March 1978

This document is intended for use by the system manager or system-level programmer responsible for generating and installing RT-11 at the user site. This manual, used with the *RT-11 System Release Notes,* provides all the information necessary to customize, generate, and exercise the RT-11 monitors and system programs.

RT-11

System Generation Manual

Order No. AA-5283B-TC

SUPERSESSION/UPDATE INFORMATION:	This manual supersedes DEC-11-ORGMB-A-D.
OPERATING SYSTEM AND VERSION:	RT-11 V03B

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, September 1977
Revised: March 1978

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
DECsystem-10
DECtape
DIBOL
EDUSYSTEM
FLIP CHIP
FOCAL
INDAC
LAB-8
DECSYSTEM-20
RTS-8
MASSBUS
OMNIBUS
OS/8
PHA
RSTS
RSX
TYPESET-8
TYPESET-11
TMS-11
ITPS-10

3/79-15

CONTENTS

CONTENTS (Cont.)

CONTENTS (Cont.)

FIGURES

TABLES

PREFACE

HOW TO USE THIS MANUAL

This document introduces the RT-11 V03B software kit to users receiving it for the first time. You use this document to create backup copies of your distribution medium, to build a working system if your distribution medium is not the system device, and, if necessary, to use SYSGEN to generate a customized RT-11 monitor.

Chapter 1 explains the contents of the kits. Chapter 2 contains step-by-step instructions for starting the system, creating a working system, and creating a backup copy of the system. Chapter 3 contains instructions on using SYSGEN to generate a customized RT-11 monitor. Chapter 4 describes how to assemble and link system programs.

Before performing any of the procedures described in this manual, read the RT-11 System Release Notes. That document explains all the new features of RT-11 and contains the detailed information that you need before you create a working system. Familiarity with the RT-11 system, as described in the RT-11 System User's Guide, is very helpful.

Once you understand the system, you can build your RT-11 system according to the instructions in Chapter 2. If you have special hardware considerations, or you are building a nonstandard RT-11 system, you should read Chapters 2 and 3 carefully before beginning the build.

If you are a FORTRAN user, you build your FORTRAN system after building the RT-11 system. See the RT-11 FORTRAN IV Installation Guide for instructions. If you are a BASIC-11 user, you build your BASIC-11 system after building the RT-11 system. See the BASIC-11/RT-11 Installation Guide for instructions.

DOCUMENTATION CONVENTIONS

This section describes the symbolic conventions used throughout this manual. Familiarize yourself with these conventions before you continue reading the manual.

1. Examples consist of actual computer output wherever possible. In these examples, user input is underlined where necessary to differentiate it from computer output.

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

3. To produce several characters in system commands, you must type a combination of keys concurrently. For example, hold down the CTRL key and type C at the same time to produce the CTRL/C character. Key combinations such as this one 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. The sample terminal dialogue provided in this document contains version numbers where they would normally appear. The version numbers given include xx's 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 an SPR to DIGITAL, you must include the complete version number.

RELATED DOCUMENTS

See the RT-11 Documentation Directory for information concerning related documents in the RT-11 library.

CHAPTER 1

OVERVIEW OF THE SOFTWARE KIT

1.1 RT-11 SOFTWARE KIT

The RT-11 binary software kit is available on the following media: RK05 disk, RL01 disk, RX01 and RX02 flexible diskettes, DECtape, 9-track magtape, and RK06 cartridge disk. Each kit contains an RT-11 system, user documentation, and all the materials necessary to build a customized RT-11 system for any valid target system. The components of an RT-11 software kit are inventoried on the checklist attached to the outside of the kit. You should verify the contents of the package against the checklist and report any discrepancies to DIGITAL's Software Distribution Center.

The RT-11 disk and DECtape kits contain "ready-to-run" RT-11 systems; you can mount copies of the masters and bootstrap them directly without modification.

The magtape kit contains an RT-11 magtape that you use to build an RT-11 system on the target system disk; you use a special program (MDUP) to initialize the system disk, copy the requisite files over to the system disk, and start the new system.

For instructions on building a working system from the distribution media see Chapter 2.

In addition to the "ready-to-run" RT-11 system distributed in the software kit, RT-11 software kits also contain all the components necessary to generate a customized RT-11 monitor. Although most users find the ready-to-run systems sufficient to meet their needs, users who want to do so and who have the requisite hardware can generate a custom RT-11 system using RT-11 SYSGEN.

See Chapter 3 for instructions on generating a custom system via SYSGEN.

1.2 DELIVERED MONITORS

Three variations of monitors are distributed in the binary kits. Although all three are not present for all system devices, the common versions are present and you can generate all supported variations for all system devices via SYSGEN. The three distributed RT-11 monitor types are the base-line single-job monitor (BL), the single-job monitor (SJ), and the foreground/background monitor (FB). Table 1-1 summarizes the differences among the RT-11 monitors. The extended memory monitor (XM) and its associated device handlers are not distributed in assembled form. You must perform a SYSGEN to create these files if you require extended memory support.

The monitor swap blocks are located in an external file called SWAP.SYS. This file serves as temporary storage for part of a program currently in memory when KMON, or the USR, or both must swap over the program. When KMON or the USR are no longer needed, the system reads the external swap file, which contains part of the executing program, back into main memory. In previous versions of RT-11, each monitor file reserved 24 blocks for swapping.

SWAP.SYS file is a necessary part of every RT-11 monitor; it must reside on your system volume along with the RT-11 monitor you choose to use. The external swap file is used in common by all the monitors. Thus, if you store more than one monitor on a particular volume, you need to keep only one copy of SWAP.SYS. Since SWAP.SYS is 24 blocks long, this represents a considerable saving in space on your system volume if you store more than one monitor file there.

1.2.1 The Base-Line Single Job Monitor

The base-line monitor (BL) is a specially generated version of the single-job monitor that has all optional monitor and device functions disabled. It has the minimum residency requirement of any possible RT-11 monitor and is intended for use in very small hardware configurations (8K words of memory), or in larger configurations where the application requires minimal executive support. The differences between the base-line monitor and the standard single-job monitor are as follows.

Base-line SJ	Standard SJ
Device support limited to RK, DX, DT, NL, LP, TT, and PC	General device support, with almost all normal devices installed
No graphic display support	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
No BATCH support	Supports BATCH
Minimal memory and system device requirements for monitor	Slightly larger memory and system device requirements for monitor

In a very limited hardware configuration or application, the base-line monitor is capable of performing all the system commands and can run most of the utilities. For 12K or larger systems, however, or 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 an easier to use and more flexible base. Note that installations that must run MAC8K in an 8K machine are required to use the base-line monitor for that purpose; MAC8K cannot run under the standard single-job monitor on less than a 12K machine.

1.2.2 The Single-Job Monitor

The single-job monitor (SJ) is the smallest standard RT-11 executive. It supports all hardware devices (with the exception of multi-terminal support and extended memory support) and runs all system utilities. It runs in any supported configuration with at least 8K words of memory but cannot make use of more than 28K words of memory.

The single-job monitor is the best choice for many RT-11 users. It has the fastest response times (at interrupt and keyboard level) and the minimum memory requirements of all the standard RT-11 monitors. 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.

1.2.3 The 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 interactive program development, nor is it intended to make RT-11 into 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 enhanced facilities above those offered in the SJ monitor. These include:

- The ability to run a foreground job.
- Enhanced terminal service.
- Optional multi-terminal support.
- Extended executive facilities for real-time applications.

The FB monitor requires at least 16K words of memory and a clock, and cannot use more than 28K words of memory.

If your application includes the need for a software priority, real-time application to run concurrently with normal system development and data-processing applications, then 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.

1.2.4 The 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 128K words of memory. The extended memory monitor provides a set of directives that allow advanced applications to make use of additional memory above 28K.

The XM monitor is significantly larger than the FB monitor, and its requirements that device handlers be resident and that the USR not swap add even more to the monitor memory overhead. If you do not have more than 32K words of memory, you should use the FB or SJ monitor. Even if you do have more than 32K words of memory, 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 more frugal FB and SJ monitors.

The XM monitor requires 32K words of memory, a memory management unit (KT11), and the extended instruction set (EIS) to operate. The XM monitor is not distributed on the RT-11 kit. You must perform a SYSGEN to create this monitor and its device handlers.

Table 1-1
Summary of Monitor Differences

	BL	SJ	FB	XM
Approximate Resident Size for RK System	1.8K	2K	4K	7K (includes resident USR)
Memory Support	8-28K	8-28K	16-28K	32-128K
Support FG Job	no	no	yes	yes
Enhanced Terminal Service	no	no	yes	yes
Multi-Terminal Support	no	no	optional	optional
Timer Facilities (.MRKT, .CMKT - including midnight date/time rollover)	no	optional	yes	yes
BATCH Support	no	optional	optional	optional
VT11/VS60 Support	no	optional	optional	optional
Error Messages On Fatal System Errors	no	optional	yes	yes
Devices Supported	RK,DX,DT, NL,LP,TT, PC	all	all	all
Approximate System Disk Requirements (Monitor and SWAP.SYS)	83 blocks	86 blocks	97 blocks	107 blocks

1.3 OPTIONS INCLUDED IN DISTRIBUTED MONITORS

The monitors distributed in the RT-11 binary kit were generated to include the most common options for the most common applications. The following information describes in detail which options are enabled and which are disabled in the distributed systems.

All the BL monitors on the distribution kit are identically configured; the only difference among them is the system device on which the monitor runs. The same is true for the SJ and FB monitors in the kit.

Most installations should be able to use these distributed monitors as is, thus eliminating the need for a SYSGEN. If minor customization is necessary (for example, RJS04 instead of RJS03, RP03 instead of RPR02 or 50 Hz clock instead of 60 Hz clock), see Chapter 2 for instructions on making these customizations via patches rather than SYSGEN. You should undertake the SYSGEN process only if you require support that is not immediately available or is not available with a customization patch in the distributed system. You must perform a SYSGEN for extended memory support.

Table 1-2 lists the options available in the distributed monitors. The table uses the following abbreviations:

STD = Standard in monitor
N/A = Not available in monitor
no = Disabled; if needed, must be enabled by generating a new system with SYSGEN. Attempted use produces an error.
yes = Available in monitor.
* = Support is present but must be installed with INSTALL command.

Table 1-2
Options Available In Distributed Monitors

Option	BL	SJ	FB
Timer Support (.MRKT and .CMKT- including midnight date/time rollover)	no	no	STD
Error Messages on System I/O Errors	N/A	yes	STD
Idle Loop Light Pattern	N/A	N/A	no
Multi-Terminal	N/A	N/A	no
Clock Frequency	60Hz	60Hz	60Hz
Line Clock versus Programmable Clock	N/A	Line	Line
Startup Command File In Bootstrap	yes	yes	yes
Parity Memory	N/A	no	no

(continued on next page)

Table 1-2 (Cont.)
Optional Available In Distributed Monitors

Option	BL	SJ	FB
Error Message on Power Fail	N/A	yes	yes
BATCH	N/A	yes	yes
Escape Sequence Processing in Terminal Service	N/A	no	no
Error Logging	N/A	no	no
RK11/RK05	yes	yes	yes
RL01	N/A	yes-2 drives	yes-2 drives
RJS03/RJS04	N/A	RJS03	RJS03
RK06/07	N/A	yes	yes
RF11	N/A	yes-2 platters	yes-2 platters
RP11/RPR02/RP03	N/A	RPR02	RPR02
RX01	yes	yes	yes
RX02	N/A	yes	yes
DECtape	yes	yes	yes
File-Structured Magtape			
TM11/TU10/TS03	N/A	yes-2 units	yes-2 units
TJU16	N/A	yes-2 units	yes-2 units
Hardware Magtape			
TM11/TU10/TS03	N/A	yes*	yes*
TJU16	N/A	yes*	yes*
TA11 Cassette Support	N/A	yes*	yes*
VT11/VS60 Graphics as ASCII terminals	N/A	yes-VT11	yes-VT11
Line Printer	yes	yes	yes
High-speed Reader/Punch	yes*	yes*	yes*
CR11 Card Reader	N/A	yes*	yes*
Null Handler	yes*	yes*	yes*
Extra Device Slots	0	0	0

CHAPTER 2

GETTING STARTED WITH RT-11

2.1 CONTENTS OF DISTRIBUTION KITS

All RT-11 binary software kits contain the same system components. For RK05, RL01, and RK06 cartridge disk kits, all the system files are included on one volume. In the other kits (9-track magtape, DECtape, and diskette) the same files are included, but they are spread across several volumes.

The files in the software kit can be logically grouped into several classes. Those files with the .SYS and .BL file types are the monitor and device handler files. Those with the .SAV file types are the system programs, while the .MAC files are source programs. The remaining miscellaneous files are subcomponents of one of the major utilities or special-purpose files required for system generation or operation. See Table 2-1 for a description of the functions of all the files in the distribution kit.

2.1.1 Monitor and Handler Naming Conventions

The monitor and handler files in the RT-11 system follow a rigid naming convention. This convention is important to the system at bootstrap time; the system bootstrap uses a file name search to determine which monitor file to boot and which handler files (and hence which devices) are present on the system.

RT-11 monitor file names are always of the following format:

```
     xxMNyy.SYS
```

The first two characters (xx) correspond to the two-character device mnemonic for this monitor's system device. Thus the first two characters are DX for an RX11/RXV11 RX01 monitor, RK for an RK11/RKV11 RK05 monitor, DY for an RX211/RXV21 RX02 monitor, DL for an RL11 RL01 monitor, DM for an RK611 RK06/07 monitor, DT for a TC11 TU56 DECtape monitor, or DP for an RP11 RP02 monitor. The last two characters (yy) are SJ, FB, or XM depending on whether this monitor is a single job, foreground/background, or extended memory version of the monitor. The middle two characters of any monitor file name are always MN, and the file type of any active monitor is always .SYS.

Thus, RKMNSJ.SYS is an RK11 RK05 single-job monitor; DMMNFB.SYS is an RK06/07 foreground/background monitor; and, DPMNXM.SYS is an RP11 RPR02 extended memory monitor.

Note that nothing prevents you from naming monitor files any name you desire for storage purposes, but only those files conforming to the above conventions are recognized at bootstrap time.

Three files in the RT-11 kit are named xxMNSJ.BL. These files are the base-line single job monitors. Because they are not active in the distribution kits and must be distinguished from the standard single-job monitors, they have a .BL file type. To use a base-line monitor, simply rename the existing xxMNSJ.SYS to another name, rename the base-line monitor file to xxMNSJ.SYS, install the bootstrap with a COPY/BOOT command, and reboot the system.

The RT-11 bootstrap is constructed in such a way that when the volume is booted, the bootstrap always looks for a monitor file that conforms to the system naming conventions. The name of the file that the bootstrap seeks has nothing to do with the current name of the file from which the bootstrap was written with the COPY/BOOT command. Rather, the file name to be sought is built into the file and always conforms to the naming conventions. Thus, if you have an RK05 foreground/background monitor named MYMON.ABC and you use the COPY/BOOT command to install the bootstrap from it (COPY/BOOT MYMON.ABC RK:), the system looks for the file RKMNFB.SYS at bootstrap time. If no such file is present, an error message appears; if another file named RKMNFB.SYS is present, the system bootstraps it instead of MYMON.ABC. For these reasons, DIGITAL recommends that you adopt the following procedures:

1. Use file types (such as .BL, .OLD, .NEW, etc.), rather than file names, to distinguish among various monitors on your disk. Always use file names that conform to the system naming conventions.

2. When switching monitors, perform the operations in the following order:

   a. Rename the monitor currently in use to another name (if you want to preserve it).

   b. Rename the monitor desired to xxMNyy.SYS (if necessary).

   c. Install the bootstrap with the COPY/BOOT command (if you desire permanent change of the bootstrap).

   d. Boot the new monitor.

By renaming the monitor file before installing the bootstrap, you avoid confusion as to which monitor file will be booted.

For example, to switch from the standard diskette single-job monitor to the base-line diskette single-job monitor, use the following series of commands.

```
.RENAME/SYSTEM DXMNSJ.SYS DXMNSJ.STD RET
.RENAME/SYSTEM DXMNSJ.BL DXMNSJ.SYS RET
.COPY/BOOT DXMNSJ.SYS DX:*.* RET
.BOOT DX: RET
```

RT-11 handler files are named according to the following convention:

yy.SYS and yyX.SYS

where yy is the two-character device mnemonic corresponding to the device for which the file serves as a handler. The SJ and FB monitors use the yy.SYS form of handler; the XM monitor uses the yyX.SYS form. Thus, RK.SYS is an RK11 handler for the SJ and FB monitors and LPX.SYS is a line printer handler for the XM monitor. The XM handlers are not distributed with RT-11. You must perform a system generation to select extended memory support.

Note that all three monitors (SJ, FB, XM) and both sets of handlers can reside on the same volume without conflict.

In the distribution kits, the file-structured magtape handlers are properly named as described above. The hardware handlers are provided with names of the form:

```
yyHD.SYS
```

If you need the hardware handlers rather than the file-structured handlers, you must rename and install them before use (see Section 2.4.2).

2.1.2 **Software Components**

The Table 2-1 identifies each file in the RT-11 binary kit.

Table 2-1
RT-11 Software Components

Name	Description
✓ SWAP.SYS	External monitor swap blocks
RKMNSJ.SYS	RK05 single-job monitor
RKMNFB.SYS	RK05 foreground/background monitor
RKMNSJ.BL	RK05 base-line single-job monitor
✓ DLMNSJ.SYS	RL01 single-job monitor
✓ DLMNFB.SYS	RL01 foreground/background monitor
DMMNSJ.SYS	RK06/07 single-job monitor
DMMNFB.SYS	RK06/07 foreground/background monitor
DXMNSJ.SYS	RX01 single-job monitor
DXMNFB.SYS	RX01 foreground/background monitor
DXMNSJ.BL	RX01 base-line single-job monitor
DYMNSJ.SYS	RX02 single-job monitor
DYMNFB.SYS	RX02 foreground/background monitor
DTMNSJ.SYS	DECtape single-job monitor
DTMNFB.SYS	DECtape foreground/background monitor
DTMNSJ.BL	DECtape base-line single-job monitor
DSMNSJ.SYS	RJS03/4 single-job monitor
DSMNFB.SYS	RJS03/4 foreground/background monitor
DPMNSJ.SYS	RP11 single-job monitor
DPMNFB.SYS	RP11 foreground/background monitor
RFMNSJ.SYS	RF11 single-job monitor
RFMNFB.SYS	RF11 foreground/background monitor
✓ TT.SYS	Terminal handler for SJ monitors
DT.SYS	SJ, FB DECtape handler
DP.SYS	SJ, FB RP11/RPR02/RP03 handler
✓ DX.SYS	SJ, FB RX11/RX01 diskette handler
DY.SYS	SJ, FB RX211/RX02 diskette handler
RF.SYS	SJ, FB RF11 handler
RK.SYS	SJ, FB RK11/RK05 handler
✓ DL.SYS	SJ, FB RL11/RL01 handler
DM.SYS	SJ, FB RK611/RK06/07 handler
DS.SYS	SJ, FB RJS03/4 handler
✓ LP.SYS	SJ, FB line printer handler
CR.SYS	SJ, FB card reader handler
MT.SYS	SJ, FB file-structured TM11 handler

(continued on next page)

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

Name	Description
MTHD.SYS	SJ, FB hardware TM11 handler
MM.SYS	SJ, FB file-structured TJU16 handler
MMHD.SYS	SJ, FB hardware TJU16 handler
NL.SYS	SJ, FB null handler
PC.SYS	SJ, FB high-speed paper tape handler
CT.SYS	SJ, FB TA11 cassette handler
BA.SYS	SJ, FB BATCH handler
PIP.SAV	File transfer utility
DUP.SAV	Disk maintenance utility
FORMAT.SAV	Disk formatting utility
RESORC.SAV	System resource display utility
DIR.SAV	Directory utility
SYSMAC.SML	System macro library
EDIT.SAV	Text editor
MACRO.SAV	MACRO assembler
MAC8K.SAV	Overlayed MACRO assembler for small configurations
CREF.SAV	Cross-reference utility
LINK.SAV	Linker
LIBR.SAV	Librarian
FILEX.SAV	Foreign file exchange utility
SRCCOM.SAV	Source compare utility
DUMP.SAV	File dump utility
PATCH.SAV	Patching utility for program memory images
PAT.SAV	Patching utility for program object modules
HELP.SAV	Help utility
HELP.TEC	Help utility command file
HELP.TXT	Help utility text file
BATCH.SAV	BATCH compiler
ERRUTL.SAV	Error logging utility
SYE.SAV	Error log reporting utility
PSE.SAV	Error log formatting utility
SYSGEN.SAV	System generation dialogue utility
SYSGEN.CND	SYSGEN input file
SYSTBL.CND	SYSGEN input file
ODT.OBJ	Debugging utility
VTMAC.MAC	Display handler macro file
VTHDLR.OBJ	VT11/VS60 display handler
PLOT55.OBJ	VT55 graphics terminal software
TEST55.FOR	Demonstration program for VT55
SYSF4.OBJ	System subroutines
GETSTR.FOR	FORTRAN subroutine source
PUTSTR.FOR	FORTRAN subroutine source
SYSMAC.MAC	System macro library source
MDUP.SAV	Magtape utility
MDUP.MT	Magtape bootstrap utility for TM11
MDUP.MM	Magtape bootstrap utility for TJU16
MBOOT.BOT	Magtape primary bootstrap
MSBOOT.BOT	Magtape secondary bootstrap
DEMOBG.MAC	Demonstration source
DEMOFG.MAC	Demonstration source
DEMOSP.MAC	Demonstration source
DEMOX1.MAC	Demonstration source

(continued on next page)

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

Name	Description
DEMOF1.FOR	Demonstration source
DEMOED.TXT	Demonstration source
STARTS.COM	SJ startup command file
STARTF.COM	FB startup command file
V3USER.TXT	Distribution kit message text file
SJ.MAC	SJ conditional source file for SYSGEN
FB.MAC	FB conditional source file for SYSGEN
XM.MAC	XM conditional source file for SYSGEN
SYSDEV.MAC	System device conditional source file for SYSGEN
KMON.MAC	Keyboard monitor source file for SYSGEN
USR.MAC	USR source file for SYSGEN
RMONSJ.MAC	SJ resident monitor source file for SYSGEN
RMONFB.MAC	FB/XM resident monitor source file for SYSGEN
KMOVLY.MAC	Keyboard monitor overlay source file for SYSGEN
EDTGBL.MAC	Monitor edit log and global definition file for SYSGEN
BSTRAP.MAC	Bootstrap source file for SYSGEN
MTTEMT.MAC	Multi-terminal EMT source file for SYSGEN
MTTINT.MAC	Multi-terminal interrupt service source file for SYSGEN
TT.MAC	TT.SYS source file for SYSGEN
EL.MAC	Error logger source file for SYSGEN
BA.MAC	BATCH handler source file for SYSGEN
LP.MAC	Line printer handler source file for SYSGEN
DL.MAC	RL01 handler source file for SYSGEN
RK.MAC	RK05 handler source file for SYSGEN
DT.MAC	DECtape handler source file for SYSGEN
DS.MAC	RJS03/4 handler source file for SYSGEN
PC.MAC	High-speed paper tape handler source file for SYSGEN
RF.MAC	RF11 handler source file for SYSGEN
DM.MAC	RK06/07 handler source file for SYSGEN
DX.MAC	RX01 handler source file for SYSGEN
DY.MAC	RX02 handler source file for SYSGEN
CR.MAC	Card reader handler source file for SYSGEN
DP.MAC	RP11 handler source file for SYSGEN
CT.MAC	Cassette handler source file for SYSGEN
TM.MAC	TM11 handler source file for SYSGEN
TJ.MAC	TJU16 handler source file for SYSGEN
NL.MAC	Null handler source file for SYSGEN
FSM.MAC	Magtape file support source file for SYSGEN
BATCH.MAC	BATCH compiler source file (for maintenance purposes)

2.1.3 Unsupported Software

The RT-11 distribution kits include eight other files in addition to those listed in Table 2-1. This software is not supported by DIGITAL, but is available from the DECUS program library. These files are included in the distribution kit for your convenience. The files are as follows:

README.TXT	Information file for TECO macros
TECO.SAV	Text editor
LOCAL.TEC	TECO macro
SORT.TEC	TECO macro
INSERT.TEC	TECO macro
EDIT.TEC	TECO macro
VT52.TEC	TECO macro
VEG.TEC	TECO macro

The file README.TXT contains information about the other software. To read the file, use one of the following commands:

```
.TYPE    README.TXT RET

.PRINT   README.TXT RET     (if a line printer is available)
```

2.2 STARTING THE SYSTEM

The remaining discussion in this chapter deals with the specifics of bootstrapping the distribution media, building a working system from the distribution kits, and making minor customizations to the system without doing a SYSGEN. Finally, it contains instructions for an exercise to demonstrate that the system is functioning.

