031246  .USRIO  031274  .$OWNE  031376  .CHKEY  031770  .BLKEY  032022
                        .USRBU  032034  .$IMPU  032146  .CNTXT  032170  LK4DEV  032420  .$UNAM  032426  .$PNAM  032452
                        .SYIND  032526  .$ENTR  032532  .STKSV  032554  .STATW  032642  .INBFP  032702  EMTMOD  033262
                        .FILDE  033270  .SVSTK  034024  .HANSP  034054  .CSIER  034106  .DVSTS  034122  GETFD   034166
                        DEV1    034226  GETNAM  034260  OCTNUM  034414  DECNUM  034416  CVTNUM  034420  DEVSTS  034540
RTDATA   035000 000614   (RW,I,LCL,REL,CON)
                        RTDATA  035000  $RMON   035000  $CSW    035004  SB17    035234  $SYSCH  035244  BLKEY   035256
                        CHKEY   035250  $DATE   035262  DFLG    035264  $USRLC  035266  QCOMP   035270  SYUNIT  035274
                        CONFIG  035300  SCROLL  035302  TTKS    035304  TTKB    035306  TTPS    035310  TTPB    035312
                        CNTXT   035320  JOBNUM  035322  SYNCH   035324  LOWMAP  035326  USRLOC  035352  GTVECT  035354
                        $MTPS   035360  $MFPS   035362  SYINDX  035364  STATWD  035366  CONFG2  035370  SYSGEN  035372
                        ERRLEV  035376  IFMXNS  035377  PNPTR   035404  MONAME  035406  HSUFFX  035412  DECNET  035414
                        $SWPBL  035422  $KMLOC  035432  KMLOC   035440  $TIME   035442  CORPTR  035446  IMPLOC  035452
                        KMONIN  035454  PROGDF  035456  WILDEF  035460  GETPSW  035462  PUTPSW  035506  CCB     035522
                        KMBLK   035556  DRINTN  035566  IFSVST  035574  SYDVSZ  035602  BLIMIT  035604  $JBLIM  035604
                        SYSLOW  035604
OWNER$   035614 000104   (RW,I,LCL,REL,CON)
                        $OWNER  035614
UNAM1$   035720 000046   (RW,I,LCL,REL,CON)
                        $UNAM1  035720  DKASSG  035762  SYASSG  035764
UNAM2$   035766 000046   (RW,I,LCL,REL,CON)
                        $UNAM2  035766
PNAME$   036034 000042   (RW,I,LCL,REL,CON)
                        $PNAME  036034
ENTRY$   036076 000044   (RW,I,LCL,REL,CON)
                        $ENTRY  036076  MSGENT  036102
STAT$    036142 000042   (RW,I,LCL,REL,CON)
                        $STAT   036142
DVREC$   036204 000042   (RW,I,LCL,REL,CON)
                        $DVREC  036204
HSIZE$   036246 000042   (RW,I,LCL,REL,CON)
                        $HSIZE  036245
DVSIZ$   036310 000042   (RW,I,LCL,REL,CON)
                        $DVSIZ  036310
RMON     036352 016166   (RW,I,LCL,REL,CON)
                        BKGND   036352  BKGND1  036354  BKGND2  036374  BKGND3  036402  TTIBUF  036472  TTOBUF  036712
                        PBSYCH  037032  BCNTXT  037036  $IMPUR  037036  FCNTXT  037040  FPPERR  037174  TRAPPF  037220
                        POWFUP  037234  FATAL   037276  SYHALT  037362  TRAP10  037366  TRAP4   037366  ERRCOM  037472
                        E376    040120  TTWIDT  040205  TTCNFG  040214  EMTPRO  040216  EMTCOM  040404  TOOBIG  040476
                        ERRARG  040506  MARKTM  040524  TIMIO   040602  C$DFN   040666  FUDGE2  040752  T$WAIT  040756
                        FUDGE1  040760  S$SPND  040766  R$SUME  041036  M$RKT   041044  S$SWAP  041052  M$SV11  041100
                        S$FPP   041122  S$TRAP  041140  D$VICE  041152  S$CCA   041172  T$TIN   041200  GOEXIT  041302
                        EMTDON  041330  P$LAS   041366  S$PCPS  041366  M$TTIO  041366  EMTRTI  041366  RTICML  041376
                        S$EPRV  041434  S$ERTN  041532  PRINTR  041554  T$LOCK  041620  MONOUT  041710  EMT17   041734
                        S$ERR   041752  H$ERR   041754  D$ATE   041762  G$VAL   041766  QUIESC  042044  EMT16   042070
                        LST16   042076  C$LOSE  042136  P$URGE  042150  CSIERR  042154  R$NAME  042616  L$OOK   042620
                        E$NTER  042622  D$LETE  042630  C$HAIN  042662  EXTFLG  043064  EXRDKM  043246  EXITFG  043260
                        SWPERR  043326  CALUSR  043376  SWOPTR  043476  SW1PIP  043522  $MONBL  043526  RDOVLY  043554
                        ENTRPG  043566  ENQUSR  043606  RTDUSR  043652  DEQUSR  043726  TTLQE   044040  TTCQE   044042
                        TTHIN   044134  TTCMPL  044244  $RQABT  044312  IGET    044326  TTRSET  044372  TTOUT   044456
                        CTRL.C  044502  CTRL.Q  044600  CTRL.B  044620  XCOM    044624  TTOENB  044640  CTRL.F  044650
                        TTICCA  044704  TTIINT  044712  CTRL.S  045042  TTIDSP  045052  TTINCC  045070  TTINC3  045114