The procedures that follow contain RT-11 keyboard commands. To understand completely the processes that are being carried out, you must read and understand the RT-11 System User's Guide before continuing with this manual.

NOTE

> You should be able to execute most of the procedures that follow without error. If an error not explained in this manual occurs, refer to the RT-11 System Message Manual.
>
> If the processor halts, a serious error has occurred and you must repeat the entire process from the beginning. If a user error occurs within a section, repeat from the beginning of that particular section.
>
> You can correct typing errors at any time using the standard RT-11 input editing techniques (DELETE and CTRL/U).

Installing RT-11 on a particular configuration involves the following logical steps:

1. Bootstrapping the distribution medium.
2. Preserving the distribution medium by making a copy.
3. If the system device is the same as the distribution device, building a working system volume by deleting unwanted components from the copy and customizing the system for the particular installation.
4. If the system device is not the same as the distribution device, building a working system volume by transferring the relevant components to the system volume and customizing them for the particular installation.
5. Preserving the working system on a backup volume so that you need not repeat this procedure every time you reinitialize the system.

The discussion generally describes the logical operation that you are performing, followed by specific commands that perform that operation. If you wish to follow this material in a step-by-step format, you can enter the commands literally for the desired result. An understanding of what operations you are performing, and why, is important, however, because such an understanding better prepares you for dealing with error conditions and requirements specific to your installation. If you understand how the system works, you can substitute any command that has the desired result for any of the commands given below. The essential aspect is to complete all the logical steps discussed.

The commands and options used are given in their full form; you can abbreviate them. Default devices and file types are used explicitly; you can eliminate them.

2.2.1 Bootstrapping the Distribution Medium

If the distribution medium is disk, diskette, or DECtape, you can bootstrap the system directly and use it to copy the distribution volume. (In the following sections, the term "disk" is used to mean both cartridge disks and flexible diskettes.) Stop the processor to halt any previous program that is running, mount the distribution volume (the first volume, if you have a set of diskettes) in Unit 0, WRITE PROTECTED if possible, and proceed to Section 2.2.1.2.

If the distribution medium is magtape, you must build a minimal system on the system disk before you can copy the distribution tape. Proceed to the next section.

2.2.1.1 **Booting the Distribution Magtape** - Mount the RT-11 system magtape #1 on unit 0, ensuring that no write ring is inserted in the back of the tape reel. Ensure that the magtape is positioned at the load point; if it is not, manually position it.

If the system has a hardware bootstrap capable of bootstrapping the magtape, boot the tape and proceed. If not, use the bootstrap in Appendix C.

The magtape moves as the primary bootstrap loads the secondary bootstrap file MSBOOT.BOT.

MSBOOT responds by printing on the terminal.

Response:
```
MSBOOT V01-xx
*
```

Type the name of the magtape build program that is to be used to build the system disk. If you are using a TM11, type MDUP.MT (RET) ; if you are using a TJU16, type MDUP.MM (RET) .

Type:
```
MDUP.MT (RET)
```

or

```
MDUP.MM (RET)
```

The magtape should move as the specified MDUP program is loaded.

Response:
```
*
```

The next step is to initialize the system disk and scan it for bad blocks before you can copy the system files over to it. If the system disk is RF11, you must specify the number of platters. If the system disk is RJS04 instead of RJS03, you must flag the extra capacity. 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
/T:2 to specify RJS04 instead of RJS03
/T:n to specify n RF11 platters
```

Make sure a formatted disk is mounted, WRITE-ENABLED, and ready.

If the system disk is other than RJS04 or RF11,

Type:
```
xxn:/Z/B (RET)
```

where xxn is the device name and unit.

If the system disk is RJS04,

Type:
```
DSn:/Z/B/T:2 (RET)
```

If the system disk is RF11,

Type:
```
RF:/Z/B/T:n (RET)
```

where n is the number of disk platters present on the RF11 subsystem.

In all cases, there can be a significant delay as the disk is scanned for bad blocks and the directory is created.

Response:
```
*
```

Issue the following command to build a minimal system on the target disk.

Type:
```
xxn:A=MT0: (RET)     if the magtape is TM11, or

xxn:A=MM0: (RET)     if the magtape is TJU16
```

where xxn is the name and unit of the system device.

The tape moves as a single-job monitor, swap file, terminal handler, line printer handler, the magtape handlers, and the utilities PIP, DUP and DIR are copied to the system volume. When all the files are copied, MDUP boots the minimal system from the disk.

```
Response:  RT-11SJ    V03B-xx
           ?KMON-F-Command file not found
           .
```

You are now running from the system volume and have enough file maintainence commands at your disposal to complete the building process.

NOTE

> MDUP does not support automatic replacement of bad blocks for RK06, RK07, or 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.

Copy four additional files (MBOOT.BOT, FORMAT.SAV, STARTS.COM, V3USER.TXT) needed for the rest of this process from the distribution magtape.

Type:

```
     COPY xx0:(MBOOT.BOT,FORMAT.SAV,STARTS.COM,V3USER.TXT)/POSITION:-1 DK:*.* (RET)
```

where xx is MT if a TM11 is being used, MM if TJU16.

```
Response:  Files copied:
           xx0:MBOOT.BOT to DK:MBOOT.BOT
           xx0:FORMAT.SAV to DK:FORMAT.SAV
           xx0:STARTS.COM to DK:STARTS.COM
           xx0:V3USER.TXT to DK:V3USER.TXT
```

2.2.1.2 **Formatting New Media** - Some kinds of media, such as RK05 disks, are not formatted for use on RT-11 systems when you purchase them. Flexible diskettes are generally 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 as double density in order to obtain optimum performance. Use the FORMAT program to format your new volumes, if they are either RK05 disks or RX02 diskettes.

If your system has more than one disk drive, mount a new RK05 disk or diskette in device unit 1. Boot the distribution volume.

```
Response:  RT-11SJ    V03B-xx

           .D 56=5015
           .TYPE V3USER.TXT
                 .
                 .
                 .
           .D 56=0
```

Invoke the FORMAT program.

Type: .R FORMAT RET

Response: *

If your system has more than one disk drive, specify unit 1 as the volume to format.

Type: dd1: RET

where dd represents the physical device name, either RK or DY.

Response: dd1:/FORMAT-Are you sure?

Type: Y RET

Formatting begins immediately.

Response: ?FORMAT-I-Formatting complete

*

If you have more new volumes to format, repeat this process for each volume.

Type: CTRL/C

Response: .

If your system has only one disk drive, you must switch the system volume with the volume you need to format. First, specify unit 0 as the volume to format, and include the /W option.

Type: dd0:/W RET

where dd represents the physical device name, either RK or DY.

Response: dd0:/FORMAT-Are you sure?

Type: Y RET

Response: Insert volume you wish to format. CONTINUE (Y/N)?

Now remove the distribution volume from device unit 0 and insert the new volume you need to format.

Type: Y RET

Formatting takes place immediately.

Response: Replace original volume. CONTINUE (Y)?

Now remove the newly formatted volume from device unit 0 and replace the distribution volume.

Type: Y RET

Response: ?FORMAT-I-Formatting complete

*

If you have more new volumes to format, repeat this process for each volume.

Type: CTRL/C

Response: .

2.2.1.3 **Preserving the Distribution Medium** - The first operation that you must perform with the running RT-11 system is to copy the distribution volume. If machine failure or human error were to destroy the distribution volume, you would need the backup to continue.

If there is only one drive for the distribution medium, you cannot make a copy. Keep the distribution volume WRITE PROTECTED, and proceed to Section 2.3.

If there is a second drive for the distribution medium, mount a blank, formatted volume on the second drive (Unit 1) WRITE ENABLED. Wait for the drive to come to the READY state; if the distribution medium is magtape, position the tape at the beginning-of-tape mark.

If you have a hardware bootstrap capable of bootstrapping the system (distribution) disk, boot the disk and proceed; if not, see Appendix C for bootstrapping instructions.

Even if you just completed the previous section on booting from magtape and the system is already running from disk, reboot the disk. You must execute the startup command file to ensure proper operation of the rest of the procedure on all systems.

NOTE

> When bootstrapped, the distributed system responds by printing the identifying message, then executes a startup command file that is distributed with the system. This startup file sets the terminal service so that the system works with all serial baud rates, prints a file containing a message for previous RT-11 system users, then resets the terminal service for normal use. When execution of the startup file is complete, the monitor prompts with a dot.
>
> If you have an LA36 DECwriter II, a VT52 DECscope terminal, or other terminals that do not require fill characters, you will notice unusual delays after each RET LF sequence in the startup file dialogue. This is because the first DEPOSIT command in the command file sets the RT-11 terminal service to output 10 null characters after each carriage return, which is necessary on certain older terminals. This delay is removed by the second DEPOSIT command.
>
> If you are using a serial LA30, VT05, or other terminal that requires fill characters, you should deposit the correct fill character and count in location 56 each time that you reboot the system during the installation process. Use a DEPOSIT command. Section 2.4.1 describes the appropriate values for specific terminals.

```
Response:  RT-11SJ    V03B-xx

           .D 56=5015

           .TYPE V3USER.TXT
           Welcome to RT-11 Version 3B ......
           .D56=0
```

Set the date with the monitor DATE command.

```
Type:      DATE dd-mmm-yy RET
Response:  .
```

Prepare to copy the distribution volume by initializing the backup volume on Unit 1.

If you are copying onto a disk, you must initialize the directory and scan the volume for bad blocks.

If your disk is an RK06, RK07, or RL01, you can select automatic replacement for bad blocks.

```
Type:      INITIALIZE/REPLACE xx1: RET
```

If your disk is not an RK06, RK07, or RL01, you must cover the bad blocks.

```
Type:      INITIALIZE/BADBLOCKS xx1: RET
```

where xx is the name of the distribution device.

If you are copying onto a DECtape, the bad block scan is unnecessary.

```
Type:      INITIALIZE DT1: RET
```

If you are copying onto a magtape, you must initialize the tape and write the primary bootstrap.

```
Type:      INITIALIZE/FILE:MBOOT.BOT xx1: RET
```

where xx is MT or MM.

In all cases, the system responds with an "are you sure?" query because initializing a volume destroys any previous information on that volume.

```
Response:  xx1:/Init are you sure?
```

where xx is the name of the device you are initializing.

```
Type:      Y RET
```

There can be a significant delay as the output device is scanned for bad blocks and the new directory is created.

Response:

Next, copy all the files from Unit 0 to Unit 1.

If you are copying onto a disk or DECtape,

```
Type:      COPY/SYSTEM xx0:*.* xx1:*.* RET
```

where xx is the device name.

If you are copying onto a magtape, use the POSITION option to rewind and then to avoid rewinding between each file as the tape is copied.

```
Type:       COPY/SYSTEM xx0:*.*/POSITION:0 xx1:*.*/POSITION:-1 RET
```

where xx is MT or MM.

In both cases, the system logs each of the files copied on the terminal as they are transferred and prompts with a dot when the operation is complete.

```
Response:   Files copied:
            xx0:aaaaaa.eee to xx1:aaaaaa.eee
                      .
                      .
                      .
            xx0:zzzzzz.eee to xx1:zzzzzz.eee
            .
```

If you are copying a disk or DECtape, complete the operation by installing a bootstrap on the backup volume. In the case of magtape, this step is not necessary because the bootstrap was installed with the INIT command.

If you are copying onto a disk or DECtape,

```
Type:       COPY/BOOT xx1:xxMNSJ.SYS xx1: RET
```

where xx is the device name.

```
Response:   .
```

In all cases, you should now be able to bootstrap the backup copy of the distribution volume.

If the copy is a disk,

```
Type:       BOOT xx1: RET
```

where xx is the device name. The system responds by printing the identifier and the message for previous users. If it does, dismount the copy from unit 1 and mount it on unit 0.

If the copy is a DECtape, change the unit number of the copy from Unit 1 to Unit 0. Using the hardware bootstrap, boot the copy from Unit 0; the system should respond by printing the system identifier and the message for previous users.

If the copy is a magtape, rewind the tape and mount it on Unit 0. Using the hardware bootstrap, boot the copy from Unit 0; the system responds with the MSBOOT identifier as in Section 2.2.1.1.

In all cases, if the copy does not bootstrap properly, repeat this section.

Store the distribution master in a safe place along with the other distribution volumes.

2.2.2 Summary

To boot a distribution magtape:

1. Boot the tape on Unit 0.
2. Specify to MSBOOT the name of the MDUP file to be used.
3. Initialize the target system disk with MDUP using the /Z, /B and /T:n options.
4. Create a minimal system on the target system disk by typing xx0:A=Mx0: RET to MDUP. The minimal system boots automatically.
5. Copy the files MBOOT.BOT, FORMAT.SAV, STARTS.COM, and V3USER.TXT to the minimal system.

To copy the distribution volume:

1. Boot the distribution disk or DECtape on Unit 0, WRITE PROTECTED, or in the case of magtape, boot the minimal system created above.
2. Mount a blank volume on Unit 1.
3. Enter the date.
4. Initialize the blank volume using /REPLACE on RK06, RK07, or RL01, /BADBLOCKS on all other disks, and /FILE:MBOOT.BOT on magtape.
5. Copy all files using the /SYSTEM option for all media and the /POSITION option on magtape to avoid rewinds.
6. If copying a disk or DECtape, install the bootstrap on the backup volume with COPY/BOOT.

2.3 BUILDING A WORKING SYSTEM DISK

By now you have built a minimal system from the distribution magtape (if the distribution medium was magtape) and, if your configuration allows, you have copied the distribution medium. The remainder of this section deals with building your working system by deleting unnecessary components from the copy or creating a system on a device other than the distribution device. Finally, it describes creating a backup for the working system.

2.3.1 Preparing the Disk

If the system device is to be a disk other than the distribution device, mount the copy of the distribution volume (or the original if no copy is available) in unit 0, WRITE PROTECTED if possible.

If the system device is to be the same as the distribution device, mount the copy of the distribution volume in unit 0, WRITE ENABLED.

Bootstrap the copy of the distribution disk (or the original if you have no copy) from unit 0, or in the case of magtape distribution, bootstrap the minimal system from the system volume. The system should respond by printing its identifying message and executing the startup command file. When the message is completely printed, the system prompts with the dot and waits.

```
Response:  RT-11SJ    V03B-xx

           .D 56=5015

           .TYPE V3USER.TXT
           Welcome to RT-11 Version 3B ......

           .D 56=0
```

If you have not already read it, see the NOTE in Section 2.2.1.3 concerning the startup command file.

Enter the date:

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

Response:  .
```

The essential steps during this phase are:

1. Initialize the target system disk if it is not the distribution medium.
2. Select the components to be included in the working system.
3. Transfer the desired components to the target system disk, or delete undesired components from the existing system disk.
4. Install the bootstrap on the target system.
5. Optimize the target system and compress any free space.
6. Preserve the working system.

The first step is to initialize the target system volume and scan it for bad blocks. If you are already running off the target system volume, or if you are building a system from magtape, DO NOT perform this step; proceed to Section 2.3.2. If you are about to build a system on a volume other than the one from which you are running, you must scan the disk for bad blocks.

If the disk is an RK06, RK07, or RL01, you can select automatic replacement for bad blocks.

```
Type:      INITIALIZE/REPLACE xxn: (RET)
```

where xxn is the device and unit number on which the system volume is to be built.

If the disk is not an RK06, RK07, or RL01, you must cover the bad blocks.

```
Type:      INITIALIZE/BADBLOCKS xxn: (RET)
```

where xxn is the device and unit number on which the system volume is to be built.

```
Response:  xxn:/Init are you sure?
```

Type:
```
Y (RET)
```

There can be a significant delay as the output device is scanned for bad blocks and the new directory is created.

Response:
```
.
```

2.3.2 Determining Disk Capacity

Obtain a directory of the system volume to determine how much free space is available for system files. If you are already running from the target system volume, the directory tells you how many blocks are occupied by the components currently residing on the disk and how much free space is available for additional components. If the target system volume is a disk other than the distribution volume, the directory shows an empty volume and reflects the total capacity of the target disk.

Type:
```
DIRECTORY xxn: (RET)
```

where xxn: is the device and unit number on which the system volume is to be built.

Response:
```
dd-mmm-yy
filnam.typ n dd-mmm-yy  (if any files are on volume)
    .
    .
    .
n Files, bbb Blocks
fff Free blocks
.
```

The number of free blocks, fff, represents the total number of blocks additional system components can occupy. If you wish to use part of your system volume for data storage as well, you should subtract the space desired for data storage to determine the number of blocks available for system components.

The next step is to determine which system components are to be included in the working system. If the distribution medium is RK05, RL01, or RK06 disk, you can use the list of kit components in Table 2-1 to decide which files to include; if not, obtain directories of the distribution volumes to determine which components are on which volume and how much space they occupy. Mount each volume in the distribution kit (on a drive other than that being used for the system volume) and use the DIRECTORY command to list the directory for each volume before proceeding.

2.3.3 Choosing System Components

In general, you need only one monitor on a particular working system. If you desire to use more than one monitor in a particular application, all the monitors can be present on the same volume, but this practice consumes disk space that may be more useful for data storage. You only need one copy of SWAP.SYS on a system volume regardless of how many monitor files you store on the volume.

You do not need any system programs you do not intend to use; but remember that you do need some system programs to use the keyboard monitor commands. Appendix B of the RT-11 System User's Guide summarizes the keyboard monitor commands and lists the system utility program that each command requires.

You do not need any source files (except the demonstration sources) until you do a SYSGEN. You need the demonstration sources to perform the exercises in Section 2.5 of this manual or in the Introduction to RT-11. Once you perform these exercises, the demonstration source files are no longer useful.

You do not need VTMAC.MAC and VTHDLR.OBJ unless you have a VT11/VS60.

You do not need a startup command file unless you want one.

You do not need the MDUP and .BOT files unless you have magtape.

You do not need the handler files for any devices you do not have.

You must have TT.SYS on your system volume if you plan to use the BL or SJ monitor.

You do not need TECO.SAV or the .TEC files unless you plan to use the TECO text editor.

2.3.4 **Creating the Working Disk**

Having chosen the system components, you can create the working disk in one of two ways:

1. You can delete files from a copy of the distribution disk, resulting in a smaller working system.

2. You can copy the desired system components from the distribution volumes to another legal system device, creating a working system for the new device.

If the working disk is to consist of all the files on the distribution volume and you are already running from the distribution copy, you have your working disk. Proceed to Section 2.3.5.

If the working disk is being built on a disk other than the distribution volume, proceed to Section 2.3.4.2.

2.3.4.1 **Creating a System on the Distribution Device** - If the working disk is to consist of a subset of the files on the current system disk (that is, you are running off the copy of the distribution disk and you will be deleting unwanted components):

Type: `DELETE/SYSTEM *.*` (RET)

The system responds by listing each file name on the distribution volume followed by a question mark. If you want the listed file included in the working system type (RET) or N (RET) to prevent file deletion. If you do not want the listed file to be included in the working system, type Y (RET) to delete the file.

Response:

```
Files deleted:
aaaaaa.ttt?
```

Type:

```
Y (RET)      to exclude file from or
N (RET)      to include file in working system
```

Continue until all the files on the disk have been listed. When all files are listed:

Response:
```
?PIP-W-Reboot
.
```

At this point, the working disk contains the files selected for the working system. Proceed to Section 2.3.4.3.

2.3.4.2 **Creating a System on Another Device** - If the working disk is being built on a disk other than the distribution volume, or if it is to include files from other distribution volumes, successively mount each volume in the distribution kit.

If the distribution medium is magtape, use the POSITION option to first rewind, then to avoid rewinding between each file.

Type:
```
COPY/SYSTEM/QUERY xx0:*.*/POSITION:0 yyn:*.* (RET)
```

where xx is MT or MM and yyn: is the target working system device.

Otherwise,

Type:
```
COPY/SYSTEM/QUERY xxn:*.* yyn:*.* (RET)
```

where xxn: is the distribution volume and yyn: is the target working system device.

The system responds by listing each file name on the distribution volume followed by a question mark. If you want the listed file included in the working system, type Y (RET) to include the file. If you do not want the file included, type N (RET) to prevent the file from being copied.

Response:
```
Files copied:
xxn:aaaaaa.ttt to yyn:aaaaaa.ttt?
```

Type:
```
Y (RET)        to include file or
N (RET)        to exclude file from working system
```

Continue until all the files on the distribution volume have been listed. When all the files are listed, the system prompts with a dot.

Repeat this step for each volume in the distribution kit (if there is more than one), until all the files desired in the working system have been copied.

At this point, the working disk contains all the files needed for the working system. Proceed to the next section.

2.3.4.3 **Booting the Working Disk** - If you wish to use the base-line monitor instead of the standard SJ monitor, rename the standard SJ monitor (if you want to save it), then rename the base-line monitor on the working disk to xxMNSJ.SYS.

Type:
```
RENAME/SYSTEM xxn:xxMNSJ.BL xxn:xxMNSJ.SYS (RET)
```

where xx is the system device and n is the unit number for the working system.

```
Response:  ?PIP-W-Reboot      (If the working system is the current
                              running system.)
                  .
```

Install the bootstrap in the working system.

```
Type:      COPY/BOOT xxn:xxMNyy.SYS xxn: RET
```

where xxn: is the working system device and unit number and xxMNyy is the file name of the monitor to be started each time the working system is bootstrapped.

Finally, bootstrap the working system:

```
Type:      BOOT xxn: RET
```

where xxn: is the working system device and unit number.

The system responds with its identifying message and executes any startup command file present. If a startup file is not present, KMON prints an error message. To avoid printing the error message each time you bootstrap the system, simply create a null startup file.

2.3.5 Compressing the System Disk

Once you have created the system disk, you should compress it to consolidate all the free space and minimize fragmentation caused during the creation process. If you are running from an RJS04 disk or multiple platter RF11 disk, make the disk size customizations described in Sections 2.4.3 and 2.4.4 before proceeding.

```
Type:      SQUEEZE xxn: RET
```

where xxn: is the working system device.

```
Response:  xxn:/Squeeze are you sure?

Type:      Y RET
```

The system automatically reboots after the compression of the system disk is complete.

```
Response:  RT-11SJ    V03B-xx
                .
                .
                .
```

The system is now ready for customization and preservation.

2.3.6 Building SYSLIB

The RT-11 linker always searches the default system library (SYSLIB.OBJ) to resolve any undefined globals at the end of a link operation. For most installations, SYSLIB contains the system subroutines found in SYSF4.OBJ, the FORTRAN OTS routines, and installation specific libraries of application subroutines.

To add subroutines to SYSLIB, you simply append them to the current system library by using the librarian. This example creates the initial SYSLIB from the system subroutines in SYSF4.OBJ.

To create the original SYSLIB,

```
Type:       LIBRARY/CREATE SYSLIB.OBJ SYSF4.OBJ RET

Response:   .
```

To add new modules to SYSLIB from the file xxxxxx.OBJ,

```
Type:       LIBRARY/INSERT SYSLIB.OBJ xxxxxx.OBJ RET
```

where xxxxxx is the name of the file containing the object modules that you are inserting into the library.

2.3.7 Adding FORTRAN Subroutines to SYSLIB

The GETSTR and PUTSTR routines in SYSF4 are compiled as threaded code. If you generate your FORTRAN system for in-line code only, you must use in-line library subroutines as well. If you attempt to link FORTRAN in-line code object modules with the threaded code subroutines GETSTR and PUTSTR contained in SYSF4, undefined globals result. To create in-line code versions of GETSTR and PUTSTR, follow the following procedure.

First, compile the files GETSTR.FOR and PUTSTR.FOR that are on the distribution kit:

```
    .FORTRAN/CODE:xxx  GETSTR,PUTSTR RET
```

where xxx represents the type of object code to be generated. The default is selected at FORTRAN installation time. The valid values are:

EAE	(Selects EAE hardware)
EIS	(Selects EIS hardware)
FIS	(Selects EIS and FIS hardware)

Replace the threaded code modules in the library with the in-line code modules you created above:

```
    .LIBRARY   SYSLIB   GETSTR/REPLACE,PUTSTR/REPLACE RET
```

2.3.8 Building Working Systems For Small System Devices

For most RT-11 systems with larger mass storage devices (RK05, RL01, RK06/07, RP11, RJS04), the system volume is large enough to store all the system components and still have sufficient space for data storage.

Systems with smaller system devices, however (RX01, RX02, DECtape, RF11, RJS03) are more constrained. The following are several techniques for effectively using the system when system volume capacity is limited.

1. Limit the system volume to only those necessary, frequently used system components. Place only one monitor file on the system volume, and limit the handler files to those that are actually available on the configuration. For example, a typical RX01 dual diskette system need have only:

    SWAP.SYS
    DXMNSJ.SYS
    TT.SYS
    LP.SYS - only if the system includes a line printer

as the system files. Note that the RX01 diskette handler (DX.SYS) is not necessary because it is already resident as part of DXMNSJ.SYS. If you use the FB monitor, TT.SYS is not necessary because it is resident in the monitor.

2. Limit the system programs to those that you use frequently. PIP, DUP and DIR are necessary for most of the keyboard commands to function. EDIT, LINK, and perhaps HELP are also programs you use frequently. The other utilities (BATCH, FILEX, PATCH, LIBR, PAT, ODT, FORMAT, SRCCOM, DUMP, and RESORC) you use less frequently and you can run them off a second utilities disk mounted in unit 1 when you need them. Thus, a typical diskette system might have:

System Disk	SWAP.SYS	
	DXMNSJ.SYS	
	TT.SYS	
	LP.SYS	
	BA.SYS	
	PIP.SAV	
	DUP.SAV	
	DIR.SAV	
	EDIT.SAV	
	LINK.SAV	
	HELP.SAV	- required for the HELP command
	HELP.TEC	- required for the HELP command
	HELP.TXT	- required for the HELP command
Utilities Disk	BATCH.SAV	
	LIBR.SAV	
	FILEX.SAV	
	PATCH.SAV	
	SRCCOM.SAV	
	DUMP.SAV	
	RESORC.SAV	
	PAT.SAV	
	ODT.OBJ	
	FORMAT.SAV	

The system disk contains all the components necessary to execute the majority of keyboard commands and perform common program preparation functions. Mount the utilities disk in unit 1 instead of the data disk when you need one of the utilities; then either temporarily copy the utility off the utilities disk to the system disk, or run it directly from the utilities disk.

3. When running from a small system disk, assign the default device DK: to the data device (unit 1 or another peripheral). This causes most temporary files, data files, etc., to default to the second unit, and minimizes demand for system device capacity.

4. Create several system disks, each devoted to a particular function, and change the system disk as normal job flow changes the functions you need. Wait for a logical stopping point in the job flow to change disks; do not arbitrarily remove the system disk in the middle of an operation. For example, in addition to the system and utilities disks mentioned above, you can create a MACRO disk for use when doing assemblies and a FORTRAN disk for use when using FORTRAN. Note that SYSMAC.SML is frequently required for assembling, and SYSLIB.OBJ is frequently required for linking.

A typical MACRO disk has:

SWAP.SYS
DXMNSJ.SYS
TT.SYS
LP.SYS
PIP.SAV
DUP.SAV
DIR.SAV
EDIT.SAV or TECO.SAV
LINK.SAV
MACRO.SAV
CREF.SAV
SYSMAC.SML
ODT.OBJ

A typical FORTRAN disk has:

SWAP.SYS
DXMNSJ.SYS
TT.SYS
LP.SYS
PIP.SAV
DUP.SAV
DIR.SAV
EDIT.SAV or TECO.SAV
FORTRA.SAV
FORTRA.HLP - useful for the novice user

A typical LINK disk for use when linking FORTRAN programs has:

SWAP.SYS
DXMNSJ.SYS
TT.SYS
LP.SYS
PIP.SAV
DUP.SAV
DIR.SAV
LINK.SAV
SYSLIB.OBJ - includes FORTRAN library functions

The typical job flow for use of such multiple disks is as follows:

a. With the system disk in unit 0 and a data disk in unit 1, set the default device DK: to unit 1 and edit the source program onto DK:.

b. Change the system disk to the MACRO disk, reboot, and assemble and link the program using DK for the .OBJ and .SAV files.

c. Change unit 0 back to the system disk, reboot, and debug the program.

2.3.9 Positioning the System Files for Optimal Performance

RT-11 operates correctly regardless of the position of the system components on the system device. The system accesses certain components more frequently than others. By positioning the files so that the most frequently accessed files are on the lowest disk tracks, you can minimize head motion and seek times and thus improve system performance. The effect of positioning the files is significant on DECtape and noticeable on diskette.

If you desire minimal system response times to keyboard commands, copy the system components onto the working system in the following order:

```
SWAP.SYS
Monitor files
Handler files
Commonly used utilities (PIP, DUP, DIR, EDIT, LINK)
Commonly used libraries (SYSMAC.SML, SYSLIB.OBJ)
Commonly used language processors (MACRO, FORTRAN, BASIC)
Less frequently used utilities
```

Within each category, order the files from most frequently used to least frequently used. Thus, you should make the most frequently used handler the first file in the handler section.

Once you establish the proper file order, you should use the keyboard monitor SQUEEZE command on the system disk to force all the system components toward the front of the disk. This serves to minimize the seek time, minimize the fragmentation, and avoid further motion of system files during future SQUEEZE operations, eliminating the need to reboot.

2.3.10 Preserving the Working System

Once you have factored in all the above considerations and built a satisfactory working system, you must preserve the working system on a backup medium. This backup protects you from having to rebuild the working system should machine failure or human error destroy the copy of the system in use.

If your system requires minor customization (50HZ clock, hardware magtape support, etc.), perform the customizations described in Section 2.4 and return to this section to preserve your customized system. If the system has minimal memory available, and you desire to use the smaller, slower MAC8K instead of MACRO, you should rename MACRO.SAV to another name, then rename MAC8K.SAV to MACRO.SAV to ensure proper operation of the MACRO command.

If patches are to be installed in the system, install them at this point.

If the backup medium is disk and no further customizations are necessary, you are ready to preserve the working system. Proceed to Section 2.3.10.2.

2.3.10.1 **Preserving the System on Magtape** - If you are preserving the system on magtape, the following files, if present at all, must be copied to the magtape in the following order:

```
SWAP.SYS
monitor
TT.SYS
LP.SYS
MT.SYS or MM.SYS
PIP.SAV
DUP.SAV
DIR.SAV
```

There can be additional files between them, but the above order is important to proper operation of MDUP.

If no customizations are necessary, you are ready to preserve the working system. Mount a blank magtape on the backup drive, WRITE ENABLED.

```
Type:       INITIALIZE/FILE:MBOOT.BOT xxn: (RET)
```

where xx is MT or MM and n is the unit number.

```
Response:   xxn:/Init are you sure?

Type:       Y (RET)

Response:   .
```

You must also copy the secondary boot and the MDUP program onto the tape to allow you to rebuild the system from the backup tape.

```
Type:       COPY MSBOOT.BOT xxn:MSBOOT.BOT (RET)
```

where xx is MT or MM and n is the unit number.

```
Response:   .

Type:       COPY MDUP.xx xxn:MDUP.xx (RET)
```

where xx is MT or MM and n is the unit number.

```
Response:   .
```

Copy all the remaining files to the magtape. Use a separate command line for each file to be copied. Include the /POSITION:-1 option to prevent the tape from rewinding after each file is copied. Be sure to copy the files to the magtape in the order specified at the beginning of this section. Use commands of the following form:

```
Type:       .COPY/SYSTEM filnam.typ xxn:*.*/POSITION:-1 (RET)
```

where filnam.typ represents the file name and file type of the file to be copied; xx is MM or MT, and n is the unit number.

The files transferred are logged on the terminal as they are copied, and the system prompts with a dot when done.

```
Response:   File copied:
            DK:aaaaaa.ttt to xxn:aaaaaa.ttt

            .
```

If the working system needs restoration, you can bootstrap the backup copy and use it to rebuild the working system using the procedures previously described in this section.

2.3.10.2 **Preserving the System on Disk** - If the backup medium is disk, mount a blank volume on the backup medium drive, WRITE ENABLED.

If the disk is an RK06, RK07, or RL01, you can specify automatic replacement for bad blocks.

Type: `.INITIALIZE/REPLACE xxn:` RET

where xxn is the device name and unit number.

If the disk is not an RK06, RK07, or RL01, you must cover the bad blocks.

Type: `INITIALIZE/BADBLOCKS xxn:` RET

where xxn is the device name and unit number.

Response: `xxn:/Init are you sure?`

Type: `Y` RET

Response: `.` (After delay for initialization)

Type: `COPY/SYSTEM *.* xxn:*.*` RET

where xx is the device name and n is the unit.

The files transferred are logged on the terminal as they are copied, and the system prompts with a dot when done.

Response:

```
Files copied:
DK:aaaaaa.ttt to xxn:aaaaaa.ttt
    .
    .
    .
DK:zzzzzz.ttt to xxn:zzzzzz.ttt

.
```

If the backup is disk, you must install the bootstrap.

Type: `COPY/BOOT xxn:xxMNyy.SYS xxn:` RET

where xxn is the backup device and unit number and xxMNyy.SYS is the monitor file name.

If the working system needs restoration, you can bootstrap the backup copy and use it to rebuild the working system using the procedures previously described in this section.

2.3.10.3 **Summary** - To build a working system from the distribution kits:

1. Initialize the target system volume, scanning for bad blocks.
2. Obtain a directory of the system volume to determine the space available for system components.
3. Obtain directories of the distribution volumes to determine which files are on which volumes and how much space each file requires.
4. Choose which components must go into the working system.
5. If the working system is on the same medium as the copy of the distribution kit, make the working system by deleting unwanted components from the copy of the distribution volume.

6. If the working system is on a medium other than the distribution medium, make the working system by transferring the desired components to the working system volume.

7. If the base-line SJ monitor is to be used instead of the standard SJ monitor, rename the standard SJ monitor and then rename the base-line monitor to xxMNSJ.SYS.

8. Install the bootstrap on the working system volume.

9. Boot the working system volume.

10. Reorder the system files (if necessary or desired) for optimal performance.

11. Build the system library SYSLIB.

12. SQUEEZE the working system volume to minimize fragmentation.

13. Perform any minor customizations or install any patches that are necessary.

14. Preserve the working system on a backup volume by:

    a. INITIALIZing a bootable volume.

    b. COPYing the working system.

    c. INSTALLing the bootstrap (disk only)

2.3.11 Switching Monitors

The COPY/BOOT command installs the bootstrap for the specified monitor file as the permanent bootstrap for the particular volume. Every time you bootstrap the device, the permanent monitor is loaded.

You can alter the bootstrap to permanently boot another monitor file with the COPY/BOOT command. Each time you use this command, it permanently installs the named monitor file as the monitor to be booted from this volume.

If you have more than one monitor on a given volume (for example, SJ and FB) and you need to temporarily switch from one to another, use the monitor BOOT command. When you specify a file name rather than a device name to the BOOT command, the specified file is booted. The monitor bootstrapped in this fashion remains active until you issue another BOOT command or you physically reboot the volume.

If you have an FB monitor on your working disk and you need to switch from the running monitor to the FB monitor,

Type: `BOOT xxMNFB.SYS` RET

where xx is the working system device name.

Response: `RT-11FB    V03B-xx`
(followed by any commands executed from
the startup command file)

Return to the single-job monitor by rebooting the device.

Type:
```
BOOT xxn: RET
```

where xxn: is the working system device.

Response:
```
RT-11SJ    V03B-xx
```
(followed by any commands executed from the startup command file)
```
.
```

2.3.12 Installing The Bootstrap

Whenever you create a new system device, you must not only copy the system files to the new volume, but you must also install the bootstrap so that the new volume can be booted.

You install the bootstrap on block-replaceable devices (disk or DECtape) via the COPY/BOOT command. You install the bootstrap on magtape by using the /FILE:MBOOT.BOT option to the INITIALIZE command.

To create a bootable disk,

1. Copy the monitor file to the new disk.

2. Install the bootstrap from the new monitor with the COPY/BOOT command.

For example:

```
.COPY/SYSTEM RKMNSJ.SYS RK1:RKMNSJ.SYS RET
.COPY/BOOT RK1:RKMNSJ.SYS RK1:*.* RET
```

To create a bootable magtape:

1. Initialize the magtape, specifying the primary bootstrap file (MBOOT.BOT) as an argument to the INITIALIZE command.

2. Write the secondary bootstrap program (MSBOOT.BOT) as the first file on the tape.

For example:

```
.INITIALIZE/FILE:MBOOT.BOT MT1: RET
.COPY MSBOOT.BOT MT1:MSBOOT.BOT RET
```

2.4 CUSTOMIZING THE SYSTEM WITHOUT DOING A SYSGEN

The RT11 system components are suitable for most applications and configurations as distributed. In addition, the most common alterations can be made by patching the distributed monitors, eliminating the need for most installations to do a SYSGEN. Only if your installation requires customization not described in this section (such as extended memory or multi-terminal support) should you undertake a SYSGEN.

Many of the patches given below are made to symbolic locations rather than absolute addresses. For the actual values to be used when making the patches, see Table 4-2 of the RT-11 System Release Notes. When making a patch that contains a symbolic value, simply look up the symbol in the table and substitute the absolute value for the symbolic name in the patch.

2.4.1 Establishing Fill Characters

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 established a mechanism by which any number of fills may follow any character. The byte at location 56 (octal) contains the character to be followed by fillers and the byte at location 57 (octal) contains the number of null fills to be used. These locations are initially set to zero, which results in no fillers being generated (normal operation for LT33, LA30, LA36, VT50, VT52, VT55, and VT61).

Depending on the terminal, modify the locations as follows:

Terminal	Loc 56	Loc 57	Resulting Word (octal)
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

You can change the proper octal word permanently in the monitor by using PATCH to modify locations 56 and 57 in the monitor file. For example, the following patch causes a fill of four nulls after each carriage return.

```
.R PATCH RET

FILE NAME--
*xxMNyy.SYS/A RET
*56\0    15 LF
57\0     4 RET
*E
```

Once you make this change, recopy the bootstrap with the COPY/BOOT command and reboot the monitor with the BOOT command, all programs that use the monitor for console I/O will operate correctly.

2.4.2 Installing Other Devices

RT-11 is distributed with hardware TM11 handler, hardware TJU16 handler, high-speed paper tape reader/punch handler, cassette handler, card reader handler and null handler support all available but not installed in the running monitors. Installation of support for these devices involves use of the monitor SET, SHOW, INSTALL, and REMOVE commands, the startup command file capability, and an occasional patch.

The distributed monitors (with the exception of the base-line SJ monitor) are distributed with RK05, RL01, RJS03, RK06/07, RP11, RX01, RX02, DECtape, RF11, TM11 file-structured magtape, line printer, and TJU16 file-structured magtape support all preinstalled. The configuration that requires support for all the aforementioned, preinstalled devices is extremely rare. The procedure for installing one of the noninstalled handlers or a user-written handler is:

1. Use the monitor SHOW command to determine which devices are active in the current monitor.

2. Add commands to the startup command file that REMOVE an unwanted device and INSTALL the desired device. If more than one version of the monitor is on the system device (SJ and FB, for example), add the same commands to all startup files.

In the rare case where there are not sufficient device slots to support all the devices in the configuration, you can SYSGEN a monitor to include extra device slots. If use of the startup command file is an unacceptable method for installing the handler, you can permanently install the handlers at SYSGEN time.

For example, to remove the RK06/07 handler (DM.SYS) and install the null handler (NL.SYS), edit the startup command file as follows:

```
Type:         EDIT STARTx.COM RET
```

where x is S, F or X

```
Response:     *

Type:         R/J ESC  ESC

Response:     *

Type:         IREMOVE DM RET
              INSTALL NL RET
              ESC EX ESC  ESC

Response:     .
```

Whenever you reboot the monitor, the system automatically removes the undesired RK06/07 handler and installs the null handler.

2.4.2.1 **Installing Card Reader Support** - Make certain CR.SYS is present on the system device, then use the SHOW command to determine if a free slot is available. If not, add a REMOVE command to the startup command file that removes an unwanted device, then add an INSTALL command to add CR. For example, to add the necessary commands to an already existing startup file:

```
Type:         EDIT STARTx.COM RET
```

where x is S, F or X.

```
Response:     *

Type:         R/J ESC  ESC

Response:     *

Type:         IREMOVE xx RET
              INSTALL CR RET
              ESC EX ESC  ESC
```

where xx is an unwanted device name.

```
Response:     .
```

Whenever you reboot the monitor, the system will automatically install the CR handler.

2.4.2.2 **Installing High-Speed Paper Tape Support** - If the configuration includes a high-speed reader/punch, the desired handler is PC.SYS. If it includes only a high-speed reader, you also use PC.SYS.

Make certain the paper tape handler is present on the system device, then use the SHOW command to determine if a free slot is available. If not, add a REMOVE command to the startup command file to remove an unnecessary device, then add an INSTALL command to install the paper tape support. For example, to add the necessary commands to an already existing startup file:

```
Type:       EDIT STARTx.COM (RET)
```

where x is S, F or X

```
Response:   *

Type:       R/J (ESC) (ESC)

Response:   *

Type:       IREMOVE xx (RET)
            INSTALL PC (RET)
            (ESC) EX (ESC) (ESC)
```

where xx is an unnecessary device

```
Response:   .
```

Each time you reboot the monitor with the new startup command file, the system will automatically install paper tape support.

2.4.2.3 **Installing Hardware Magtape Support** - The system is distributed with file-structured TM11 and TJU16 support enabled as the magtape support. If you desire hardware support, you can switch from file-structured to hardware handler. Rename the file-structured handler if you wish to save it, then rename the hardware handler to MT.SYS or MM.SYS and reboot the system. If the file-structured handler was already installed, no re-installation is necessary; if not, you must also install the device.

To install TM11 hardware support in place of TM11 file-structured support, make certain that MTHD.SYS is present on the system volume.

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

Response:   ?PIP-W-Reboot
            .
```

Reboot the system.

Because the TM11 file-structured handler and hardware handler are for the same device and MT support is already installed, you can switch between the two by simply renaming the handler and rebooting the system. INSTALL and REMOVE are not necessary but can be used to switch between hardware and file-structured handlers if you need to switch without rebooting the system.

To install TJU16 hardware support in place of TJU16 file-structured support, make certain that MMHD.SYS is present on the system volume.

```
Type:         RENAME/SYSTEM MMHD.SYS MM.SYS (RET)

Response:     ?PIP-W-Reboot
              .
```

Reboot the system or make the handler active with INSTALL and REMOVE, as with TM11 hardware support.

2.4.2.4 **Setting Magtape Parity and Density** - RT-11 includes magtape support for both 7- and 9-track drives at 800 BPI and odd parity. If you need to operate at different parity or density settings, you can use the monitor SET command as described in Section 4.4 of the RT-11 System User's Guide.

2.4.3 Specifying the Number of RF11 Platters

RT-11 is distributed with RF11 fixed-head disk support initialized for one platter. To allow RT-11 to make use of more than one platter, you must modify the device size table in the various monitor files and the device size entry in the RF handler as follows:

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

Use the following patch to modify an RF monitor for multiple RF/RS11 platters.

```
.R PATCH (RET)

FILE NAME--
*RFMNxx.SYS/A (RET)           (where xx is SJ, FB or XM)
*SYDVSZ/2000     yyyy (RET)   (where yyyy is value from table
                              above)
*E
```

NOTE

See the RT-11 System Release Notes, Table 4-2, for the address of SYDVSZ in each of the RF monitors.

Use the following patch to modify the RF handler for multiple RF/RS11 platters.

```
.R PATCH (RET)

FILE NAME--
*RF.SYS (RET)
*54/2000 yyyy (RET)   (where yyyy is value from above table)
*E
```

Once you modify the monitor and install the handler, you can use the INITIALIZE command to automatically initialize the RF11 directory to proper capacity, or you can use the SQUEEZE command on an RF11 disk already in use to update the directory to the proper capacity.

2.4.4 Changing RJS03 Support to RJS04 Support

RT-11 is distributed with RJS03/4 disk support initialized for RJS03. To allow complete use of all the space available on an RJS04 disk, modify the device size table in the various monitor files and the device size entry in the DS handler as follows:

Location	RJS03 Value	RJS04 Value
SYDVSZ in monitors; 54 in DS.SYS	2000	4000

Use the following patch to modify a DS monitor for RJS04 disk:

```
.R PATCH (RET)

FILE NAME--
*DSMNxx.SYS/A (RET)                 (where xx is SJ, FB or XM)
*SYDVSZ/2000     4000 (RET)
*E
```

NOTE

See the RT-11 System Release Notes, Table 4-2, for the address of SYDVSZ in each of the DS monitors.

Use the following patch to modify the DS handler to support RJS04 disks.

```
.R PATCH (RET)

FILE NAME--
*DS.SYS (RET)
*54/2000     4000 (RET)
*E
```

Once you modify the monitor and handler, you can use the INITIALIZE command to automatically initialize the RJS04 directory to proper capacity or the SQUEEZE command to update ' the capacity of a disk already in use.

2.4.5 Changing RP02 Support to RP03 Support

The RP11 support provided in the distribution kit is initialized for RP02 only. The RT-11 file structure can accommodate a maximum of 65536(10) blocks. The 40000(10) block RP02 cartridge, therefore, can be accommodated as a single logical unit, while an RP03 cannot.

You can alter the RT-11 RP02 support to accommodate RP03 as follows:

1. Consider each RP03 drive as two logical units of 40000 blocks each; in essence, a single RP03 drive looks like two RP02

drives to the system. Access the cartridge on physical unit n as logical DPn and DPn+4; thus, you reference drive 0 as DP0: and DP4:, drive 1 is DP1: and DP5:, etc. Note that although an RP03 is physically one device, it is two separate devices to the system. Each logical unit has its own complete directory and data space.

2. Patch the DP monitors and the DP handlers as described below.

Use the following patch to modify a DP monitor for RP03 drives:

```
.R PATCH RET

FILE NAME
*DPMNxx.SYS/A RET                    (where xx is SJ or FB)
*RP23/404         1404  RET
*E
```

NOTE

See RT-11 System Release Notes, Table 4-2, for the actual location of RP23 in each monitor and in the handlers.

Use the following patch to modify the DP handler for RP03 drives.

```
.R PATCH RET

FILE NAME--
*DP.SYS RET         (or DPX.SYS for extended memory handler)
*RP23/404           1404 RET
*E
```

Note that the system can support a maximum of 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. If the system contains only RP02s, you must not make the above changes and the system can support as many as eight units.

Once you make these patches, you can access the second half of the RP03 disks as units DPn+4.

2.4.6 Specifying a 50-Cycle Clock Rate

In the distributed version of RT-11, the keyboard TIME command calculations are based on a 60-cycle clock rate. To cause the TIME command to base calculations on a 50-cycle clock rate, modify the monitor so that bit 5 is set in the monitor configuration word.

Use the following patch to modify the monitor for a 50-cycle clock rate:

```
.R PATCH RET

FILE NAME--
*xxMNyy.SYS/A RET                (where xx is system device name
                                 and yy is SJ or FB)
*CONFIG/nn       nn+40 RET       (The value of CONFIG varies
                                 from configuration to
*E                               configuration;  add 40(8) to it
                                 and type new value.)
```

NOTE

> See the RT-11 System Release Notes, Table 4-2, for the exact address of CONFIG in each monitor.

2.4.7 Changing the Location of VT11/VS60 Floating Vectors

The floating vector region on the PDP-11 is situated in locations 300 to 476; VT11/VS60 display processor vectors are normally located at 320 to 332. However, the VT11/VS60 display vectors can be forced to float by the addition of other devices. For example, VT11/VS60 device vectors may be changed if a DL11 device (a communications device) is added.

The RL01 disk has its vector at 330. If the VT11/VS60 is left at its normal location, 320, it uses 330 as its shift-out vector. If your system has both an RL01 disk and a VT11 or VS60 display processor, you must move the VT11/VS60 vector.

The following patch modifies the monitor for a different VT11/VS60 vector address.