                        SCNOP   045230  SCLNK2  045232  CTRL.O  045234  CTRL.U  045310  TTOUSR  045430  TTIUSR  045432
                        LISTFB  045444  TTOPT3  045500  TTOPT4  045542  TTOPT2  045546  ECHO    045622  TTOINT  045676
                        ..TTON  046110  S$PFUN  046452  R$CVD   046524  R$EAD   046532  S$DAT   046726  W$RITE  046734
                        TSWCNT  046776  S$AVST  047150  R$OPEN  047204  ADERR   047230  M$WAIT  047236  W$AIT   047244
                        G$TJB   047270  C$STAT  047352  CHKSP   047410  OTHRJB  047502  SAVE30  047536  MQLQE   047564
                        MQCQE   047566  MQABRT  050060  MSGQ    050070  $SYS    050162  SYENTR  050210  QMANGR  050232
                        CHWAIT  050470  QWAIT   050534  COMPLT  050550  CQLINK  050754  AQLINK  051012  $SYNCH  051032
                        G$TIM   051146  GTM.HI  051204  GTM.LO  051226  C$MKT   051244  CMARKT  051306  S$DTTM  051462
                        P$ROTE  051536  C$PYCH  051714  FPPINT  052032  LKINT   052076  TIKCTR  052100  TIMER   052120
                        $ENSYS  052250  $INTEN  052276  INTLVL  052302  RMONSP  052314  RMONPS  052320  FRK     052354
                        EXUSER  052442  INTACT  052474  RTICMN  052510  FPSTST  052536  EXSWAP  052606  SCHDLR  052636
                        SCNALL  052704  UABORT  052714  ABORT   052732  $FORK   053104  CNTXSW  053160  .$CRTN  053502
                        IORSET  053526  CMKALL  054002  REVERT  054020  QRESET  054110  $CRTNE  054200  $CRTR2  054204
                        $CRRTI  054245  $SYSWT  054346  DLYUSR  054416  SWAPME  054430  $RQTSW  054440  $RQSIG  054446
                        $RQSG1  054472  PSWLST  054476  UNBLOK  054506
STACK$   054540 000304   (RW,I,LCL,REL,CON)
                        SPTR    054704  RMSTAK  055044
PATCH$   055044 000734   (RW,I,LCL,REL,CON)
                        $RMEND  055044
OVLY0    056000 002000   (RW,I,GBL,REL,OVR)
                        $RTEND  056000  CLOSE   056000  HELP    056004  AT2     056010  DATE    056014  TIME    056344
                        ..MNTH  056752  ..HELP  057444  ..ATDK  057536
OVLY2    060000 002000   (RW,I,GBL,REL,OVR)
                        UNLOAD  060006  LOAD    060042  OL1     060102
OVLY4    062000 002000   (RW,I,GBL,REL,OVR)
                        FRUN    062050  ..FRDK  062174  FSR2    062224  STRTPG  062266  OPTX2   062712
OVLY6    064000 002000   (RW,I,GBL,REL,OVR)
                        FSRU2   064000
OVLY10   066000 002000   (RW,I,GBL,REL,OVR)
                        SET     066000
OVLY12   070000 002000   (RW,I,GBL,REL,OVR)
                        B       070000  D       070004  E       070010  INSTAL  070022  REMOVE  070632  ..EMON  071532
OVLY14   072000 000000   (RW,I,GBL,REL,OVR)
OVLYE    072000 002000   (RW,I,GBL,REL,OVR)
                        ERRCOD  072000
OVLY16   074000 000000   (RW,I,GBL,REL,OVR)
OVLYF2   074000 002000   (RW,I,GBL,REL,OVR)
                        RTERR   074000  BRTERR  074034
OVLY20   076000 002000   (RW,I,GBL,REL,OVR)
                        SUSPEN  076000  RESUME  076004  ASSIGN  076010  DEASSI  076014  SQUEEZ  076066  FORMAT  076206
                        BOOT    076276  CREATE  076372  ..ASSY  076560  OPTX3   077062
OVLY22   100000 002000   (RW,I,GBL,REL,OVR)
                        DIFFER  100000  DUMP    100120  TECO    100146  MAKE    100146  MUNG    100146  ..TECN  100212
                        EDIT    100310  ..EDDV  100366  ..EDIF  100652  ..TECF  100660  OPTX4   101432
OVLY24   102000 002000   (RW,I,GBL,REL,OVR)
                        DELETE  102000  RENAME  102140  PRINT   102332  TYPE    102602  INITIA  102734  ..QULP  103316
                        OPTX5   103536
OVLY26   104000 002000   (RW,I,GBL,REL,OVR)
                        FORTRA  104000  COMPIL  104004  BASIC   104010  FOCAL   104026  APL     104044  PASCAL  104062
                        MACRO   104232  DIBOL   104434  OPTX6   105432
OVLY30   106000 002000   (RW,I,GBL,REL,OVR)
                        EXECUT  106372  OPTX7   107264
OVLY32   110000 002000   (RW,I,GBL,REL,OVR)
                        REENTE  110000  START   110004  STRE    110010  LINK    110232  LIBRAR  110740  OPTX10  111356
OVLY34   112000 002000   (RW,I,GBL,REL,OVR)
                        GET1    112000  SAV1    112424  SHOW    112504  OPTX11  112576  OPTX12  113030
OVLY36   114000 002000   (RW,I,GBL,REL,OVR)
                        GT      114000  INDF1   114026  SCNO    114436  AT4     114540
OVLY40   116000 001640   (RW,I,GBL,REL,OVR)