```
.R PATCH (RET)

FILE NAME--
*xxMNyy.SYS/A (RET)               (where xx is the system device
                                  and yy is SJ or FB).
*GTVECT/320      yyy (RET)        (where yyy is the new vector
                                  address)
*E
```

NOTE

> See the RT-11 System Release Notes, Table 4-2, for the exact address of GTVECT in each monitor.

This patch enables the text scroller (GT ON), the display file handler, EDIT, BASIC graphic extensions, and FORTRAN graphic extensions to function properly on the system without further patching.

2.4.8 Customizing the Editor

2.4.8.1 Reducing the Size of the Text Window - The editor is constructed in such a way that if a VT11/VS60 is in use, 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 feeds (or form feed characters), the window can overflow onto the scrolled editing commands, making that portion of the screen difficult to read.

In most applications, this problem does not occur; when it does, you can see the obscure lines by advancing the cursor several lines to bring them into clear view.

If the problem is troublesome for your particular application, you can remove it by reducing the size of the window displayed with the following patch:

```
.R PATCH (RET)

FILE NAME--
*EDIT.SAV (RET)
*DSARG/12        n (RET)
*E
```

where n is the number of lines to be displayed above and below the cursor; n should be smaller than the value currently in DSARG to eliminate the problem.

NOTE

> See the RT-11 System Release Notes, Table 4-2, for the exact address of DSARG.

2.4.8.2 **Using EDIT on Terminals with Non-standard ESCAPE Code** - Certain older terminals generate 175 (octal) or 176 (octal) when you type the ESCAPE (or ALTMODE) key, rather than the standard 33 (octal). Because codes 175 (octal) and 176 (octal) represent legitimate characters on more modern terminals, EDIT does not recognize ESCAPE as the command terminator in these older terminals.

You can patch the editor as follows to allow it to operate correctly with a non-standard ESCAPE code.

```
.R PATCH (RET)

FILE NAME--
*EDIT.SAV (RET)
*ALTMDE\         33       n (RET)
*E
```

In the patch shown above, n represents the octal code that is generated when you type ESCAPE on your terminal.

NOTE

> See the RT-11 System Release Notes, Table 4-2, for the actual address of ALTMDE.

Once you alter EDIT in the preceding manner, you can use ESCAPE as documented in the EDIT chapter of the RT-11 System User's Guide.

2.4.9 Setting an Upper Limit on File Size

RT-11 is distributed in such a way that the maximum size of a file allocated in a general .ENTER request is half the largest space, or the entire second largest space available, whichever is larger. This is satisfactory for most applications and should be left unchanged. It is possible that certain applications require that an upper limit be set on the size of a file; for these applications (and these only), make the following patch.

```
.R PATCH (RET)

FILE NAME--
*xxMNyy.SYS/A (RET)                  (where xx is system device
                                     name and yy is SJ, FB or XM)
*MAXBLK/177777   nnnn (RET)          (where nnnn is the octal
                                     number of blocks defining
                                     the maximum file size for a
                                     general .ENTER)
*E
```

NOTE

See the <u>RT-11 System Release Notes</u>, Table 4-2, for the actual value of MAXBLK for each monitor.

2.4.10 Running RT-11 in Less Memory Than Is Available

The SJ and FB monitors have bootstraps that allow the system to run in less memory than is available on the system (for example, RT-11 can be bootstrapped to run in the lower 12K or 8K of a 16K machine). Most applications require that RT-11 make use of all memory available, and the system is distributed so that it automatically does so.

If (and only if) an application requires that RT-11 run in less memory than is available and a hardware switch register is present, make the following patch.

```
.R PATCH (RET)

FILE NAME--
*xxMNyy.SYS/A (RET)         (where  xx is the system device name
                            and yy is SJ or FB)
*BHALT/ 407      0 (RET)
*E
```

NOTE

Obtain the value of BHALT from Table 4-2 of the <u>RT-11 System Release Notes</u>.

Once you make the patch and write a new system bootstrap on the device (with the COPY/BOOT command), a halt occurs whenever the system is booted.

At this point, set the switch register to one of the values from Table 2-2 and press CONTINUE; the bootstrap operation completes for the specified memory size.

If your application requires that RT-11 run in less memory than is available and a hardware switch register is not present, use the following patch.

```
.R PATCH RET

FILE NAME--
*xxMNyy.SYS/A RET                      (where xx is the system
                                        device name and yy is SJ
                                        or FB)


*BHALT/ 407       415 RET              (where nnnnnn is a value
*BHALT+34/ 5712  12702 LF              from Table 2-2)
*BHALT+36/ 771   nnnnnn RET

*E
```

Table 2-2
Values for Memory Sizes

Value	Size in Words
40000	8K
44000	9K
50000	10K
54000	11K
60000	12K
64000	13K
70000	14K
74000	15K
100000	16K
104000	17K
110000	18K
114000	19K
120000	20K
124000	21K
130000	22K
134000	23K
140000	24K
144000	25K
150000	26K
154000	27K
160000	28K
>160000	Use all available memory.

2.4.11 Modifying the Listing Page Length in MACRO and CREF

RT-11 MACRO and CREF set the number of lines printed per listing page at 60. This line count is satisfactory for applications with line printers that use paper 10.5 inches long. If you use paper of a different size (for example, 8.5 inches long) or if you do not have a line printer, patch MACRO and CREF as follows.

For MACRO:

```
.R PATCH RET

FILE NAME--
*MACRO.SAV RET
*LINPPG/74        n RET                 (where n is the new line
                                         count in octal)
*E
```

For CREF:

```
        .R PATCH (RET)

        FILE NAME--
        *CREF.SAV (RET)
        *LINPPG/74       n (RET)              (where n is the new line
                                              count in octal)
        *E
```

NOTE

See the RT-11 System Release Notes, Table 4-2, for the actual address of LINPPG in MACRO and CREF.

2.4.12 Changing the Default Output Device from Line Printer to Terminal

Several monitor commands default the output device to LP (for example, DUMP and PRINT). If your configuration does not have a printer, you can cause these commands to default to the terminal by adding an ASSIGN TT LP command to the startup command file. This causes all system references to the device LP: to use the terminal.

For example, to default all output to the terminal, add the ASSIGN TT LP command to an existing startup file as follows.

Type: `EDIT STARTx.COM (RET)`

where x is S, F or X.

Response: *

Type: `R/J (ESC) (ESC)`

Response: *

Type:

```
IASSIGN TT LP (RET)
(ESC) EX (ESC) (ESC)
```

Response: .

Each time you bootstrap the system, the reassignment of the default device takes place.

2.4.13 Modifying LINK to Change the Default SYSLIB Device

The linker normally looks on the system device (SY:) when attempting to locate the default system library (SYSLIB.OBJ).

To change the device on which SYSLIB.OBJ resides, make the following PATCH to LINK.SAV.

```
        .R PATCH (RET)

        FILE NAME--
        *LINK.SAV/O (RET)
        *1:SYSLIB/75250 P(dev) (RET)   (where dev is the name of
                                       the desired SYSLIB device)
        *E
```

NOTE

> See the RT-11 System Release Notes, Table 4-2, for the actual values of SYSLIB in the released system.

2.4.14 Extending Directory Size

When you initialize a device, you can specify additional directory segments by use of the INITIALIZE command /SEGMENTS option. Once you have written files on the device, however, you cannot change the directory size without using the INITIALIZE command again, thus destroying existing directory information.

Although DIGITAL recommends that, whenever possible, you use the INITIALIZE command to determine directory size, you can use the following program to change the size of a directory on line without reinitializing the disk. You should use the program only as a last resort and only on a disk that has been thoroughly backed up. Enter it as instructed in the comments. Instructions for its use are also in the comments.

```
;+
;DIREXT - PROGRAM TO ADD SEGMENTS TO AN RT-11 DIRECTORY
;
;TO USE:
;
;   THE FIRST ENTRY IN THE DIRECTORY MUST BE <UNUSED>, AND ITS LENGTH MUST
;BE TWICE AS MANY BLOCKS AS THE NUMBER OF SEGMENTS THAT ARE TO BE ADDED.
;THE FIRST ENTRY MAY BE MADE <UNUSED> SIMPLY BY CALLING PIP TO MOVE THE FIRST
;FILE ON THE DEVICE.  ONCE THE UNUSED AREA HAS BEEN CREATED, DIREXT IS
;RUN BY TYPING:
;
;               .R DIREXT
;THE RESPONSE:  *
;
;A COMMAND LINE IS ENTERED OF THE FORM 'DEV:/N:M' WHERE DEV IS THE DEVICE
;WHOSE DIRECTORY IS TO BE EXPANDED AND 'M' IS THE NUMBER OF SEGMENTS
;(NOT BLOCKS) TO ADD TO THE DIRECTORY.
;
;FOR EXAMPLE:
;               *RK:/N:14       WILL ADD 12 (DECIMAL) SEGMENTS TO THE
;                               DIRECTORY ON RK0:, FOR WHICH THE FIRST
;                               FILE MUST BE AN <UNUSED> AREA OF AT
;                               LEAST 24 BLOCKS
;
;THERE ARE TWO POSSIBLE ERROR MESSAGES:
;
; ?DIREXT-F-Hard I/O error      WILL OCCUR IF THE DIRECTORY FOR THE DEVICE
;                               CANNOT BE READ OR WRITTEN WITHOUT A
;                               HARDWARE ERROR.
;
; ?DIREXT-F-Illegal command     WILL OCCUR ON ANY OF THE FOLLOWING CONDITIONS:
;                               1) INAPPROPRIATE COMMAND STRING
;                               2) FIRST ENTRY IN DIRECTORY NOT <UNUSED>
;                               3) REQUESTED NUMBER OF SEGMENTS WOULD CAUSE
;                                  THE TOTAL TO EXCEED THE ALLOWABLE 31(10)
;                               4) SIZE OF <UNUSED> AREA TOO SMALL TO
;                                  ACCOMODATE REQUESTED DIRECTORY INCREASE
;
;IF THE ?DIREXT-F-Illegal command MESSAGE OCCURS, THE DIRECTORY IS UN-
;MODIFIED AND THE PROGRAM IS RESTARTED.
;
;IF THE ?DIREXT-F-Hard I/O error MESSAGE OCCURS, THE DIRECTORY HAS PROBABLY
;BEEN DESTROYED.
;
;TO CREATE DIREXT, TYPE THIS TEXT INTO THE FILE 'DIREXT.MAC' WITH THE EDITOR.
;
;TO ASSEMBLE,   .MACRO DIREXT
;
;TO LINK,       .LINK DIREXT
;
;-
        .TITLE  DIREXT - DIRECTORY EXTENDER
        .MCALL  .CSIGEN,.READW,.WRITW,.EXIT,.PRINT
ILLCMD: .PRINT  #MSG1                   ;TYPE '?ILLEGAL COMMAND'
START:  .CSIGEN #DSPACE,#DEXT,#0        ;GET COMMAND STRING
        DEC     (SP)+                   ;ANY OPTIONS?
        BNE     ILLCMD                  ;NO-BAD COMMAND
        CMP     (SP)+,#101516           ;/N:M GIVEN ON FIRST INPUT FILE?
        BNE     ILLCMD                  ;NO-BAD COMMAND
        MOV     (SP)+,R1                ;NUMBER OF SEGMENTS TO ADD INTO R1
        MOV     #6,R2                   ;BLOCK FOR CURRENT SEGMENT IN R2
        MOV     #DIRBUF,R3              ;R3 POINTS INTO DIRECTORY BUFFER
        .READW  #AREA,#3,R3,#256.,R2            ;READ FIRST DIRECTORY SEGMENT
        BCS     HERR                    ;READ ERROR
        CMP     12(R3),#1000            ;FIRST ENTRY <UNUSED>?
        BNE     ILLCMD                  ;NO-BAD COMMAND
        MOV     (R3),R4                 ;NUMBER OF SEGMENTS NOW IN DIR INTO R4
        ADD     R1,R4                   ;TOTAL SEGMENTS FOR NEW DIR IN R4
        MOV     R4,R5                   ;REMEMBER NEW SEGMENT TOTAL
        CMP     R4,#31.                 ;IS NEW TOTAL TOO LARGE?
        BHI     ILLCMD                  ;YES-BAD COMMAND
```

```
         ASL     R1                   ;# OF BLOCKS NEEDED FOR NEW SEGMENTS
         CMP     R1,22(R3)            ;<UNUSED> LARGE ENOUGH TO TAKE NEW SEGMENTS?
         BGT     ILLCMD               ;NO-BAD COMMAND
         ADD     R1,10(R3)            ;UPDATE FILE START ADDRESS
         SUB     R1,22(R3)            ;REDUCE <UNUSED> SIZE
LOOP:    MOV     R4,(R3)              ;UPDATE NUMBER OF SEGMENTS TO NEW TOTAL
         .WRITW  #AREA,#3,R3,#256.,R2     ;WRITE SEGMENT BACK OUT
         BCS     HERR                 ;WRITE ERROR
         DEC     R5                   ;ALL SEGMENTS UPDATED?
         BLE     DONE                 ;YES
         TST     (R2)+                ;NO-ADD 2 TO BLOCK NUMBER
         .READW  #AREA,#3,R3,#256.,R2     ;READ NEXT SEGMENT
         BCC     LOOP                 ;AND UPDATE
HERR:    .PRINT  #MSG2                ;PRINT '?HARD I/O ERROR'
DONE:    .EXIT
         .ENABL  LC
MSG1:    .ASCIZ  '?DIREXT-F-Illegal command'
MSG2:    .ASCIZ  '?DIREXT-F-Hard I/O error'
         .EVEN
DEXT:    .WORD   0,0,0,0
AREA:    .BLKW   5
DIRBUF:  .BLKW   256.
DSPACE:
         .END    START
```

2.4.15 Building a System for Use With Introduction to RT-11

The Introduction to RT-11 contains exercises that you execute on a running RT-11 system. The following lists the minimum components that the system volume must contain so that you can successfully execute the exercises.

The files needed for the Introduction to RT-11 exercises are:

File Name	Approximate Block Size
SWAP.SYS	24
xxMNSJ.SYS	62
xxMNFB.SYS	74
TT.SYS	2
LP.SYS (if appropriate)	2
other handlers (if appropriate)	2 each
DIR.SAV	17
PIP.SAV	16
DUP.SAV	21
LINK.SAV	29
EDIT.SAV	19
SRCCOM.SAV	11
LIBR.SAV	10
MACRO.SAV	45
CREF.SAV	6
SYSMAC.SML	37
ODT.OBJ	9
DEMOED.TXT	1
DEMOX1.MAC	4
DEMOSP.MAC	10

Approximately	484 Subtotal for MACRO only exercises

FORTRA.SAV	205	
SYSLIB.OBJ	199	
DEMOF1.FOR	2	

Approximately	406	Additional if FORTRAN exercises
BASIC.SAV Approximately	85	Additional if BASIC exercises

Approximately	892	Total for all exercises.

If your system volume is 1000 blocks or larger, it is to your advantage to have all these files on the same system volume.

If, however, your system volume cannot accommodate all files at once, you must build two or more disks for use with the Introduction to RT-11.

If your system volume is an RX01 diskette, build the following four disks.

The system volume:	SWAP.SYS DXMNSJ.SYS DXMNFB.SYS TT.SYS LP.SYS (if appropriate) PIP.SAV DUP.SAV DIR.SAV LINK.SAV EDIT.SAV SRCCOM.SAV LIBR.SAV ODT.OBJ MACRO.SAV SYSMAC.SML CREF.SAV DEMOED.TXT DEMOF1.FOR DEMOX1.MAC DEMOSP.MAC
The FORTRAN language volume:	SWAP.SYS DXMNSJ.SYS TT.SYS LP.SYS (if appropriate) PIP.SAV DUP.SAV DIR.SAV EDIT.SAV FORTRA.SAV DEMOF1.FOR
The LINK volume:	SWAP.SYS xxMNSJ.SYS LP.SYS (if appropriate) TT.SYS PIP.SAV DUP.SAV DIR.SAV LINK.SAV SYSLIB.OBJ

The BASIC volume:

SWAP.SYS
xxMNSJ.SYS
LP.SYS (if appropriate)
TT.SYS
PIP.SAV
DUP.SAV
DIR.SAV
LINK.SAV
EDIT.SAV
BASIC.SAV

If your system disk is an RX02 diskette, build the following two disks.

The system volume:

SWAP.SYS
DYMNSJ.SYS
DYMNFB.SYS
TT.SYS
LP.SYS (if appropriate)
PIP.SAV
DUP.SAV
DIR.SAV
LINK.SAV
EDIT.SAV
SRCCOM.SAV
LIBR.SAV
ODT.OBJ
MACRO.SAV
SYSMAC.SML
CREF.SAV
DEMOED.TXT
DEMOF1.FOR
DEMOX1.MAC
DEMOSP.MAC
SYSLIB.OBJ
BASIC.SAV

The FORTRAN language volume:

SWAP.SYS
DYMNSJ.SYS
TT.SYS
LP.SYS (if appropriate)
PIP.SAV
DUP.SAV
DIR.SAV
SRCCOM.SAV
EDIT.SAV
FORTRAN.SAV
DEMOF1.FOR
SYSLIB.OBJ
LINK.SAV

2.4.16 Accessing Nonsystem Disks on Single Disk Systems

You can access source disks and other nonsystem disks on single disk systems with more than 8K words of memory as follows:

1. Boot the system disk and enter the date. SET USR NOSWAP.

2. LOAD the handler for the device to which the desired files will be transferred.

3. Run PIP. When the prompting * appears, dismount the system disk.

4. Mount the source disk, WRITE PROTECTED, in place of the system disk.

5. Transfer the desired files to the backup device (such as magtape or cassette) using CSI level commands.

6. Remount the system disk.

7. Type CTRL/C to return to the monitor.

With the USR resident and the handler loaded, you do not need the system disk mounted if you do not reference any other devices and if you do not use any monitor level commands.

For example, to transfer RT-11 sources from the source disk to magtape on a single-disk system:

1. Boot system and enter the date.

2. Type: `SET USR NOSWAP` RET

   Response: `.`

   Type: `LOAD MT` RET

   Response: `.`

   Type: `INITIALIZE MT0:` RET

   Response: `MT0:/Init are you sure?`

   Type: `Y` RET

   Response: `.`

   Type: `R PIP` RET

   Response: `*`

3. Dismount system disk, mount source disk.

4. Type: `MT0:*.*/M:-1=*.*` RET

5. When done, remount system disk and use CTRL/C to return to the monitor.

2.4.17 Changing the CSR Addresses Used by the FORMAT Utility

The FORMAT utility program uses the standard CSR addresses to access devices on the system. If any devices that FORMAT supports are installed at non-standard addresses, you must patch the FORMAT utility to change these addresses. Use the following patch:

```
.R PATCH RET

FILE NAME--
*FORMAT.SAV RET
*CSRydd/ xxxxxx nnnnnn RET
*E
```

where:

y must be 1 to indicate the first or only controller; must be 2 to indicate the second controller.

dd is the two-letter device mnemonic.

nnnnnn is the correct CSR address for the controller and device specified.

NOTE

See the RT-11 System Release Notes, Table 4-2, for the exact locations to patch.

2.4.18 Using CAPS-11 to Load Files

CAPS-11 cassette file headers differ from RT-11 cassette file headers. When you are transferring files to cassette that are to be loaded through the CAPS-11 CABLDR or through 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. Make the following patch to cause all header records to contain a level byte of 0 when transferring from an RT-11 device to cassette.

1. Patch to CT.SYS:

```
.R PATCH (RET)

FILE NAME--
*CT.SYS (RET)
*4461      1      0 (RET)
*E
```

2. If you intend to build your own system via SYSGEN, edit CT.MAC as follows:

```
.EDIT CT.MAC (RET)
*R (ESC) FLEVEL: (ESC) (ESC)
*G1 (ESC) =CO (ESC) (ESC)
*EX (ESC) (ESC)
```

Install the preceding patch only if you will use CAPS-11 to load RT-11 files.

2.4.19 BATCH Patches

If you do not have a large amount of disk space on your system device, you will be interested in the following patches to BATCH. These patches allow you to store certain system programs on DK: rather than on SY: and to have BATCH access them there. These patches modify BATCH so that it invokes the corresponding system programs with the keyboard monitor RUN command (which assumes DK: as the default device) rather than with the R command (which always uses SY:).

You can make any or all of the patches. To use these patches, simply make the patches to BATCH for the system programs that you wish to remove from your system device. Copy the programs for which you installed patches to the device on which you want them to reside, then delete them from your system device (SY:). Finally, use the ASSIGN command to assign the logical name DK: to the device to which you copied the system programs. You can then run BATCH as usual. The .CTL file that BATCH generates will also reside on DK: and will contain RU commands instead of R commands for the desired system programs.

For example, to make FORTRA.SAV reside on DX1: in a dual diskette system, use the following commands:

```
.R PATCH RET
*BATCH.SAV RET
*26554;1R RET
*1,1066\ 40 125 RET
*E

.COPY SY:FORTRA.SAV DX1:*.* RET

.DELETE SY:FORTRA.SAV RET

Files deleted:
SY:FORTRA.SAV?Y RET

.ASSIGN DX1 DK RET
```

Keep in mind that, 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 also have to modify your BATCH jobs to explicitly reference certain files on SY: since that is no longer the same device as DK:. You may choose to keep your .BAT and .CTL files on SY: by invoking BATCH as follows:

```
.R BATCH RET
*SY:MYJOB=SY:MYJOB RET
```

Make the following patches to BATCH to change the default storage volume for system programs.

1. FORTRAN

```
.R PATCH RET
*BATCH.SAV RET
*26554;1R RET
*1,1066\      40      125 RET
*E
```

2. MACRO

```
.R PATCH RET
*BATCH.SAV RET
*26554;1R RET
*1,1055\      40        125 RET
*E
```

3. BASIC

```
.R PATCH RET
*BATCH.SAV RET
*26554;1R RET
*1,1137\      40        125 RET
*E
```

4. Linker

```
.R PATCH RET
*BATCH.SAV RET
*26554;1R RET
*1,1105\      40        125 RET
*E
```

5. PIP

```
.R PATCH RET
*BATCH.SAV RET
*26554;1R RET
*1,1120\      40        125 RET
*E
```

6. DIR

```
.R PATCH RET
*BATCH.SAV RET
*26554;1R RET
*1,450\       40        125 RET
*E
```

2.5 SIMPLE DEMONSTRATION PACKAGE

Once you have built, customized, and preserved the working system as described in the previous sections, you can execute the following demonstration exercise. The demonstration exercise does not serve as a comprehensive system exercise, but by utilizing several major system components, it does serve as a minimal integrity check.

To execute this demonstration, the working system must include at least the following components:

SWAP.SYS
monitor
TT.SYS
EDIT.SAV
MACRO.SAV
SYSMAC.SML
LINK.SAV
PIP.SAV
DUP.SAV
DIR.SAV
DEMOBG.MAC
DEMOFG.MAC

2.5.1 Running the RT-11 Single-Job Monitor

For purposes of this demonstration, you edit, assemble, link, and execute a small program (DEMOBG.MAC). When you execute it, DEMOBG displays a message on the terminal.

1. If the system is not running, bootstrap the system. Ensure that the system device is WRITE ENABLED.

   Response: System identifier, followed by any startup file commands.

   Type: DATE dd-mmm-yy RET

   where dd-mmm-yy is the current date in the form 12-MAY-78.

   Response: .

   If the system configuration does not include a VT11 or VS60 display processor and scope or does not have more than 8K words of memory, go to Step 2. Otherwise, verify that the scope is on by turning the BRIGHTNESS knob to an adequate level.

   Type: GT ON RET

   The system output will shift to the display scope. You still enter commands at the keyboard, but the echo is on the screen.

   Response: . (on screen)

2. If the system volume is diskette, go to Step 3. If the system device is RF11, RL01, RK06/07, RK11, RP02, or RJS03/4 disk, go to Step 4.

   If the system volume is DECtape, mount a blank tape on Unit 1, WRITE ENABLED.

   Type: ASSIGN DT1 DK RET

   Response: .

   Type: INITIALIZE DT1: RET

   Response: DT1:/Init are you sure?

   Type: Y RET

   Response: .

   Go to Step 4.

3. If the system volume is RX01 or RX02 diskette, mount a blank diskette on Unit 1, WRITE ENABLED. In the example below, xx represents the physical device name: DX for RX01 diskettes, DY for RX02.

   Type: ASSIGN xx1 DK RET

   Response: .

   Type: INITIALIZE xx1: RET

Response: `xx1:/Init are you sure?`

Type: `Y` RET

Response: `.`

4. Display the directory of the system volume on the terminal.

   Type: `DIRECTORY/BRIEF/COLUMNS:1 SY:` RET

   Response:

```
dd-mmm-yy
DT     .SYS
LP     .SYS
PP     .SYS
PR     .SYS
TT     .SYS
PATCH .SAV
EDIT   .SAV
MACRO .SAV
SYSMAC.SML
LINK   .SAV
ODT    .OBJ
PIP    .SAV
.
.
.
xxx Files, bbb Blocks
fff Free blocks
.
```

NOTE

> Depending on the medium from which you build RT-11 and the individual system configuration, the preceding directory will vary. As long as a directory is printed, you need not worry about its exact content.

5. Use the text editor to modify the demonstration program, DEMOBG.MAC.

   Type: `EDIT SY:DEMOBG.MAC` RET

   Response: `*`

   Type: `F;` TAB `.ASCII` ESC ESC

   Response: `*`

   Type: `OAD` ESC ESC

   Response: `*`

   Type: `EX` ESC ESC

   Response: `.`

6. If a line printer is available, assure that it is on line and ready.

Type: ASSIGN LP LST (RET)

Response: .

If a line printer is not available, type:

ASSIGN TT LST (RET)

Response: .

7. Assemble DEMOBG.MAC and obtain a listing.

If running on a system with 12K or more words of memory

Type: MACRO/LIST:LST: SY:DEMOBG (RET)

Proceed to step 7a for the response.

If running on an 8K system

Type: MACRO SY:DEMOBG (RET)

Response: ERRORS DETECTED: 0
.

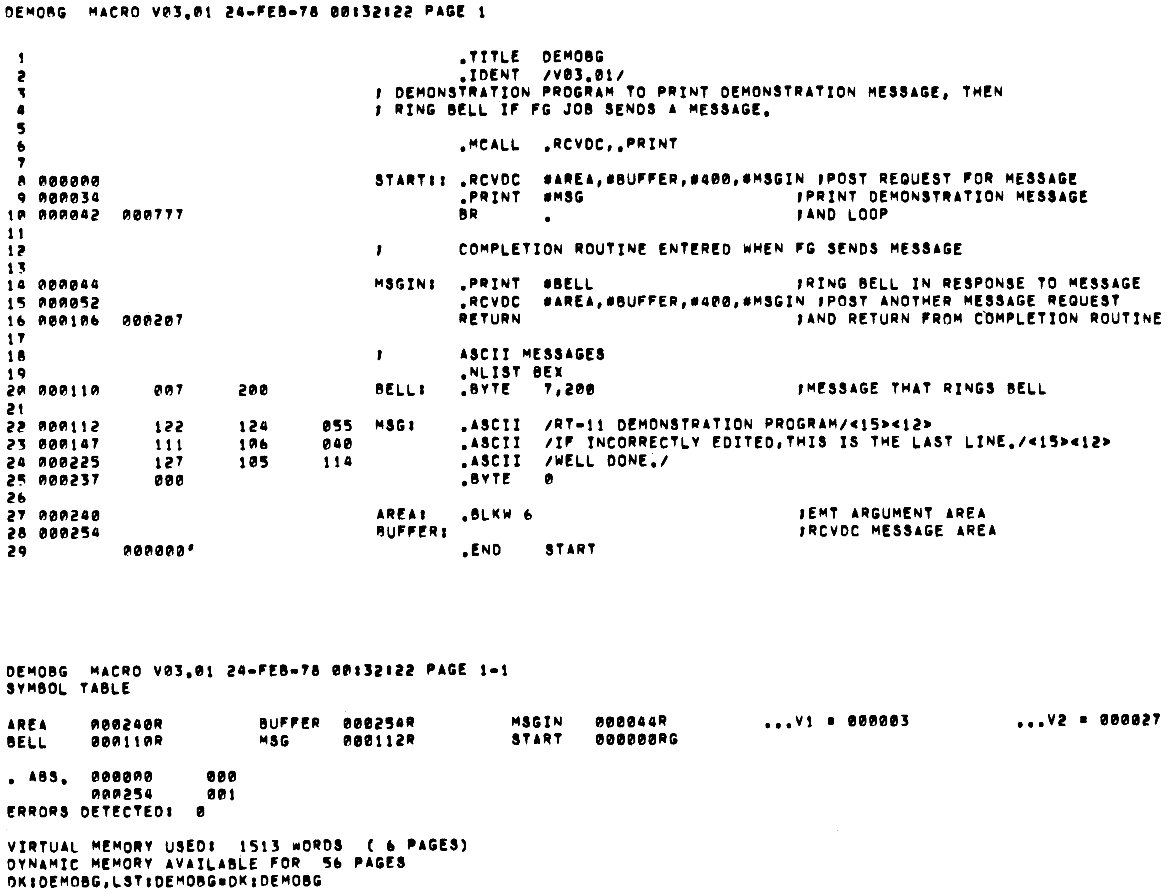
If the response is a monitor error, the most common cause is failure to rename MAC8K into MACRO.SAV when building the working system.

Type: MACRO/LIST:DEMOBG/NOOBJECT SY:DEMOBG (RET)

Response: ERRORS DETECTED: 0
.

Type: COPY DEMOBG.LST LST: (RET)

7a. Response: (see following)



```
DEMOBG  MACRO V03.01 24-FEB-78 00:32:22 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 V03.01 24-FEB-78 00:32:22 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:  1513 WORDS  ( 6 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  56 PAGES
DK:DEMOBG,LST:DEMOBG=DK:DEMOBG
```

8. Link and execute the program DEMOBG.

Type: `LINK DEMOBG` (RET)

Response:

```
.
```

Type: `RUN DEMOBG` (RET)

Response:

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

Type:

(CTRL/C)
(CTRL/C)

Response:

```
^C
^C

.
.
```

If the system has a graphics display,

Type: `GT OFF` (RET)

Response: `.` (Output will resume on the terminal.)

If you incorrectly edited the file, you may repeat the procedure although this is not necessary for successful continuation. If repeating the editing procedure:

Type: RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC RET

Response: .

Then go to Step 5.

Otherwise, if the system will be running the foreground/background monitor, proceed to Section 2.5.2.

2.5.2 Running the RT-11 Foreground/Background Monitor

For the purposes of this demonstration, you assemble a second program (DEMOFG.MAC), link it for the foreground, and execute it in conjunction with DEMOBG. This portion of the demonstration requires 16K words of memory and a clock to run. 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 Section 2.5.1, DEMOBG recognizes these messages and rings the bell once for each message sent.

Although DEMOFG is always active, sending messages to the backbround every two seconds, this demonstration will execute other programs in the background besides DEMOBG. Only when DEMOBG is active, however, is the circuit complete and messages successfully received and honored. During those periods when DEMOBG is not running, DEMOFG will enter the messages in the monitor message queue. Once you restart DEMOBG in the background, the system immediately dequeues 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.

1. To the running RT-11 single-job monitor:

   Type: BOOT xxMNFB.SYS RET

   where xx is the system device name.

   Response: RT-11FB V03B.xx

   (followed by any commands executed from the startup command file).

   .

   The FB monitor is now running.

   If the system configuration does not include a VT11 or VS60 display processor and scope, go to Step 2. Otherwise, verify that the scope is on by turning the BRIGHTNESS knob to an adequate level.

   Type: GT ON RET

   The system output will shift to the display scope. You still enter commands at the keyboard, but the echo will be on the screen.

   Response: . (on screen)

2. If the system volume is diskette, go to Step 3. If the system volume is RK06/07, RF11, RK11, RL01, RJS03/4, or RP02, go to Step 4.

   If the system volume is DECtape, mount a blank, formatted DECtape on Unit 1, WRITE ENABLED.

```
Type:          ASSIGN DT1 DK (RET)

Response:      .

Type:          INITIALIZE DT1: (RET)

Response:      DT1:/Init are you sure?

Type:          Y (RET)

Response:      .
```

   Go to Step 4.

3. If the system volume is RX01 or RX02 diskette, mount a blank diskette on Unit 1, WRITE ENABLED. In the example below, xx represents the physical device name: DX for RX01 diskette, DY for RX02.

```
Type:          ASSIGN xx1 DK (RET)

Response:      .

Type:          INITIALIZE xx1: (RET)

Response:      xx1:/Init are you sure?

Type:          Y (RET)

Response:      .
```

4. Enter the time of day.

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

   where hh:mm:ss is the current hour, minutes, and seconds in the form 13:12:00 (1:12 p.m.).

```
Response:      .
```

   Assemble the foreground demonstration program, DEMOFG.MAC.

```
Type:          MACRO SY:DEMOFG (RET)

Response:      ERRORS DETECTED:  0
               .
```

   Link DEMOFG for the foreground.

```
Type:          LINK/FOREGROUND DEMOFG (RET)

Response:      .
```

   Start DEMOFG as the foreground job.

```
Type:          FRUN DEMOFG (RET)

Response:      F>
```

```
FOREGROUND DEMONSTRATION PROGRAM, SENDS A
MESSAGE TO THE BACKGROUND PROGRAM "DEMOBG"
EVERY 2 SECONDS, TELLING IT TO RING THE
BELL.
B>
```

DEMOFG is now running and queueing the message for DEMOBG every two seconds.

Execute DEMOBG in the background and receive the messages.

Type:

```
RUN DEMOBG RET
```

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

Response:

```
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.

Type:

```
CTRL/C
CTRL/C
```

(The bell will stop.)

Response:

```
^C
^C
.
```

Type:

```
DIRECTORY RET
```

Response:

```
dd-mmm-yy
```

(The directory of the device DK: prints on the terminal. Its exact contents are not of consequence.)

```
.
```

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

Type:

```
RUN DEMOBG RET
```

(The bell will ring several times in rapid succession, then will begin ringing once every two seconds.)

Response:

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

Type:

```
CTRL/C
CTRL/C
```

Response:

```
^C
^C
.
```

(The bell will stop ringing.)

Stop the foreground program and remove it from memory.

Type:

```
CTRL/F
```

Response:

```
F>
```

Type:

```
CTRL/C
CTRL/C
```

Response:

```
^C
^C
B>
```

Type:

```
UNLOAD FG RET
```

Response:

```
.
```

If the configuration includes a VT11 or VS60 graphics display,

Type:

```
GT OFF RET
```

Response:

```
.
```

This concludes the demonstration exercise.

CHAPTER 3

SYSGEN

3.1 INTRODUCTION TO SYSGEN

Most RT-11 users can use the system components as distributed and do not need to generate a unique monitor. For those of you with unique requirements, however, RT-11 has a system generation facility that allows you to set the parameters for the system to be generated, then assemble and link a monitor and device drivers using those parameters.

RT-11 system generation requires significant mass storage and execution time. If one of the distributed monitors meets your needs, DIGITAL recommends that you do not undertake a SYSGEN. You should weigh the economies of creating a new monitor unique to the installation against the costs of doing so.

If you are unfamiliar with RT-11 you should not undertake a system generation. You must be comfortable with RT-11 commands and operating characteristics to understand the system generation process and to successfully deal with error conditions that can arise.

Although a knowledgeable user can accomplish a system generation with a smaller configuration, supported use of SYSGEN requires at least 16K words of memory and 2000 blocks of disk storage. The minimum recommended configuration for SYSGEN is a machine with at least two disk drives, 24K words of memory and an 11/34 or larger processor.

Appendix A describes a procedure for undertaking a SYSGEN on a smaller system with two RX01 drives and no other disks.

3.2 CATEGORIES OF CHOICE

The RT-11 system generation procedure allows you to customize the monitor and device handlers. Note that system generation does not impact the utility programs or any other non-executive system components.

When selecting the options available in the generated system, you have choices in two categories: (1) which services to include in the monitor and (2) which peripherals to support in the resulting system. The dialogue also allows you to specify peculiarities of the hardware configuration on which the resulting system runs.

3.2.1 Monitor Services

The following options concerning monitor services are available through SYSGEN.

1. Which monitors to generate
2. Which system device (all monitors)
3. Inclusion of timer support (SJ)
4. Inclusion of device time-out support
5. Error messages on system I/O errors (SJ)
6. Inclusion of the idle loop light pattern (FB and XM)
7. Inclusion of multiple terminal support (FB and XM)
8. 50-cycle or 60-cycle clock rate (all monitors)
9. Inclusion of startup indirect file support (all monitors)
10. Error messages on power failure (all monitors)
11. Inclusion of BATCH support (all monitors)
12. Escape sequence processing in the terminal service (all monitors)
13. Inclusion of error logging (all monitors)

3.2.2 Peripheral Support

The following options concerning peripheral support are available through SYSGEN.

1. Which devices are to be supported
2. For RF11 disks, the number of disk platters
3. For diskettes, a second controller for units 2 and 3
4. For RX02 diskettes, double-density only support
5. For RL01 disks, the number of units
6. For magtape support, file-structured or hardware support
7. The number of empty device slots
8. For FB or XM multi-terminal support, the number of interfaces of each type, and the number of local and remote lines
9. For FB or XM multi-terminal support, the size of ring buffers, inclusion of asynchronous terminal status and multi-terminal time-out support

3.2.3 Configuration Information

During SYSGEN, you also supply the following hardware configuration information.

1. Which clock to use for the system clock

2. Nonstandard interrupt vector addresses for LP11, VT11/VS60, or second RX11 or RX211 support

3. For FB or XM multi-terminal support, the interrupt vector and CSR (device Control/Status Register) addresses for each interface

3.3 FLOW OF THE SYSGEN PROCESS

The system generation process consists of three basic steps: the planning necessary to determine the desired configuration, execution of the SYSGEN program to generate conditional and command files, and actual assembling and linking of the system using the SYSGEN program output.

3.3.1 Planning

You should not sit down at a system generation session without first doing the planning and information gathering necessary to enable you to intelligently respond to the SYSGEN dialogue.

The items of information you need about the target configuration include:

- A list of the peripheral devices on the target machine

- The CSR and vector addresses for any nonstandard peripheral interfaces, especially extra terminals and graphics devices

- A list of the monitor services required for the target application

You should also have read and understood the entire SYSGEN dialogue described in Section 3.4. Understanding this dialogue and the implications of the various responses is important to making intelligent choices.

3.3.2 Execution of the SYSGEN Program

Once you complete the planning, you then execute the system generation dialogue provided through the SYSGEN program. SYSGEN asks you a series of questions and uses your responses to direct further questions and eventually to construct five output files.

The first two of these output files are SYCND.MAC and SYSTBL.MAC. These files are the conditional files that are assembled with the standard monitor components to generate the customized system.

The last three output files are MONBLD.COM, DEVBLD.COM and SYSBLD.COM. These are indirect command files that can be used to assemble and link the desired system without further user intervention. MONBLD.COM assembles and links the monitor(s) and also produces link maps. DEVBLD.COM assembles and links the device handlers, and SYSBLD.COM simply invokes the other two.

The SYSGEN program requires two script files as input: SYSGEN.CND and SYSTBL.CND. Both files must be present on DK: when you execute SYSGEN.

SYSGEN also requires sufficient space on DK: for the conditional and command files that it creates.

3.3.3 Assembly and Linking of System

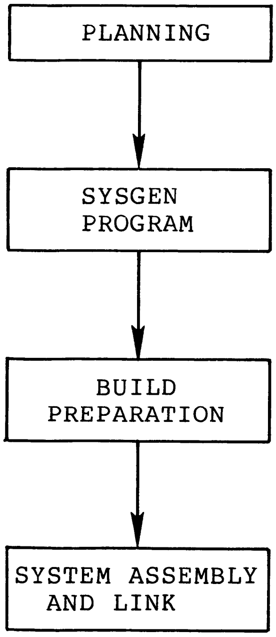
Once you generate the conditional and command files with SYSGEN, the final phase of the process is to use these files to assemble and link the new monitors and handlers.

The system sources (without comments) necessary to this step are all distributed on the binary kit. You can assemble these sources using the standard command files output by SYSGEN, or, if space does not permit, you can substitute equivalent commands by hand.

See Section 3.6 for the details of the assembly and link process.

3.3.4 Summary of SYSGEN Process

The flow chart in Figure 3-1 summarizes the steps necessary in the SYSGEN process. Figure 3-2 illustrates the files that SYSGEN requires.



Step 1: Collect necessary information for dialogue.

Step 2: Execute SYSGEN dialogue; output of conditional and indirect command files for building the system.

Step 3: Collect appropriate files on appropriate media.

Step 4: Create new system.

Figure 3-1 System Generation Process

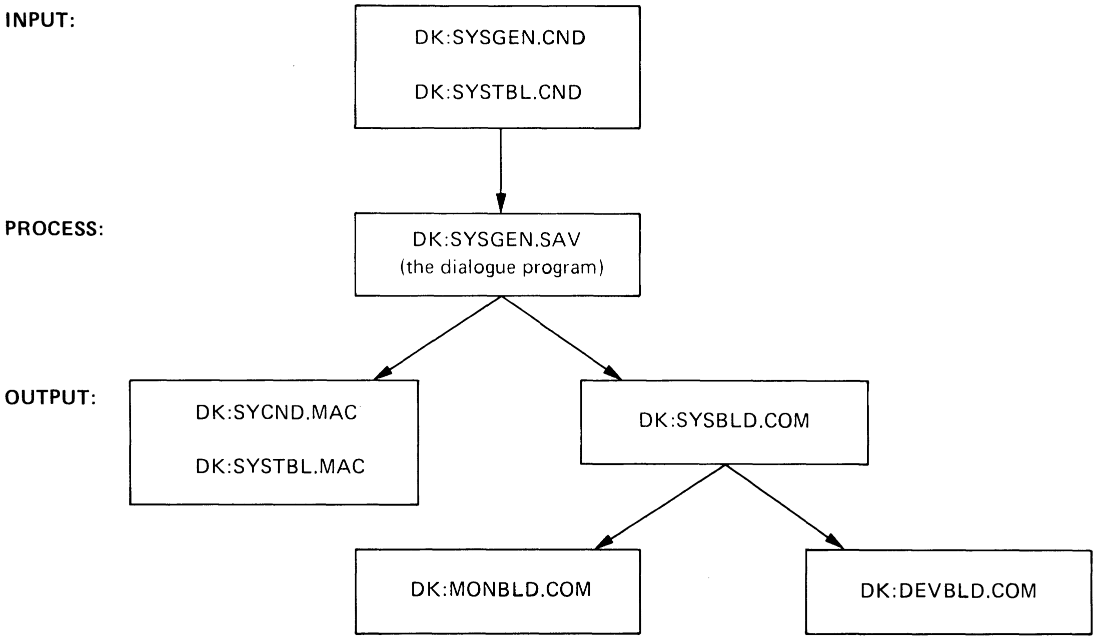


Figure 3-2 Files Required for System Generation Process

3.4 THE SYSGEN DIALOGUE

This section describes the SYSGEN dialogue in detail. It describes all text, questions and responses displayed during the dialogue. However, during any single run of the SYSGEN dialogue, many of these questions and comments do not appear since they depend on the answers to previous questions. Appendix A contains a console output listing of complete, actual SYSGEN runs.

The SYSGEN dialogue has two forms. The long form precedes each question with an explanatory text paragraph; the short form limits the dialogue to the questions themselves, providing no explanatory information. If you have run SYSGEN many times and understand each question, you will find the short form saves time. If you are still unfamiliar with the dialogue, the long form is helpful. Which form to use is the subject of the first question in the dialogue, and the response to this question governs the remainder of the session.

3.4.1 Question Types and Defaults

Each question that the SYSGEN dialogue presents has an associated default response. After the program asks each question, you can respond by typing the desired response or by typing RET to select the default. You can edit inputs using the standard RT-11 editing techniques (DELETE and CTRL/U).

There are three types of questions in the SYSGEN dialogue.

Questions that require a choice are in the following form:

```
     Question [Y/N](default)?
```

You can respond by typing Y RET for YES, N RET for NO, or just RET to select the default value, which is shown in parentheses.

Questions that require a numeric response are marked with [O] to signal that the response is an octal number or [D] for a decimal number, followed by the default value indicated in parentheses.

```
     Question [O] (nnnnn)?
              [D]
```

Respond by typing the desired value or by typing RET to select the default.

Questions that require an alphanumeric device name as a response are marked [DD] (where only the name is required) or [DDU] (where name and unit are required), followed by the default name indicated in parentheses.

```
     Question [DD] (xxn)?
              [DDU]
```

You can respond by typing the desired device name (and unit if called for) followed by RET or by typing just RET to select the default device.

The program checks the responses you type for validity during this phase of the SYSGEN process. If you type an incorrect response (such as a numeric response that exceeds the highest or lowest acceptable response), SYSGEN prints an appropriate error message and repeats the question. You should then provide the correct response.

3.4.2 SYSGEN Dialogue Explanation

The following is the actual SYSGEN dialogue interspersed with explanatory text. Note that when you run SYSGEN, you do not see all the dialogue shown in this section; this section contains all the possible questions.

If you are about to run SYSGEN for the first time, you should use the information in this section to guide your responses.

To invoke the SYSGEN program, respond to the monitor's prompt (.) by typing:

```
     R SYSGEN RET

     RT-11 SYSTEM GENERATION PROGRAM V02.11

     Do you want the expanded form of dialogue [Y/N] (Y)?
```

The expanded form precedes each question with explanatory information. If you are thoroughly familiar with SYSGEN, answering NO saves time; if you are not familiar with SYSGEN, or are in doubt, use the expanded form.

```
All answers requiring a choice are indicated by [Y/N]
and require you to type either 'Y' for yes or 'N'
for no followed by a carriage return.  Answers requiring
numeric input are indicated by [D] for decimal input and
[O] for octal input.  Answers requiring a device name
response are indicated by [DD]; answers requiring a
device name followed by a unit number are indicated
by [DDU]. If you enter no response (just a CARRIAGE RETURN),
your answer will default to the response indicated in
parentheses before the question mark (e.g. [DD] (RK)?).
Your responses are checked for validity wherever possible.

The outputs of this SYSGEN program are two conditional
files (SYCND.MAC and SYSTBL.MAC) and three indirect
command files (SYSBLD.COM, MONBLD.COM and DEVBLD.COM)
which are used to assemble and link the monitor(s) and
device handlers for the target system. The assembly
and linking process requires a machine 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.

The following dialogue asks questions about
the configuration for which the system is being
generated, and about monitor and device support
options you may desire.  You should have read the
RT-11 System Generation Manual before proceeding.
You can terminate SYSGEN at any time by typing CTRL/C.

The RT-11 system includes a variety of monitors, allowing
you to select the monitor best suited to your
application.  The monitors are customized to your configuration
by this SYSGEN process, and, while you may build any number of
monitors, you should realize that SYSGEN places a large
demand on time and data storage.

If you elect to build the baseline
single-job (SJ) monitor, no other monitors
may be built during the same SYSGEN pass.

The baseline single-job (SJ) monitor is the minimal monitor
necessary to execute standard system functions. It is intended
for systems with only 8K words of memory, or for larger
systems that require the lowest possible monitor overhead.
Device support is limited, and some features of the
SJ monitor usable in larger configurations are omitted,
such as BATCH and VT11 support.

Do you want the baseline single-job monitor [Y/N] (N)?
```

If you want to generate a base-line SJ monitor, as opposed to a standard SJ monitor, or an FB or XM monitor, respond YES; if not, respond NO. Respond YES only if you are building DECtape, RX01 flexible diskette or RK05 based monitor. RT-11 does not support base-line monitors for other system devices. If you do select the base-line monitor, the conditional and command files output by this SYSGEN run can build only the base-line monitor. Creating additional monitors requires additional SYSGEN runs. The default is NO.

```
The single-job (SJ) monitor provides an environment suitable
for the development of simple applications in FORTRAN or
BASIC.  It provides a subset of the foreground/background
(FB) features and offers maximum ability for minimum size.
Some additional features may be added during SYSGEN, but at
the cost of increased monitor size.

Do you want the single-job (SJ) monitor [Y/N] (Y)?
```

Respond YES if you want to build a standard single-job monitor, NO if you do not. A response of YES here does not preclude also creating an FB and an XM monitor on the same run. The default is YES.

```
The foreground/background (FB) monitor offers an extended
set of facilities which are helpful for applications
involving real-time or data acquisition. These
include timer support, serialized asynchronous I/O, and
support for a foreground job.

Do you want the foreground/background (FB) monitor [Y/N] (Y)?
```

If you want an FB monitor answer YES; if not, answer NO. Neither response affects the ability to build an SJ or an XM monitor on this run. The default is YES.

```
The extended memory (XM) monitor is a version of the FB
monitor which supports up to 128K words of memory. It is
useful for advanced applications that require an extended
addressing space.  The XM monitor is larger than FB and
requires a resident USR and drivers.

Do you want the extended memory (XM) monitor [Y/N] (N)?
```

If you want to build an XM monitor, respond YES; if not, answer NO. The response to this question does not affect the building of SJ or FB monitors. The default is NO.

Note that you must respond YES to at least one of the preceding four questions; that is, each SYSGEN run must build at least one monitor. Because the system must build the monitor and handlers according to the same parameters, you cannot rebuild device support without rebuilding the corresponding monitors. If you respond NO to all four monitor selections, SYSGEN prints an error message and terminates the session. If this happens, you must restart SYSGEN from the beginning.

```
There are a number of monitor options you may select:

The following options apply only to the SJ monitor.

The SJ monitor normally does not have timer support.
You may optionally include the .MRKT and .CMKT requests.

Do you want timer support in the SJ monitor [Y/N] (N)?
```

The single-job monitor does not normally support the .MRKT and .CMKT programmed requests. If you want to include support for these requests in the SJ monitor, respond YES. The default is NO.