Transfer address = 000001, High limit = 117636 = 20431. words
```

Appendix I
Customization Patches for Specially Generated Monitors

NOTE

In the following patches, lower-case alphabetic x represents a character that varies according to the specific software component you are patching.

I.1 Installing Only the Generated Handlers

Normally, when you bootstrap an RT–11 system, the monitor automatically installs any device handlers that are present on the system device if there are enough device slots for them. Under certain circumstances, you may want the system to include only the handlers that you named during your system generation process. To prevent the system from recognizing handlers that you did not specify during your system generation, install the following patch.

NOTE

Do not install this patch on any of the distributed monitors.

In the patch, monitr.SYS is the name of the monitor file that you want to modify and ..INSA is the value of that symbol from the monitor link map.

```
.RUN SIPP(RET)
*monitr.SYS(RET)
Base?     0(RET)
Offset?   ..INSA(RET)
  Base        Offset      Old    New?
   000000     ..INSA    010105   240(RET)
   000000     ..INSA+2  020200   (CTRL/Y)(RET)
* (CTRL/C)

.
```

If the monitor you want to alter is the hardware bootable monitor, write a new system bootstrap with the COPY/BOOT command.

I.2 Suppressing the Multi-terminal Polling Routines

If you selected DZ–11 modem support or the multi-terminal time-out feature when you performed your system generation, the monitor automatically polls the terminal lines twice every second. This can cause ODT to malfunction. If you install the following patch to the monitor, it will not do this periodic polling. This allows you to use ODT, but defeats the value of the multi-terminal time-out feature, and prevents the monitor from recognizing remote DZ–11 or DZV–11 lines. In the patch, monitr.SYS is the name of the monitor file that you want to modify and ..DZOD is the value of that symbol from the monitor link map.