Including timer support in the SJ monitor adds approximately 200 words to the resident monitor, and the additional interrupt-level overhead reduces performance slightly. Timer support is only necessary if your application requires it; the RT-11 system itself does not. This option applies only to the SJ monitor.

```
The device time-out option permits device drives to issue
mark time requests.  This feature is currently not used
by any RT-11 drivers, but is necessary for DECNET support.
Select this option if you are planning to use DECNET
with your monitor.

Do you want device time-out support [Y/N] (N)?
```

If you want device time-out support, answer YES; if not, respond with NO. Device time-out support is required for DECNET applications. The default is NO.

```
The SJ monitor will normally halt at the "system halt"
if there is a fatal system I/O error. This halt can
be replaced with a system error message at the cost of
about 30 words additional code in the resident
monitor. This option is highly recommended if the system
is going to be used by anyone other than the expert.

Do you want an error message on system I/O errors [Y/N] (Y)?
```

If you want an error message instead of a system halt when the SJ monitor detects a fatal I/O error, respond YES. The default is YES. This option is highly recommended for all but the most space-conscious applications; the confusion it saves is well worth the small incremental memory requirement. This option applies only to the SJ monitor.

```
The following options apply only to the FB and XM monitors.

The FB and XM monitors' scheduler idle loop will display
a moving light pattern in the display register of any
PDP-11/45, 11/50, 11/55 or 11/70 processor.

Do you want idle loop light pattern [Y/N] (N)?
```

The FB and XM idle loop display is useful as an indicator of how "busy" the central processor is. If you want this display, answer YES. The default is NO. Selecting the display adds approximately 25 words to the resident monitor. This option applies only to the FB and XM monitors.

```
RT-11 normally supports a single terminal interfaced through
the console KL11 or DL11, which is shared by both background
and foreground jobs. You may optionally select multi-
terminal support of up to 16 terminals interfaced through
a choice of DL11 and DZ11 interfaces.

Do you want support of more than one terminal [Y/N] (N)?
```

Respond YES if you want monitor support of more than one terminal. The default is NO. Respond NO if the application itself provides the support for the additional terminals (such as MU BASIC V01) and you require only console support in the monitor. Multi-terminal support adds considerably to the space requirements of the resident monitor: a minimum of 750 words for the first additional terminal, and approximately 100 words for each additional terminal thereafter. These numbers are highly variable, depending on interfaces, configuration and other options elected (such as escape sequence support). This option applies only to the FB and XM monitors.

```
The following options apply to all monitors.

RT-11 assumes a 60 Hz power line frequency, but can
optionally support a 50 Hz frequency.

Do you want the optional 50 Hz clock support [Y/N] (N)?
```

If the line frequency is 50 Hz, respond YES. The default is NO (60Hz). Note that this response and all succeeding responses apply to all monitors being generated.

```
RT-11 normally uses a line clock for the system clock.
You may substitute the KW11-P programmable clock if you wish,
in which case the KW11-P is not available for program use.

Do you want to use the KW11-P clock as the system clock [Y/N] (N)?
```

Respond YES if you want to use the KW11-P programmable clock instead of the line clock as the system clock. The default is NO (the system clock is the line clock). The system clock is used for time-of-day calculations, timer requests, and scheduling purposes in the FB and XM monitor. Note that if the monitor is using the programmable clock, an application cannot access the clock directly or change its rate without affecting monitor function. If a line clock is available, DIGITAL recommends that you use it as the system clock.

```
The bootstrap optionally executes an indirect file when
starting the system.

Do you want the startup indirect file [Y/N] (Y)?
```

If you want the monitor bootstraps to execute the startup indirect file STARTx.COM every time that you boot the system, respond YES. The default is YES. This option requires no additional resident monitor space. If you elect it, the monitor prints an error message each time that you bootstrap it and no command file is present.

```
RT-11 supports parity memory and processes memory
parity errors. If error logging is present, parity errors
are also logged.

Do you want memory parity support [Y/N] (N)?
```

If you want parity memory support, respond YES. The default is NO. If you elect memory parity support, the system detects, logs and reports memory and cache parity errors. Selection of memory parity support adds about six words of memory to the resident monitor if you do not also select error logging, and about 25 words if you do select error logging.

```
The monitor normally halts on power recovery startup
after a power failure. You may elect to have the monitor
print a message explaining that power failure caused the halt.

Do you want power failure messages [Y/N] (Y)?
```

The RT-11 monitors normally halt after a power failure. If you prefer an error message on startup to report the cause of the halt, respond YES. The default is YES. Election of this option adds about 25 words of memory to the resident monitors. This option is highly recommended for all but the most core-conscious of applications, because transient power failures that are too brief to detect visually are a frequent cause of spurious system crashes. This message helps to identify the problem quickly. The target configuration must have core memory or battery back-up for this option to function correctly; do not select this option if you have volatile semiconductor memory.

```
     RT-11 supports BATCH as an option. Indirect command
     file support is separately available in all monitors,
     and BATCH support need not be selected to obtain
     indirect command file support.

     Do you want BATCH support [Y/N] (N)?
```

RT-11 BATCH support is distinct from indirect file support; if you want to run RT-11 BATCH streams in addition to indirect files, respond YES. The default is NO. BATCH offers several advantages over indirect files (file-based logs, job-stream programmability, operator interaction), and electing it adds 20 words to the resident monitor. If you do not specifically need the features of RT-11 BATCH (that is, if indirect files will serve the purpose), DIGITAL does not recommend selecting this option.

```
     The RT-11 terminal service is capable of recognizing
     ANSI or DEC standard escape sequences and processing
     those sequences according to application instructions.
     This support is not necessary for normal system operation;
     if your application does not make special use of escape
     sequences, do not select this support.

     Do you want escape sequence processing [Y/N] (N)?

     Do you want ANSI standard escape sequences [Y/N] (N)?

     DEC standard escape sequences assumed.
```

RT-11 terminal service normally treats ESCAPE the same as all other characters. It is buffered, transmitted, and echoed (as a dollar sign) according to normal system rules. If your terminal uses ESCAPE sequences to perform command or other asynchronous functions, and your application uses these features, the monitor optionally recognizes the sequence and processes it according to various flags set by the application.

If your terminal does not generate escape sequences, or your terminal uses them but your application does not, respond NO to the first question.

If your terminal generates escape sequences and your application will use them, respond YES to the first question. NO is the default.

The second question allows you to specify whether you want ANSI or DEC standard sequences. Very few terminals conform to the ANSI standard; if yours does, respond YES. NO is the default.

Electing escape sequence support adds approximately 210 words to the resident monitor; it is not necessary for operation of any of the RT-11 system programs.

```
     Selection of error logging creates the error logging
     driver EL and incorporates error logging into the device
     handlers. When EL is loaded and initialized, logging of
     disk, DECtape and memory parity errors will occur.

     Do you want error logging [Y/N] (N)?
```

If you want to use the error logger on the target configuration, respond YES. The default is NO. Selecting error log support adds approximately 1000 words to the resident monitor when EL is loaded (depending on the options you select) and approximately 50 words to each of the device handlers in the target system. Error log support is not required for your RT-11 system, but it is useful in monitoring system reliability.

```
The error logger can capture either initial errors
or the most recent errors that have occured. For most
types of problems, information concerning initial
errors is most useful.

Should the error logger capture the most recent errors [Y/N] (N)?
```

The error logger can use its error buffer to capture as many errors as occur until the buffer is full, or it can treat the error buffer as a ring buffer and retain only as many of the most recent errors as possible. Initial errors are usually most useful, but a particular problem can warrant collecting the most recent errors. Respond YES if you want only the most recent errors retained. The default is NO. This option has no impact on the size of the resident monitor.

```
Normal error logging usage requires the utility ERRUTL
to be run occasionally from the console
to write the error information out to the error log file. If
an application is to be run continuously for a long period
of time, however, the error logger has a facility that
allows the application program to dump the error log buffer
to the error file at regular intervals. Unless your
application specifically uses this feature, you do not need
this support.

Do you want a program other than ERRUTL
    to write the error log buffer [Y/N] (N)?
```

Respond YES if your application has a facility that automatically dumps the error log buffer. Respond NO if your use of error log support will be limited to the facilities distributed with the system. The default is NO. Selecting this option adds approximately 180 words to the error logger. If you select this option, ERRUTL continues to operate as an alternate means of dumping the error log buffer.

```
The error logger requires a resident buffer to collect
errors. The buffer size may be increased to accomodate
an unusual error load.

How many 256 word memory blocks should be
    set aside for the error buffer [D] (1)?
```

Respond by specifying the number of blocks to be set aside for the error buffer. The default is 1. The range for valid responses is from 1 through 4. Each buffer consumes 256 words of resident memory, so you should limit the buffer size to the minimum number of blocks necessary to store the important error information.

```
The error logger can handle up to 35 individual device
units. Space can be conserved by reducing the number
of units the logger can handle to the specific number
of units present on the target configuration. Only disk
and DECtape units need be included.

How many disk and DECtape device units can be logged [D] (10)?
```

Respond by specifying the total number of disk (including diskettes) units and DECtape transports on the target system. The default is 10. If the system has three RK05 drives, a dual DECtape drive (two transports) and a dual diskette drive, the response is seven (one for each unit). Each unit adds seven words to the error logger. The range for valid responses is from 1 through 35.

```
The following questions ask about the
peripheral devices that are to be supported by the
generated system. By selecting support for a specific
device, you make the device known to the monitor's
device tables, and you cause a device handler
to be generated for the device. If you do not select
a specific device, you will have to create the
device handler separately and use the monitor
INSTALL command when device access is desired.

RT-11 Version 3B supports the following devices as system devices:

        RK  RK05 Cartridge Disk
        DT  DECtape
        DX  Flexible Diskette

        RK  RK05 Cartridge Disk
        DL  RL01 Cartridge Disk
        RF  RF11 Fixed-head Disk
        DT  DECtape
        DX  RX01 Flexible Diskette
        DY  RX02 Double Density Diskette
        DS  RJS03/4 Massbus Fixed-head Disk
        DP  RP11 Disk Pack
        DM  RK06/RK07 Cartridge Disk

You must select one as your system device; use the
two character device name as indicated above.

What is the name of the system device [DD] (RK)?
```

This query establishes the system device for the monitor being generated. Enter the two-character RT-11 physical device name for the system device, followed by a (RET) . The default is RK. Use only legal RT-11 physical device mnemonics as illustrated in the expanded comments, and do not include a unit number. The system device indicated corresponds to all the monitors built as a result of this dialogue.

For the following questions, you need only select support for the devices that appear in the target system. You must select the support option that corresponds to the system device.

```
Do you want RJS03 or RJS04 disk support [Y/N] (N)?
```

Respond YES if the target system has an RJS03 or RJS04 disk subsystem. The default is NO.

```
Do you want RJS03 support (as opposed to RJS04 support) [Y/N] (Y)?
RJS04 support assumed.
```

Respond YES if the target system has an RJS03 disk subsystem. The default is YES. If you have already named DS as the system device, you should elect this option if the disk is RJS03 (to distinguish it from RJS04).

```
Do you want RF11 fixed-head disk support [Y/N] (N)?
```

Respond YES if the target configuration has an RF11/RS11 disk subsystem. The default is NO. If you have already named RF as the system device, this question is not asked.

```
How many disk platters are installed on the RF11 controller [D] (1)?
```

Respond with the number of RS11 platters present on the RF11 subsystem, followed by RET ; 1 is the default. The range for valid responses is from 1 through 8.

```
Do you want RP11/RPR02 or RP11/RP03 disk support [Y/N] (N)?
```

Respond YES if the target configuration has an RP11/RP02 or RP11/RP03 subsystem. The default is NO. If you have already named DP as the system device, this question is not asked.

```
Do you want RPR02 support (as opposed to RPR02/RP03) [Y/N] (Y)?

RPR02 and RP03 support assumed.
```

Respond YES if the target configuration has an RP11/RP02 subsystem. The default is YES.

```
Do you want RX01/RX11 flexible diskette support [Y/N] (N)?
```

Respond YES if the target system has an RX11/RX01 flexible diskette subsystem. The default is NO. If you have already named DX as the system device, this question is not asked.

```
The RX11 flexible diskette subsystem supports only two
drives for each controller. If your system contains four
diskette drives, there is a second controller present
supporting the third and fourth units.
Do you want support for a second RX11 controller [Y/N] (N)?
```

If the target system has four RX01 flexible diskette drives, respond YES. If it has only two drives, respond NO. The default is NO.

```
What is the CSR address for the
    first RX11 controller [O] (177170)
```

Respond by typing the address of the control register for the first RX11 controller, followed by RET . The default is 177170. The range for valid responses is from 160000 through 177570.

```
What is the vector address for the first RX11 controller [O] (264)?
```

Respond by typing the vector address of the first RX11, followed by RET . The default is 264. The range for valid responses is from 100 through 770.

```
What is the CSR address for the
    second RX11 controller [O] (177150)?
```

Respond by typing the address of the control register for the second RX11 controller, followed by RET . The default is 177150. The range for valid responses is from 160000 through 177570.

```
What is the vector address for the second RX11 controller [O] (270)?
```

Respond by typing the vector address of the second RX11, followed by RET. The default is 270. The range for valid responses is from 100 through 770.

```
     Do you want RX02 diskette support [Y/N] (N)?
```

Respond YES if the target configuration has an RX211/RX02 diskette subsystem. The default is NO. If you have already named DY as the system device, this question is not asked.

```
     The RX02 diskette subsystem supports only two drives for
     each controller. If your system contains four drives,
     there is a second controller present supporting the
     third and fourth units.

     Do you want support for a second RX02 controller [Y/N] (N)?
```

If the target system has four RX02 flexible diskette drives, respond YES. If it has only two drives, respond NO. The default is NO.

```
     The RX02 diskette supports both single and double densities
     by default. You can obtain a slight performance
     enhancement and a reduction in the size of the device
     handler by specifying double density support only.

     Do you want RX02 double density support only [Y/N] (N)?
```

Respond YES if you plan to use only double density diskettes. The default is NO.

```
     What is the CSR address for the
         first RX02 controller [O] (177170)?
```

Respond by typing the address of the control register for the first RX211 controller, followed by RET. The default is 177170. The range for valid responses is from 160000 through 177570.

```
     What is the vector address for the first RX02 controller [O] (264)?
```

Respond by typing the vector address of the first RX211, followed by RET. The default is 264. The range for valid responses is from 100 through 770.

```
     What is the CSR address for the
         second RX02 controller [O] (177150)?
```

Respond by typing the address of the control register for the second RX211 controller, followed by RET. The default is 177150. The range for valid responses is from 160000 through 177570.

```
     What is the vector address for the second RX02 controller [O] (270)?
```

Respond by typing the vector address of the second RX211, followed by RET. The default is 270. The range for valid responses is from 160000 through 177570.

```
     Do you want RL01 disk support [Y/N] (N)?
```

Respond YES if the target configuration has an RL11/RL01 disk subsystem. If you have already named DL as the system device, this question is not asked. The default is NO.

```
     How many RL01 units are to be supported [D] (2)?
```

Respond by typing the number of RL01 units on the target system. The default is 2. The range for valid responses is from 1 through 4. Each additional unit adds 20 words for table space, increasing the size of the handler.

```
Do you want RK05/RK05F disk support [Y/N] (N)?
```

Respond YES if the target system has an RK11/RK05/RK05F disk subsystem. The default is NO. If you have already named RK as the system device, this question is not asked.

```
Do you want TC11 DECtape support [Y/N] (N)?
```

Respond YES if the target system has an TC11/TU56 DECtape subsystem. The default is NO. If you have already named DT as the system device, this question is not asked.

```
Do you want RK06/RK07 disk support [Y/N] (N)?
```

Respond YES if the target system has an RK611/RK06/07 disk subsystem. The default is NO. If you have already named DM as the system device, this question is not asked.

```
Do you want magnetic tape support [Y/N] (N)?
```

If the target system has an industry-compatible TM11 or TJU16 magtape subsystem, respond YES. The default is NO.

```
RT-11 magtape support is available in two forms. The standard
form is file-structured magtape, which supports a subset of
ANSI file format. The file-structured handler may be used to
store and retrieve data in a file format usable with most
RT-11 system programs. The second form of magtape support
omits file structuring and reads and writes data directly in variable
length records. This type of support is called the "hardware
handler", and while this handler is significantly smaller than
the file-structured handler, it can not be used with any
system programs. The file-structured handler is also
capable of performing direct hardware functions.

Do you want the file-structured magtape handler [Y/N] (Y)?
```

Respond YES if you want the standard RT-11 magtape support. If you intend to use magtape with any of the system programs or commands, or from a FORTRAN or BASIC program, you should select the file-structured handler. The default is YES.

```
The file-structured 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.

How many magtape units are to be supported [D] (2)?
```

Respond by typing the number of drives present on the magtape subsystem followed by a RET . The default is 2. The range for valid responses is from 1 through 8.

```
Do you want TM11 (UNIBUS) magtape support [Y/N] (Y)?
```

Respond YES if the magtape controller is a TM11 family controller. The default is YES. Only if the target magtape is a MASSBUS tape (TJU16 family) should you respond NO.

```
Do you want TJU16 (MASSBUS) magtape support [Y/N] (Y)?
```

Respond YES if the magtape controller is a TJU16. If you responded NO to the previous question and you do not have a TJU16 compatible drive, an error message appears and the system generation terminates; you must restart the SYSGEN process. The default is YES.

```
Do you want TA11 cassette support [Y/N] (N)?
```

Respond YES if the target system has a TA11/TU60 cassette subsystem. The default is NO.

```
Do you want VT11 or VS60 graphics support [Y/N] (N)?
```

Respond YES if the target system includes a VT11 or VS60 graphics subsystem. The default is NO.

```
Do you want VS60 support [Y/N] (N)?
```

Respond YES if the graphics subsystem is a VS60. Respond NO otherwise. The default is NO (VT11 support is assumed).

```
VT11 hardware is assumed.

The VT11/VS60 is normally installed at vector 320 with
its CSR address at 172000, but these addresses may float.

What is the CSR address for the VT11/VS60 [O] (172000)?
```

Enter the address of the VT11 or VS60 control register, followed by a RET . The default is 172000. The range for valid responses is from 160000 through 177570.

```
What is the vector address for the VT11/VS60 [O] (320)?
```

Enter the vector address of the VT11 or VS60, followed by a RET . The default is 320. The range for valid responses is from 300 through 770.

```
Do you want line printer support [Y/N] (Y)?
```

Respond YES if the target system has a line printer; respond NO if it does not. The default is YES.

```
The line printer vector address is normally 200
and the CSR address is normally 177514. If your printer
is installed at another vector or CSR address, you
must specify the correct values.

Does your printer have a non-standard vector or CSR address [Y/N] (N)?
```

Respond YES if the printer control register address is NOT 177514 or the vector is NOT 200. This usually occurs only if the printer is an LA180S. The default is NO.

```
What is the CSR address for the printer [O] (177514)?
```

Enter the address of the printer control register followed by a RET . The default is 177514. The range for valid responses is from 160000 through 177570.

```
What is the vector address for the printer [O] (200)?
```

Enter the vector address of the printer followed by RET. The default is 200. The range for valid responses is from 100 through 770.

```
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.

Do you want PC11 high-speed paper tape reader/punch support [Y/N] (N)?
```

Respond YES if the target system has a high-speed paper tape reader and punch. If the system has a high-speed reader only, respond NO. The default is NO.

```
Do you want PR11 high-speed paper tape reader support [Y/N] (N)?
```

Respond YES if the target system has a high-speed reader. The default is NO.

```
Do you want card reader support [Y/N] (N)?
```

Respond YES if the target system has a CR11 or CM11 card reader. The default is NO.

```
The null handler is a mechanism that allows logical
input and output to take place without actual physical
data transfers. It is most useful in debugging applications.

Do you want the null handler [Y/N] (Y)?
```

Respond YES if you want the NULL handler, NL.SYS, to be available on the target system. The default is YES.

```
This completes the device query. Space in the monitor is allocated
only for the devices specified, but you may 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.

How many extra device slots do you want [D] (0)?
```

Enter the number of additional devices you plan to add to the system, followed by RET. Include any device for which a handler is to be INSTALLed in the monitor. If the device will be supported only by a particular application and not by a system handler, you need not set aside a device slot for it. The default is 0. The range for valid responses is from 0 through 10.

```
The indirect command files generated by the SYSGEN program
contain commands that will delete the object modules
from which the system is built when they are
no longer needed. This measure serves to conserve disk
space while the system is being built. It is often useful,
however, to keep the object modules for later use in
patching the system. If your output device
has sufficient free space,you can elect to retain all
system object modules for future use.

Do you want to retain the system OBJ's [Y/N] (N)?
```

If the system volume on which the system will be built has abundant mass storage (approximately 500 blocks per monitor being built) and you want to retain the system .OBJ files for later patching purposes, respond YES. The default is NO. If in doubt, respond NO. Monitor patches distributed by DIGITAL will be made available in source form.

```
The following questions ask about 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. Physical unit numbers of the DL11 lines are assigned
first to local lines and then to remote lines. This assignment
is permanent and its order cannot be changed.

After the dialogue has established the number of lines of each
type, the CSR and vector addresses for each line must be
established. The first 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 response applies to the first remote
line, the second to the second remote line, etc. until all
remote lines are accounted for.

How many local DL11's, including the console [D] (1)?
```

Enter the number of LOCAL DL11 terminals in the target configuration. This number will never be less than 1 (the console is always a local DL11), and cannot be greater than 8. Do not include remote DL11 lines in this number. The default is 1. The range for valid responses is from 1 through 8.

```
How many remote DL11 lines [D] (0)?
```

Enter the number of REMOTE DL11 terminals in the target configuration. Do not count the local lines that you specified in the previous answer. This response can never be greater than 7, and the sum of this response and the previous response must be less than or equal to 8. The default is 0.

```
What is the CSR address for the
    first (console) DL11 [O] (177560)?
```

Respond by entering the keyboard control register address for the console terminal. The default is 177560. The range for valid responses is from 160000 through 177570.

```
What is the vector address for the first (console) DL11 [O] (60)?
```

Respond by entering the keyboard vector address for the console terminal. The default is 60. The range for valid responses is from 60 through 770.

```
What is the CSR address for the next (local) DL11 [O] (176500)?
```

Respond by entering the keyboard control register address of the next local DL11 interface. The default is 176500. The range for valid responses is from 160000 through 177570.

```
What is the vector address for the next (local) DL11 [O] (300)?
```

Respond by entering the keyboard vector address for the DL11 whose keyboard control register address was specified in the preceding response. The default is 300. The range for valid responses is from 60 through 770.

For each of the succeeding questions, enter the control register and vector addresses for each succeeding DL11 interface. Enter all the local interfaces first (if any), then all the remote interfaces (if any), until you have entered all interface addresses.

For example, assume 5 DL11 lines, 3 local and 2 remote. The dialogue would look like:

```
WHAT IS THE CSR ADDRESS FOR THE FIRST (CONSOLE) DL11 [O]
 (177560) ?  RET

WHAT IS THE VECTOR ADDRESS FOR THE FIRST (CONSOLE) DL11 [O]
 (60)?  RET

WHAT IS THE CSR ADDRESS FOR THE NEXT (LOCAL) DL11 [O]
 (176500)? 2ND LOCAL ADDR RET

WHAT IS THE VECTOR ADDRESS FOR THE NEXT (LOCAL) DL11 [O]
 (300)? 2ND LOCAL VECTOR RET

WHAT IS THE CSR ADDRESS FOR THE NEXT (LOCAL) DL11 [O]
 (176510)? 3RD LOCAL ADDR RET

WHAT IS THE VECTOR ADDRESS FOR THE NEXT (LOCAL) DL11 [O]
 (310)? 3RD LOCAL VECTOR RET

WHAT IS THE CSR ADDRESS FOR THE NEXT (REMOTE) DL11 [O]
 (175630)? 1ST REMOTE ADDR RET

WHAT IS THE VECTOR ADDRESS FOR THE NEXT (REMOTE) DL11 [O]
 (320)? 1ST REMOTE VECTOR RET

WHAT IS THE CSR ADDRESS FOR THE NEXT (REMOTE) DL11 [O]
 (175640)? 2ND REMOTE ADDR RET

WHAT IS THE VECTOR ADDRESS FOR THE NEXT (REMOTE) DL11 [O]
 (330)? 2ND REMOTE VECTOR RET
```

The defaults are as shown in parentheses at the end of each question.

```
What is the CSR address for the next (remote) DL11 [0] (175610)?

What is the vector address for the next (remote) DL11 [0] (300)?

What is the CSR address for the next (local) DL11 [0] (176510)?

What is the vector address for the next (local) DL11 [0] (310)?

What is the CSR address for the next (remote) DL11 [0] (175620)?

What is the vector address for the next (remote) DL11 [0] (310)?

What is the CSR address for the next (local) DL11 [0] (176520)?

What is the vector address for the next (local) DL11 [0] (320)?

What is the CSR address for the next (remote) DL11 [0] (175630)?

What is the vector address for the next (remote) DL11 [0] (320)?

What is the CSR address for the next (local) DL11 [0] (176530)?

What is the vector address for the next (local) DL11 [0] (330)?

What is the CSR address for the next (remote) DL11 [0] (175640)?

What is the vector address for the next (remote) DL11 [0] (330)?

What is the CSR address for the next (local) DL11 [0] (176540)?

What is the vector address for the next (local) DL11 [0] (340)?

What is the CSR address for the next (remote) DL11 [0] (175650)?

What is the vector address for the next (remote) DL11 [0] (340)?

What is the CSR address for the next (local) DL11 [0] (176550)?

What is the vector address for the next (local) DL11 [0] (350)?

What is the CSR address for the next (remote) DL11 [0] (175660)?

What is the vector address for the next (remote) DL11 [0] (350)?

What is the CSR address for the next (local) DL11 [0] (176560)?

What is the vector address for the next (local) DL11 [0] (360)?

What is the CSR address for the next (remote) DL11 [0] (175670)?

What is the vector address for the next (remote) DL11 [0] (360)?

RT-11 supports one DZ11-A or -B eight line multiplexor
with an additional DZ11-C or -D eight line multiplexor
for a maximum of 16 lines,or one DZV11 four line
multiplexor.  Modem support is available on
remote lines, but is limited to Bell 103 type modems or
equivalent with the "common clear to send and carrier"
option installed to operate the modem in an auto-answer mode.

Do you want DZ11 or DZV11 multiplexor support [Y/N] (N)?
```

Respond YES if the target system has a DZ11 or DZV11 multiplexor. The default is NO.

```
Do you want DZ11 multiplexor support [Y/N] (Y)?
```

Respond YES if the target system has a DZ11 multiplexor, NO if it does not. The default is YES. If you respond NO, one DZV11 is assumed.

```
Do you want support for two DZ11's [Y/N] (N)?

One DZV11 four line multiplexor assumed.
```

Respond YES if the target system has more than eight DZ11 lines. The default is NO. Note that the system will have at least 10 terminals in this case (console, plus eight on first DZ11 and n (≥1) on the second DZ11).

```
The lines must be initialized to a specific baud
rate. The choices are 300, 150, and 110 baud.

Do you want the lines initialized to 300 baud [Y/N] (Y)?
```

All DZ11 or DZV11 lines must be initialized to the same baud rate. Respond YES if the rate is to be 300 baud. The default is YES.

```
Do you want the lines initialized to 150 baud [Y/N] (Y)?
```

Respond YES if the DZ11 or DZV11 lines are to operate at 150 baud. The default is YES. If you respond NO, 110 baud is assumed.

```
110 baud is assumed.

The vectors and CSR addresses are assigned to the floating
device region and will vary with each installation.

What is the CSR address for the DZV11 [0] (160010)?
```

Respond by entering the keyboard control register address for the first DZV11. The default is 160010. The range for valid responses is from 160000 through 177570.

```
What is the vector address for the DZV11 [0] (300)?
```

Respond by entering the keyboard vector address for the first DZV11. The default is 300. The range for valid responses is from 60 through 770.

```
What is the CSR address for the first DZ11 [0] (160010)?
```

Respond by entering the keyboard control register address for the first DZ11. The default is 160010. The range for valid responses is from 160000 through 177570.

```
What is the vector address for the first DZ11 [0] (300)?
```

Respond by entering the keyboard vector address for the first DZ11. The default is 300. The range for valid responses is from 60 through 770.

```
What is the CSR address for the second DZ11 [0] (160020)?
```

Respond by entering the keyboard control register address for the second DZ11. The default is 160020. The range for valid responses is from 160000 through 177570.

```
     What is the vector address of the second DZ11 [O] (310)?
```

Respond by entering the keyboard vector address for the second DZ11. The default is 310. The range for valid responses is from 60 through 770.

```
     The following dialogue will ask 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. Physical unit numbers of the DZ11 or DZV11 lines
     are assigned first to local lines and then to remote
     lines. This assignment is permanent and its order cannot
     be changed.

     How many DZ11 lines are local terminals [D] (0)?
```

Respond by entering the number of DZ11 lines that are local terminals. The default is 0. The range for valid responses is from 0 through 16.

```
     How many remote DZ11 lines [D] (0)?
```

Enter the number of DZ11 lines that are connected to remote terminals. This response should equal the total number of DZ11 terminals less the number of local lines. The default is 0.

```
     How many DZV11 lines are local terminals [D] (0)?
```

Respond by entering the number of DZV11 lines that are local terminals. The default is 0. The range for valid responses is from 0 through 4.

```
     How many remote DZV11 lines [D] (0)?
```

Enter the number of DZV11 lines that are connected to remote terminals. This response should equal the total number of DZV11 terminals less the number of local lines. The default is 0.

```
     The terminal service requires a set of input and output
     ring buffers for each terminal supported. The output ring
     buffer default size is 40 characters. The input ring size
     depends on terminal width and amount of type-ahead
     expected. The default size is 82 characters.

     Size of the output buffers in characters [D] (40)?
```

Enter the number of characters that are to be buffered in the output ring buffers. Each character adds n bytes to the resident monitor size, where n is the total number of terminal lines. The default is 40 characters. The range for valid responses is from 10 to 132.

```
     Size of the input buffers in characters [D] (82)?
```

Enter the number of characters that are to be buffered in the input ring buffers. Each character adds n bytes to the resident monitor, where n is the number of terminals on the system. 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 74 characters, you can adjust the size of the input buffers appropriately. The minimum acceptable input buffer size is 74 characters. Specifying less causes an error during monitor assembly. The monitor requires

82-character buffers for most efficient command operation. Buffers larger than 82 characters are rarely useful unless you anticipate a great deal of type-ahead. The default is 82 characters. The range for valid responses is from 74 through 132.

```
Terminal status may be obtained by doing an .MTGET request
but some applications require asynchronous notification of
changes in terminal status. RT-11 supports asynchronous
terminal status as an option to the .MTATCH
request and provides information on buffer and modem
status as well as double CTRL/C.

Do you want asynchronous terminal status [Y/N] (Y)?
```

Respond YES if your application requires asynchronous notification of changes in terminal status. The default is YES. If in doubt, respond YES. This option adds approximately 50 words to the resident monitor.

```
Time-out support will cause the monitor to reset at regular
intervals any terminal that may have gone off-line. This
helps to minimize the impact of static (and similar)
problems, and is recommended if maximum availability
is important to the application.

Do you want multi-terminal time-out support [Y/N] (Y)?
```

Respond YES if maximum terminal availability is important to your application. Respond NO only in the most space-constrained of situations. The default is YES. This option adds approximately 60 words to the resident monitor.

```
The SYSGEN assembling and linking process requires a
source input 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
Section 3.6.1 of the RT-11 System Generation Manual.

What is the PHYSICAL name and unit of the
    source input device [DDU] (RK1)?
```

The SYSGEN command files use the logical device SRC: for source input and begin by assigning SRC to the appropriate physical device. Enter the physical device and unit number for the device on which the system sources will reside during system generation. The default is RK1.

```
What is the PHYSICAL name and unit of the
    binary output device [DDU] (RK0)?
```

The SYSGEN command files use the logical device BIN: for binary and system output and begin by assigning BIN to the appropriate physical device. Enter the physical device and unit name for the device that will receive the .OBJ and system files output during SYSGEN. This device can be the same as the source input device, it can be an independent output device, or it can be a system device if there is sufficient space. The default is RK0.

```
What is the PHYSICAL name and unit of the
    map output device [DDU] (TT)?
```

The SYSGEN command files use the logical device MAP: for listing the link maps that result when you assemble and link the system components. If you submit an SPR to DIGITAL for a system that you created through the SYSGEN process, you must also send the link maps for that system. The default device is the console terminal, TT. You can specify LP for the line printer, if your system has one. If you want to build a file for the map listings, specify the physical name of a block-replaceable device. Usually, the binary output device is suitable for the map files as well.

```
To build an entire system, mount the source and
binary disks, copy the files SYCND.MAC and SYSTBL.MAC
to the source disk, and type "@SYSBLD". To build just the monitors,
type "@MONBLD". To build just the device handlers, type
"@DEVBLD". For more information, read Section 3.6 of the
RT-11 System Generation Manual.

END OF SYSGEN PROGRAM--Don't forget to copy DK:SYCND.MAC and
        DK:SYSTBL.MAC to the source disk before executing the
        command file(s). Remember that SWAP.SYS must
        be on all bootable volumes.
```

At this point the two conditional files (SYCND.MAC and SYSTBL.MAC) and the three command files (MONBLD.COM, DEVBLD.COM and SYSBLD.COM) reside on device DK: and are ready to use to build the system. Don't forget to copy SWAP.SYS onto your final system volume along with the monitor file.

3.5 SYSGEN ERROR MESSAGES

SYSGEN is a FORTRAN program and as such reports some error conditions in FORTRAN format.

The error messages likely to be output by SYSGEN are listed below. For other errors, see the RT-11 System Message Manual.

Message	Cause
?ERR 28 Open failed for file in routine "MAIN" line 17	SYSGEN.CND not present on DK:.
?ERR 28 Open failed for file in routine "MAIN" line 19	SYSTBL.CND not present on DK:.
?ERR 62 FORTRAN start fail	Insufficient memory to run SYSGEN.
?SYSGEN-F-Answer value too large	The response exceeded the highest acceptable response. Refer to the specific question in Section 3.4.2 for the range of valid responses.
?SYSGEN-F-Answer value too small	The response was less than the lowest acceptable response. Refer to the specific question in Section 3.4.2 for the range of valid responses.
?SYSGEN-F-Inappropriate answer	The response was not of the type (alphabetic or numeric) required.
?SYSGEN-F-Illegal system device for base-line monitor	Caused by selecting an illegal device for the base-line SJ monitor. The valid devices are: RK:, DT:, and DX:.

?SYSGEN-F-Input error — Error reading the file SYSGEN.CND.

?SYSGEN-F-Insert error — Error reading the file SYSTBL.CND.

?SYSGEN-F-MAGtape specification error — Failure to select either of the magtape controller options after specifying that magtape support was desired.

?SYSGEN-F-No monitor requested — Failure to select one of the four monitors.

?SYSGEN-F-Output error — Insufficient space on DK: to store the command and conditional files, or a hard I/O error in those files.

3.6 BUILDING THE SYSTEM

This section describes the use of the conditional and command files to assemble and link the new system.

The command files output by SYSGEN are customized to assemble and link the monitor and handlers. These command files consist of assembly, link and, if you did not choose to retain the .OBJ files, file deletion commands. You can execute the command files as is after the system has been suitably prepared, or you can alter them as long as the final commands perform equivalent functions to those generated by SYSGEN. Examples of alterations you might want to make include:

1. Adding listing output files to the assembly strings. The default output device is TT:. You can change the output device to create a file instead of a listing. You can suppress the link maps by sending the output to the null device, NL:. Note that you must include a link map and SYCND listing with any SPR you submit to DIGITAL for a monitor created by SYSGEN.

2. Changing the devices on which individual source or binary files are to reside to optimize mass storage usage.

You need not execute the command files as command files. You can, of course, type the commands individually. This can be necessary if there is insufficient mass storage to execute the entire file or to repeat a particular step that generated an error without repeating the whole command file.

The conditional files (SYCND.MAC and SYSTBL.MAC), once developed with SYSGEN, can of course be used over and over without rerunning the SYSGEN program. To regenerate a system including source patches, simply edit the source file and rebuild the system using the existing SYCND.MAC and SYSTBL.MAC files. You need only rerun SYSGEN if you want to change the configuration; it is not necessary to rerun SYSGEN to rebuild an existing configuration.

If you make an error in the SYSGEN dialogue (for example, if you type an alphanumeric response where a numeric response is necessary), the program prints an error message and repeats the question so you can enter the correct response. To make other changes to your responses after the dialogue is complete, you should rerun SYSGEN. It is possible to avoid rerunning the SYSGEN program by editing the conditional file itself to correct any problem. This requires great care, however, and only the knowledgeable user should manipulate these files. If in doubt, rerun SYSGEN.

3.6.1 Customizing the SYCND File

The SYCND.MAC file created by the SYSGEN dialogue contains conditionals that reflect your target system. Before you execute the command files, you can edit SYCND to make minor changes.

3.6.1.1 **Multi-Terminal Support for United Kingdom** - RT-11 does not disconnect remote lines immediately when the carrier is lost. It waits for a period of time to avoid dropping a line due to noise. United Kingdom applications require RT-11 to disconnect a line immediately when the line hangs up. The DL11-E interface with modem support has a conditional that you can add to SYCND to include support for immediate disconnect. Insert the following line in SYCND.MAC:

```
U.K. = 1
```

3.6.1.2 **Non-Standard CSR and Vector Support** - You can customize support for devices that are located at non-standard addresses by changing the conditionals in SYCND that define those addresses. Insert a line in SYCND.MAC for each non-standard CSR and vector address. The format of the conditional is as follows:

```
dd$xxx = nnnnnn
```

where

dd   is the two-letter device mnemonic

xxx   must be replaced by one of the following:

VEC for first controller's vector
CSR for first controller's CSR address
VC2 for second controller's vector
CS2 for second controller's CSR address

nnnnnn   is the six-digit octal address

3.6.2 Requirements for Building a System

Executing the SYSGEN command files requires sufficient free space on the binary output device to accommodate the .OBJ and .SYG output files, the monitor link map files, and the proper source components (as distributed with the system) on the source input device.

3.6.2.1 **Necessary Components** - The SYSGEN command files require a running RT-11 system on the system device that includes a monitor, SWAP.SYS, necessary handlers for the source and binary devices, MACRO, LINK, PIP and SYSMAC.SML. Most working disks have these components. DIGITAL recommends that you use the SJ monitor in order to leave as much memory as possible available for MACRO's symbol tables. If disk space is a serious constraint, you can build a special SYSGEN system disk limited to these components (and perhaps DUP and DIR). The system disk should have at least 128 free blocks to accommodate the work file needed by MACRO during system assembly.

The source disk must contain the following source files (all of which are in the distribution kits). The source disk can be the same as the system disk and/or binary disk if sufficient space is available.

SYCND.MAC	(output by SYSGEN)
SYSTBL.MAC	(output by SYSGEN)
SJ.MAC	(if SJ or BL monitor being built)
FB.MAC	(if FB monitor being built)
XM.MAC	(if XM monitor being built)
SYSDEV.MAC	
KMON.MAC	
USR.MAC	
RMONSJ.MAC	(if SJ or BL monitor being built)
RMONFB.MAC	(if FB or XM monitor being built)
KMOVLY.MAC	
BSTRAP.MAC	
MTTEMT.MAC	(if multi-terminal support needed)
MTTINT.MAC	(if multi-terminal support needed)
TT.MAC	(if SJ or BL monitor being built)
EL.MAC	(if error logging included)
BA.MAC	(if BATCH support included)
LP.MAC	(if line printer support included)
RK.MAC	(if RK05 support included)
DL.MAC	(if RL01 support included)
DY.MAC	(if RX02 support included)
DT.MAC	(if DECtape support included)
DS.MAC	(if RJS03/4 support included)
PC.MAC	(if high-speed paper tape support included)
RF.MAC	(if RF11 support included)
DM.MAC	(if RK06/07 support included)
DX.MAC	(if RX01 support included)
CR.MAC	(if card reader support included)
DP.MAC	(if DP11 support included)
CT.MAC	(if TA11 support included)
NL.MAC	(if null handler support included)
FSM.MAC	(if file-structured magtape support included)
TM.MAC	(if TM11 magtape support included)
TJ.MAC	(if TJU16 magtape support included)

3.6.2.2 **Size Requirements on Binary Disk** - The binary disk must have sufficient free space to store the object modules and system files output by the system build process. If you do not retain the object modules, the space required is reduced to that necessary to store the largest set of concurrent .OBJ files needed and the system files. DIGITAL recommends that you assign a separate device other than the binary device for the temporary work files. Use the ASSIGN command to do this:

```
.ASSIGN device WF RET
```

An approximate guide for the free storage necessary is:

```
300 blocks for each monitor produced
 10 blocks for each SJ, FB handler produced
 10 blocks for each XM handler produced
200 blocks for each set of monitor object modules retained
  8 blocks for each set of handler object modules retained
```

Hence, if you are building only one monitor with a typical complement of handlers, approximately 500 blocks of free storage is sufficient. If you are building all monitors with support for all devices and are retaining the object modules (the worst case), the storage needs can approach 2000 blocks.

Make the free storage on the binary device contiguous prior to starting a system build (to do this, SQUEEZE the binary device before starting).

The SYSGEN command files create the new system files with a .SYG file type (or .BLG for sysgened BL monitors). This allows you to use the system device as the binary device, and enables you to tell the new system files from the running (.SYS) files. When installing the new system, rename the .SYG or .BLG files to .SYS.

The recommended procedure is to use separate source, binary, and system devices if possible; this speeds up the build process. If only two devices are available, combine the source and binary disks if possible.

3.6.3 **Building the Monitors**

If there is sufficient free space on the binary output device, building the monitor consists of the following simple steps:

1. Mount the source disk; transfer SYCND.MAC and SYSTBL.MAC to the source disk (if not there already).

2. Mount the binary disk (if not the source disk) and SQUEEZE it to consolidate free space.

3. Type: @MONBLD RET

   The monitor logs the commands on the terminal as they are executed.

If you do not have sufficient free space on the binary output device, or if the source files do not fit all on one volume, enter the keyboard monitor commands that appear in MONBLD.COM by hand, shuffling files and volumes as necessary. (See Appendix A for an example SYSGEN run.)

3.6.4 **Errors When Building the Monitor and Handlers**

The most common errors that occur during system building are

- Assembly errors
- FILE NOT FOUND errors
- OUTPUT FILE FULL errors
- Hard I/O errors

Assembly errors are most often caused by incorrect or conflicting responses during the SYSGEN dialogue. If you get an assembly error:

1. Carefully re-examine your responses to SYSGEN. You can often spot the problem by reading the conditional files (SYCND.MAC and SYSTBL.MAC).

2. If all is correct, make certain that you have the proper versions of the source files on the source disk and, if you have edited them, that you have not introduced an error. Compare the files on the source disk to the originals on the distribution disk to check for differences.

FILE NOT FOUND errors are most commonly caused by failure to copy the SYCND.MAC or SYSTBL.MAC files to the source disk or by failure to include all necessary source modules on the disk.

OUTPUT FILE FULL errors are caused when there is not sufficient space on the binary output device to accommodate all output files. This condition can also cause the "?MACRO-F-Input-output error on workfile" message to appear. This usually means you did not assign the work file, or it needs more space. If you did not SQUEEZE the binary device before starting, try again after SQUEEZEing. If you elected to retain the .OBJ files, rerun SYSGEN and obtain command files that delete the .OBJ files. If you have other unnecessary files on the binary device, delete them. If all this fails to correct the problem, you must enter the system build commands by hand, shuffling volumes as you go.

Hard I/O errors are caused by bad volumes or write-locked drives. If the drive is write-locked, enable it. If the volume is bad, try another.

If you encounter an error which you cannot explain or correct, send a listing of the conditional files SYCND.MAC and SYSTBL.MAC along with the console output in a SPR to DIGITAL. Be sure to include the conditional file listings and all relevant console output.

When you boot a monitor that you created with SYSGEN, it prints an identifying message like the following:

```
     RT-11xx (S)V03B-xx
```

The (S) indicates that the monitor was created by SYSGEN.

3.6.5 Building the Handlers

The instructions for generating the handlers are identical to those for generating the monitor except that you substitute the handler command file DEVBLD.COM for the monitor file MONBLD.COM.

If you desire to build both the handlers and monitors in the same operation, use the command file SYSBLD.COM, which simply invokes the other two.

In certain situations it is desirable to build a monitor with only a few device slots, build many device handlers to go along with that monitor, then use the INSTALL and REMOVE commands to activate and deactivate device handlers as needed. Each device selected as part of the SYSGEN dialogue is allocated a slot, and commands for building it are output in DEVBLD. Devices not selected are not given a slot in the device tables, nor are commands for handler generation included in DEVBLD. The procedure for building handlers for devices not included in the monitor tables is as follows:

1. Execute SYSGEN, selecting monitor options desired and selecting support for those devices for which slots are to be allocated. The resulting conditional and command files are used to build the basic system.

2. Re-run SYSGEN, specifying IDENTICAL monitor options, but this time selecting only the additional devices for which handlers are to be generated. You can then use the resulting SYCND.MAC and DEVBLD file to build the additional handlers. Do not use the MONBLD and SYSTBL files output by this second run; discard them to minimize confusion.

3.6.6 Keeping System Components

Once you have built the system, keep the conditional, command, and output files from the generation process if possible. You must retain at least the conditional and command files. All SPR's on RT-11 Version 3B must be accompanied by a listing of these files as well as link maps if a monitor created by SYSGEN is being used.

If you retain the .OBJ files, you can reassemble a particular component and relink by hand (should that be necessary), saving the need to rebuild the entire system each time you make a minor patch.

The .SYG files output by the generation become the equivalent of the distribution masters for the customized system and should be preserved and stored accordingly.

CHAPTER 4

ASSEMBLY AND LINK INSTRUCTIONS FOR SYSTEM PROGRAMS

4.1 GENERAL INSTRUCTIONS

You should assemble the RT-11 monitors and device handlers in at least 16K words of memory; less memory makes the assembly times prohibitive. Use RT-11 MACRO as the assembler and RT-11 LINK as the linker in all cases. Ensure that all assemblies (except ODT) and links (except where otherwise noted) are error free.

Throughout this chapter, the following conventions are used:

1. Default file types are not explicitly specified. For all the source files, the file types are .MAC. The assembler output is .OBJ and linker output is .SAV.

2. The system macro library, SYSMAC.SML, must be on the system device during all assemblies given below.

3. In the example command strings, the sources are kept on logical device SRC:, binary output is to device BIN:, and listing and map files are output to LST:. In actual practice, you can use any appropriate device.

4. For the purposes of these commands, SYSLIB refers to a system library containing only the FORTRAN OTS routines.

The instructions are presented as BATCH streams that you can use to generate the various system components. You can substitute equivalent indirect files or direct keyboard monitor commands if BATCH is not appropriate.

All RT-11 system assembling and linking operations are normal operations. You can alter the command strings in the descriptions below to take full advantage of all RT-11 MACRO and LINK commands. Most of the commands shown are CSI level commands; you can, of course, substitute equivalent monitor commands.

Assembly and link instructions for the monitor and handlers are output in command files as part of the SYSGEN process. Examine MONBLD.COM and DEVBLD.COM after an appropriate SYSGEN run for monitor assembly and link instructions.