```
.RUN SIPP(RET)
*monitr.SYS(RET)
Base?      0(RET)
Offset?   ..DZOD(RET)
  Base          Offset        Old   New?
   000000       ..DZOD     000000   1(RET)
   000000       ..DZOD+2   001002   (CTRL/Y)(RET)
*(CTRL/C)

.
```

If the monitor you want to alter is the hardware bootable monitor, write a new system bootstrap with the COPY/BOOT command.

I.3 Changing the Handler Filename Suffix

In an ordinary RT–11 SJ or FB system, the monitor recognizes device handlers by the format of their names. Device handler names have the format: dd.SYS. In an XM system, handlers are in files named ddX.SYS. Thus, two sets of handlers can coexist on one system volume.

If you generate one or more monitors with different combinations of the device time-out, error logging, and extended memory features, you can have need of up to eight different sets of device handlers. You can keep such handlers on one system volume by associating one or more monitors with the corresponding handlers. Do this by patching the monitor so that it recognizes device handler file names of the form ddn.SYS, where n is any alphabetic character that you choose. Then rename the corresponding handler files.

Install the following patch in the monitor file. In the patch, monitr.SYS is the name of the monitor file that you want to modify, and n is the one-character suffix that you use when you rename the associated handlers. The old values depend on the system generation options that you chose.

```
.RUN SIPP(RET)
*monitr.SYS(RET)
Base?     0(RET)
Offset?   4774(RET)
  Base          Offset       Old     New?
   000000       004774    000030    ;R(RET)
   000000       004774      <  X>   ;R   n(RET)
   000000       004776      <xxx>   (CTRL/Y)(RET)
*(CTRL/C)

.
```

If the monitor you want to alter is the hardware bootable monitor, write a new system bootstrap with the COPY/BOOT command.

I.4 Changing the Default Device for the SRUN Command

When you start a system job (by typing SRUN filnam), the default device on which the monitor looks for the program file is DK:. (You can run system jobs under an FB or XM monitor that includes the system job feature.) If you have a special application, you can change this default to any three-character device name.

In the patch, monitr.SYS is the name of the monitor file that you want to modify, ..SRDK is the value of that symbol from the monitor link map, and nnn is the new default device name.

```
.RUN SIPP(RET)
*monitr.SYS(RET)
Base?      0(RET)
Offset?   ..SRDK(RET)
  Base          Offset       Old     New?
   000000       ..SRDK     015270   ;R(RET)
   000000       ..SRDK      <DK >   ;Rnnn(RET)
   000000       ..SRDK+2    <AW1>   (CTRL/Y)(RET)
*(CTRL/C)

.
```

I.5 Changing the Default File Type for the SRUN Command

When you start a system job the default file type for the program file is .SYS. If you have a special application, you can change this default to any three-character file type.

In the patch, monitr.SYS is the name of the monitor file that you want to modify, ..SRUX is the value of that symbol from the monitor link map, and nnn is the new default file type.

```
.RUN SIPP(RET)
*monitr.SYS(RET)
Base?      0(RET)
Offset?   ..SRUX(RET)
  Base          Offset       Old     New?
   000000       ..SRUX     075273   ;R(RET)
   000000       ..SRUX      <SYS>   ;Rnnn(RET)
   000000       ..SRUX+2    <xxx>   (CTRL/Y)(RET)
*(CTRL/C)

.
```

INDEX

READER'S COMMENTS

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

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

Did you find errors in this manual? If so, specify the error and the page number.

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

- □ Assembly language programmer
- □ Higher-level language programmer
- □ Occasional programmer (experienced)
- □ User with little programming experience
- □ Student programmer
- □ Other (please specify)

Name Date

Organization

Street

City State Zip Code or Country

— — — Do Not Tear - Fold Here and Tape —

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

BSSG PUBLICATIONS TW/A14
DIGITAL EQUIPMENT CORPORATION
1925 ANDOVER STREET
TEWKSBURY, MASSACHUSETTS 01876

— — Do Not Tear - Fold Here —