4.2 BUILDING THE SYSTEM MACRO LIBRARY

```
$JOB/RT11/TIME  !SYSMAC.BAT
        TTYIO
$MES MAKE SYSMAC.SML
.R LIBR
*BIN:SYSMAC.SML=SRC:SYSMAC.MAC/M
```

4.3 BUILDING BATCH

```
$JOB/RT11/TIME   !BATCH.BAT
        TTYIO
$MES    MAKE BATCH
.R MACRO
*BIN:BA,LST:BA=SRC:SYCND,BA/C
*BIN:BAX,LST:BAX=SRC:XM,SYCND,BA/C
*BIN:BATCH,LST:BATCH=SRC:BATCH/C
.R LINK
*BIN:BA.SYS,LST:BA=BIN:BA
*BIN:BAX.SYS,LST:BAX=BIN:BAX
*BIN:BATCH,LST:BATCH=BIN:BATCH
$EOJ
```

4.4 BUILDING PIP

```
$JOB/RT11/TIME   !PIP.BAT
        TTYIO
$MES    MAKE PIP
.R MACRO
*BIN:PIP,LST:PIP=SRC:PIP/C
.R LINK
*BIN:PIP,LST:PIP=BIN:PIP/B:700
$EOJ
```

4.5 BUILDING DUP

```
$JOB/RT11/TIME   !DUP.BAT
        TTYIO
$MES    MAKE DUP
.R MACRO
*BIN:DUPROT,LST:DUPROT/C=SRC:DUPROT
*BIN:DUPIMA,LST:DUPIMA/C=SRC:DUPIMA
*BIN:DUPEXT,LST:DUPEXT/C=SRC:DUPEXT
*BIN:DUPSCN,LST:DUPSCN/C=SRC:DUPSCN
*BIN:DUPSQU,LST:DUPSQU/C=SRC:DUPSQU
*BIN:DUPZRO,LST:DUPZRO/C=SRC:DUPZRO
*BIN:DUPCRE,LST:DUPCRE/C=SRC:DUPCRE
*BIN:DUPBOT,LST:DUPBOT/C=SRC:DUPBOT
*BIN:MDUPRO,LST:MDUPRO/C=SRC:MDUP,DUPROT
*BIN:MDUPZR,LST:MDUPZR/C=SRC:MDUP,DUPZRO
*BIN:MDUPBO,LST:MDUPBO/C=SRC:MDUP,DUPBOT
*BIN:MDUPSQ,LST:MDUPSQ/C=SRC:MDUP,DUPSQU
*BIN:MBOOT,LST:MBOOT=SRC:MBOOT/C
*BIN:MSBOOT,LST:MSBOOT=SRC:MSBOOT/C
.R LINK
*BIN:DUP,LST:DUP=BIN:DUPROT/W//
*BIN:DUPZRO/O:1
*BIN:DUPSCN,DUPEXT,DUPCRE,DUPBOT,DUPIMA/O:1
*BIN:DUPSQU/O:1//
*BIN:MDUP,LST:MDUP=BIN:MDUPRO,MDUPZR,MDUPSQ,MDUPBO/W
*BIN:MBOOT.BOT,LST:MBOOT=BIN:MBOOT
*BIN:MSBOOT.BOT,LST:MSBOOT=BIN:MSBOOT
$EOJ
```

4.6 BUILDING DIR

```
$JOB/RT11/TIME  !DIRECT.BAT
        TTYIO
$MES MAKE DIR
.R MACRO
*BIN:DIR1ST,LST:DIR1ST/C=SRC:DIR1ST
*BIN:DIRDAT,LST:DIRDAT/C=SRC:DIRPRE,DIRDAT
*BIN:DIRMAN,LST:DIRMAN/C=SRC:DIRPRE,DIRMAN
*BIN:DIRT11,LST:DIRT11/C=SRC:DIRPRE,DIRT11
*BIN:DIRSWT,LST:DIRSWT/C=SRC:DIRPRE,DIRSWT
*BIN:DIRMAT,LST:DIRMAT/C=SRC:DIRPRE,DIRMAT
*BIN:DIRDK,LST:DIRDK/C=SRC:DIRPRE,DIRDK
*BIN:DIRMT,LST:DIRMT/C=SRC:DIRPRE,DIRMT
*BIN:DIRCT,LST:DIRCT/C=SRC:DIRPRE,DIRCT
*BIN:DIRSUP,LST:DIRSUP/C=SRC:DIRPRE,DIRSUP
*BIN:DIRSRT,LST:DIRSRT/C=SRC:DIRPRE,DIRSRT
*BIN:DIROUT,LST:DIROUT/C=SRC:DIRPRE,DIROUT
*BIN:DIRERR,LST:DIRERR/C=SRC:DIRPRE,DIRERR
*BIN:DIRFMT,LST:DIRFMT/C=SRC:DIRMAC/M,DIRPRE,DIRFMT
.R LINK
*BIN:DIR,LST:DIR/W=BIN:DIR1ST,DIRDAT,DIRMAN,DIRT11/U:100//
*BIN:DIRSWT,DIRMAT,DIRDK,DIRMT,DIRCT,DIRSUP
*BIN:DIRSRT,DIROUT,DIRERR,DIRFMT
*//
*PATCH
$EOJ
```

4.7 BUILDING EDIT

```
$JOB/RT11/TIME  !EDIT.BAT
        TTYIO
$MES    MAKE EDIT
.R MACRO
*BIN:VTCED1,LST:VTCED1=SRC:EDITDF,VTCAL1/C
*BIN:VTCED4,LST:VTCED4=SRC:EDITDF/P:1,VTCAL4/C
*BIN:VTBEDT,LST:VTBEDT=SRC:EDITDF/P:1,VTBASE/C
*BIN:EDIT,LST:EDIT=SRC:VTMAC,EDIT/C
.R LINK
*BIN:EDIT,LST:EDIT=BIN:VTCED1,VTCED4,VTBEDT,EDIT/W
$EOJ
```

4.8 BUILDING HELP

```
$JOB/RT11/TIME  !HELP.BAT
        TTYIO
$MES    MAKE HELP
.R MACRO
*BIN:HELP,LST:HELP=SRC:HELP,TIO/C
.R LINK
*BIN:HELP,LST:HELP=BIN:HELP,SRC:TECO/B:2000/W
$EOJ
```

4.9 BUILDING ERRUTL

```
$JOB/RT11/TIME   !ELOG.BAT
        TTYIO
$MES    MAKE ERROR LOG UTILITY
.R MACRO
*BIN:ERRUTL,LST:ERRUTL=SRC:ERRUTL/C
.R LINK
*BIN:ERRUTL,LST:ERRUTL=BIN:ERRUTL
$EOJ
```

4.10 BUILDING MACRO

```
$JOB/RT11               !MACRO.BAT
        TTYIO
$MES MAKE MACRO
.TIM
.R MACRO
*BIN:MMAIN,LST:MMAIN=SRC:MPAR3,MMAIN,MEND/C
*BIN:MNEWCL,LST:MNEWCL=SRC:MPAR3/P:1,MNEWCL,MEND/C
*BIN:MCRFIO,LST:MCRFIO=SRC:MPAR3/P:1,MCRFIO,MEND/C
*BIN:MNVMRS,LST:MNVMRS=SRC:MPAR3/P:1,MACFLM,MNVMRS,MEND/C
*BIN:MLABEL,LST:MLABEL=SRC:MPAR3/P:1,MLABEL,MEND/C
*BIN:MNEWST,LST:MNEWST=SRC:MPAR3/P:1,MNEWST,MEND/C
*BIN:MASGMT,LST:MASGMT=SRC:MPAR3/P:1,MASGMT,MEND/C
*BIN:MENDPS,LST:MENDPS=SRC:MPAR3/P:1,MENDPS,MEND/C
*BIN:MENDLN,LST:MENDLN=SRC:MPAR3/P:1,MENDLN,MEND/C
*BIN:MSETDN,LST:MSETDN=SRC:MPAR3/P:1,MSETDN,MEND/C
*BIN:MINOFL,LST:MINOFL=SRC:MPAR3/P:1,MINOFL,MEND/C
*BIN:MSETUP,LST:MSETUP=SRC:MPAR3/P:1,MSETUP/C
*BIN:MINIVM,LST:MINIVM=SRC:MPAR3/P:1,MACFLM/P:1,MINIVM,MEND/C
*BIN:MINIFL,LST:MINIFL=SRC:MPAR3/P:1,MINIFL,MEND/C
*BIN:MASSEM,LST:MASSEM=SRC:MPAR3/P:1,MASSEM,MEND/C
*BIN:MNDRCT,LST:MNDRCT=SRC:MPAR3/P:1,MNDRCT,MEND/C
*BIN:MPROPC,LST:MPROPC=SRC:MPAR3/P:1,MPROPC,MEND/C
*BIN:MACRD,LST:MACRD=SRC:MPAR3/P:1,MACRD,MEND/C
*BIN:MNMLIS,LST:MNMLIS=SRC:MPAR3/P:1,MNMLIS,MEND/C
*BIN:MNCNDL,LST:MNCNDL=SRC:MPAR3/P:1,MNCNDL,MEND/C
*BIN:MFLOAT,LST:MFLOAT=SRC:MPAR3/P:1,MFLOAT,MEND/C
*BIN:MSCDR,LST:MSCDR=SRC:MPAR3/P:1,MSCDR,MEND/C
*BIN:MLISTC,LST:MLISTC=SRC:MPAR3/P:1,MLISTC,MEND/C
*BIN:MENBDS,LST:MENBDS=SRC:MPAR3/P:1,MENBDS,MEND/C
*BIN:MNSECR,LST:MNSECR=SRC:MPAR3/P:1,MNSECR,MEND/C
*BIN:MSETMX,LST:MSETMX=SRC:MPAR3/P:1,MSETMX,MEND/C
*BIN:MDATDR,LST:MDATDR=SRC:MPAR3/P:1,MDATDR,MEND/C
*BIN:MOPTER,LST:MOPTER=SRC:MPAR3/P:1,MOPTER,MEND/C
*BIN:MWORDB,LST:MWORDB=SRC:MPAR3/P:1,MWORDB,MEND/C
*BIN:MNMACS,LST:MNMACS=SRC:MPAR3/P:1,MNMACS,MEND/C
*BIN:MREPT,LST:MREPT=SRC:MPAR3/P:1,MREPT,MEND/C
*BIN:MOVPST,LST:MOVPST=SRC:MPAR3/P:1,MOVPST,MEND/C
*BIN:MNEXPS,LST:MNEXPS=SRC:MPAR3/P:1,MNEXPS,MEND/C
*BIN:MNEWCD,LST:MNEWCD=SRC:MPAR3/P:1,MNEWCD,MEND/C
*BIN:MACOP,LST:MACOP=SRC:MPAR3/P:1,MACOP,MEND/C
*BIN:MSCNLB,LST:MSCNLB=SRC:MPAR3/P:1,MSCNLB,MEND/C
*BIN:MACRC,LST:MACRC=SRC:MPAR3/P:1,MACRC,MEND/C
*BIN:MGETLN,LST:MGETLN=SRC:MPAR3/P:1,MGETLN,MEND/C
*BIN:MSETIM,LST:MSETIM=SRC:MPAR3/P:1,MSETIM,MEND/C
*BIN:MSETDR,LST:MSETDR=SRC:MPAR3/P:1,MSETDR,MEND/C
*BIN:MGMARG,LST:MGMARG=SRC:MPAR3/P:1,MGMARG,MEND/C
*BIN:MROLPT,LST:MROLPT=SRC:MPAR3/P:1,MROLPT,MEND/C
*BIN:MNLSTG,LST:MNLSTG=SRC:MPAR3/P:1,MNLSTG,MEND/C
*BIN:MR5UNP,LST:MR5UNP=SRC:MPAR3/P:1,MR5UNP,MEND/C
*BIN:MSYMBL,LST:MSYMBL=SRC:MPAR3/P:1,MSYMBL,MEND/C
*BIN:MVMOVR,LST:MVMOVR=SRC:MPAR3/P:1,MACFLM/P:1,MVMOVR,MEND/C
```

```
*BIN:MPUSH,LST:MPUSH=SRC:MPAR3/P:1,MPUSH,MEND/C
*BIN:MSPACE,LST:MSPACE=SRC:MPAR3/P:1,MSPACE,MEND/C
*BIN:MRDWR,LST:MRDWR=SRC:MPAR3/P:1,MRDWR,MEND/C
.R LINK
*BIN:MACRO,LST:MACRO=BIN:MMAIN/T/W//
*BIN:MSPACE,MSETMX,MPUSH,MNMACS,MCRFIO
*BIN:MR5UNP,MNVMRS,MACOP,MACRC
*BIN:MSCNLB,MENDLN,MPROPC
*BIN:MNEWCL,MNCNDL,MSCDR
*BIN:MREPT,MWORDB,MNSECR
*BIN:MDATDR,MOVPST,MNEWCD
*BIN:MNEXPS,MGETLN,MSETIM,MSETDR,MGMARG
*BIN:MROLPT,MSYMBL,MVMOVR
*BIN:MINIFL,MSETUP,MINOFL/O:1
*BIN:MASSEM,MSETDN/O:1
*BIN:MINIVM,MENDPS/O:1
*BIN:MACRD,MLABEL,MASGMT,MNEWST,MNDRCT/O:1
*BIN:MFLOAT/O:2
*BIN:MRDWR,MOPTER/O:2
*BIN:MNMLIS/O:3
*BIN:MNLSTG,MENBDS,MLISTC/O:3//
*START
.TIM
$MES MAKE MAC8K
.R MACRO
*BIN:MMAIN.M8K,LST:MMAIN.LS8=SRC:MPAR2,MMAIN,MEND/C
*BIN:MNEWCL.M8K,LST:MNEWCL.LS8=SRC:MPAR2/P:1,MNEWCL,MEND/C
*BIN:MCRFIO.M8K,LST:MCRFIO.LS8=SRC:MPAR2/P:1,MCRFIO,MEND/C
*BIN:MNVMRS.M8K,LST:MNVMRS.LS8=SRC:MPAR2/P:1,MACFLM/P:1,MNVMRS,MEND/C
*BIN:MLABEL.M8K,LST:MLABEL.LS8=SRC:MPAR2/P:1,MLABEL,MEND/C
*BIN:MNEWST.M8K,LST:MNEWST.LS8=SRC:MPAR2/P:1,MNEWST,MEND/C
*BIN:MASGMT.M8K,LST:MASGMT.LS8=SRC:MPAR2/P:1,MASGMT,MEND/C
*BIN:MENDPS.M8K,LST:MENDPS.LS8=SRC:MPAR2/P:1,MENDPS,MEND/C
*BIN:MENDLN.M8K,LST:MENDLN.LS8=SRC:MPAR2/P:1,MENDLN,MEND/C
*BIN:MSETDN.M8K,LST:MSETDN.LS8=SRC:MPAR2/P:1,MSETDN,MEND/C
*BIN:MINOFL.M8K,LST:MINOFL.LS8=SRC:MPAR2/P:1,MINOFL,MEND/C
*BIN:MSETUP.M8K,LST:MSETUP.LS8=SRC:MPAR2/P:1,MSETUP/C
*BIN:MINIVM.M8K,LST:MINIVM.LS8=SRC:MPAR2/P:1,MACFLM/P:1,MINIVM,MEND/C
*BIN:MINIFL.M8K,LST:MINIFL.LS8=SRC:MPAR2/P:1,MINIFL,MEND/C
*BIN:MASSEM.M8K,LST:MASSEM.LS8=SRC:MPAR2/P:1,MASSEM,MEND/C
*BIN:MNDRCT.M8K,LST:MNDRCT.LS8=SRC:MPAR2/P:1,MNDRCT,MEND/C
*BIN:MPROPC.M8K,LST:MPROPC.LS8=SRC:MPAR2/P:1,MPROPC,MEND/C
*BIN:MACRD.M8K,LST:MACRD.LS8=SRC:MPAR2/P:1,MACRD,MEND/C
*BIN:MNMLIS.M8K,LST:MNMLIS.LS8=SRC:MPAR2/P:1,MNMLIS,MEND/C
*BIN:MNCNDL.M8K,LST:MNCNDL.LS8=SRC:MPAR2/P:1,MNCNDL,MEND/C
*BIN:MFLOAT.M8K,LST:MFLOAT.LS8=SRC:MPAR2/P:1,MFLOAT,MEND/C
*BIN:MSCDR.M8K,LST:MSCDR.LS8=SRC:MPAR2/P:1,MSCDR,MEND/C
*BIN:MLISTC.M8K,LST:MLISTC.LS8=SRC:MPAR2/P:1,MLISTC,MEND/C
*BIN:MENBDS.M8K,LST:MENBDS.LS8=SRC:MPAR2/P:1,MENBDS,MEND/C
*BIN:MNSECR.M8K,LST:MNSECR.LS8=SRC:MPAR2/P:1,MNSECR,MEND/C
*BIN:MSETMX.M8K,LST:MSETMX.LS8=SRC:MPAR2/P:1,MSETMX,MEND/C
*BIN:MDATDR.M8K,LST:MDATDR.LS8=SRC:MPAR2/P:1,MDATDR,MEND/C
*BIN:MOPTER.M8K,LST:MOPTER.LS8=SRC:MPAR2/P:1,MOPTER,MEND/C
*BIN:MWORDB.M8K,LST:MWORDB.LS8=SRC:MPAR2/P:1,MWORDB,MEND/C
*BIN:MNMACS.M8K,LST:MNMACS.LS8=SRC:MPAR2/P:1,MNMACS,MEND/C
*BIN:MREPT.M8K,LST:MREPT.LS8=SRC:MPAR2/P:1,MREPT,MEND/C
*BIN:MOVPST.M8K,LST:MOVPST.LS8=SRC:MPAR2/P:1,MOVPST,MEND/C
*BIN:MNEXPS.M8K,LST:MNEXPS.LS8=SRC:MPAR2/P:1,MNEXPS,MEND/C
*BIN:MNEWCD.M8K,LST:MNEWCD.LS8=SRC:MPAR2/P:1,MNEWCD,MEND/C
*BIN:MACOP.M8K,LST:MACOP.LS8=SRC:MPAR2/P:1,MACOP,MEND/C
*BIN:MSCNLB.M8K,LST:MSCNLB.LS8=SRC:MPAR2/P:1,MSCNLB,MEND/C
*BIN:MACRC.M8K,LST:MACRC.LS8=SRC:MPAR2/P:1,MACRC,MEND/C
*BIN:MGETLN.M8K,LST:MGETLN.LS8=SRC:MPAR2/P:1,MGETLN,MEND/C
*BIN:MSETIM.M8K,LST:MSETIM.LS8=SRC:MPAR2/P:1,MSETIM,MEND/C
*BIN:MSETDR.M8K,LST:MSETDR.LS8=SRC:MPAR2/P:1,MSETDR,MEND/C
*BIN:MGMARG.M8K,LST:MGMARG.LS8=SRC:MPAR2/P:1,MGMARG,MEND/C
*BIN:MROLPT.M8K,LST:MROLPT.LS8=SRC:MPAR2/P:1,MROLPT,MEND/C
*BIN:MNLSTG.M8K,LST:MNLSTG.LS8=SRC:MPAR2/P:1,MNLSTG,MEND/C
```

```
*BIN:MR5UNP.M8K,LST:MR5UNP.LS8=SRC:MPAR2/P:1,MR5UNP,MEND/C
*BIN:MSYMBL.M8K,LST:MSYMBL.LS8=SRC:MPAR2/P:1,MSYMBL,MEND/C
*BIN:MVMOVR.M8K,LST:MVMOVR.LS8=SRC:MPAR2/P:1,MACFLM/P:1,MVMOVR,MEND/C
*BIN:MPUSH.M8K,LST:MPUSH.LS8=SRC:MPAR2/P:1,MPUSH,MEND/C
*BIN:MSPACE.M8K,LST:MSPACE.LS8=SRC:MPAR2/P:1,MSPACE,MEND/C
*BIN:MRDWR.M8K,LST:MRDWR.LS8=SRC:MPAR2/P:1,MRDWR,MEND/C
.R LINK
*BIN:MAC8K,LST:MAC8K=BIN:MMAIN.M8K,MNEWCL.M8K/T/W/B:730//
*BIN:MCRFIO.M8K,MNVMRS.M8K
*BIN:MLABEL.M8K,MNEWST.M8K,MASGMT.M8K/O:1
*BIN:MENDPS.M8K/O:1
*BIN:MENDLN.M8K/O:1
*BIN:MSETDN.M8K/O:1
*BIN:MINOFL.M8K/O:1
*BIN:MSETUP.M8K,MINIVM.M8K,MINIFL.M8K,MASSEM.M8K,MNDRCT.M8K/O:1
*BIN:MPROPC.M8K/O:2
*BIN:MACRD.M8K,MNMLIS.M8K/O:2
*BIN:MNCNDL.M8K/O:2
*BIN:MFLOAT.M8K/O:2
*BIN:MSCDR.M8K/O:2
*BIN:MLISTC.M8K,MENBDS.M8K,MNSECR.M8K,MSETMX.M8K/O:2
*BIN:MDATDR.M8K,MOPTER.M8K,MWORDB.M8K/O:2
*BIN:MNMACS.M8K,MREPT.M8K/O:3
*BIN:MOVPST.M8K/O:3
*BIN:MNEXPS.M8K/O:4
*BIN:MNEWCD.M8K/O:4
*BIN:MACOP.M8K,MSCNLB.M8K,MACRC.M8K/O:4
*BIN:MGETLN.M8K/O:5
*BIN:MSETIM.M8K,MSETDR.M8K,MGMARG.M8K/O:5
*BIN:MROLPT.M8K/O:5
*BIN:MNLSTG.M8K,MR5UNP.M8K/O:5
*BIN:MSYMBL.M8K/O:6
*BIN:MVMOVR.M8K,MPUSH.M8K,MSPACE.M8K/O:6
*BIN:MRDWR.M8K/O:6//
*START
$EOJ
.R MACRO
*BIN:MMAIN.OPT=SRC:MPAR1,MMAIN,MEND
*BIN:MINIVM.OPT=SRC:MPAR1/P:1,MACFLM/P:1,MINIVM,MEND
*BIN:MROLPT.OPT=SRC:MPAR1/P:1,MROLPT,MEND
.R LINK
*BIN:MACFST,LST:MACFST=BIN:MMAIN.OPT/T/W//
*BIN:MSPACE,MSETMX,MPUSH,MNMACS,MCRFIO
*BIN:MR5UNP,MNVMRS,MACOP,MACRC
*BIN:MSCNLB,MENDLN,MPROPC
*BIN:MNEWCL,MNCNDL,MSCDR
*BIN:MREPT,MWORDB,MNSECR,MDATDR
*BIN:MOVPST,MNEWCD,MNEXPS,MGETLN
*BIN:MSETIM,MSETDR,MGMARG
*BIN:MROLPT.OPT,MSYMBL,MVMOVR
*BIN:MINIFL,MSETUP,MINOFL/O:1
*BIN:MASSEM,MSETDN/O:1
*BIN:MINIVM.OPT,MENDPS/O:1
*BIN:MACRD,MLABEL,MASGMT,MNEWST,MNDRCT/O:1
*BIN:MFLOAT/O:2
*BIN:MRDWR,MOPTER/O:2
*BIN:MNMLIS/O:3
*BIN:MNLSTG,MENBDS,MLISTC/O:3//
*START
$EOJ
```

4.11 BUILDING CREF

```
$JOB/RT11/TIME   !CREF.BAT
        TTYIO
$MES    MAKE CREF
.R MACRO
*BIN:CREF,LST:CREF=SRC:CREF/C
.R LINK
*BIN:CREF,LST:CREF=BIN:CREF
$EOJ
```

4.12 BUILDING LINK

```
$JOB/RT11/TIME   !LINK.BAT
        TTYIO
$MES    MAKE LINK
.R MACRO
*BIN:LINK0,LST:LINK0=SRC:LINK0/C
*BIN:LNKOV1,LST:LNKOV1=SRC:LNKOV1/C
*BIN:LNKGSD,LST:LNKGSD=SRC:LNKGSD/C
*BIN:LNKHDR,LST:LNKHDR=SRC:LNKHDR/C
*BIN:LNKMAP,LST:LNKMAP=SRC:LNKMAP/C
*BIN:LNKSAV,LST:LNKSAV=SRC:LNKSAV/C
*BIN:LNKEM,LST:LNKEM=SRC:LNKEM/C
.R LINK
*BIN:LINK,LST:LINK=BIN:LINK0/B:700/W//
*BIN:LNKOV1/O:1
*BIN:LNKGSD/O:1
*BIN:LNKHDR/O:1
*BIN:LNKMAP/O:1
*BIN:LNKSAV/O:1
*BIN:LNKEM/O:1//
$EOJ
```

4.13 BUILDING LIBR

```
$JOB/RT11/TIME   !LIBR.BAT
        TTYIO
$MES     MAKE LIBR
.R MACRO
*BIN:LIBR0,LST:LIBR0=SRC:LIBR0/C
*BIN:LIBR1,LST:LIBR1=SRC:LIBR1/C
*BIN:LIBR2,LST:LIBR2=SRC:LIBR2/C
*BIN:LIBR3,LST:LIBR3=SRC:LIBR3/C
*BIN:LIBR4,LST:LIBR4=SRC:LIBR4/C
*BIN:LIBR5,LST:LIBR5=SRC:LIBR5/C
*BIN:LBREM,LST:LBREM=SRC:LBREM/C
.R LINK
*BIN:LIBR,LST:LIBR=BIN:LIBR0/W//
*BIN:LIBR1/O:1
*BIN:LIBR2/O:1
*BIN:LIBR3/O:1
*BIN:LIBR4/O:1
*BIN:LIBR5/O:1
*BIN:LBREM/O:1//
$EOJ
```

4.14 BUILDING FILEX

```
$JOB/RT11/TIME   !FILEX.BAT
        TTYIO
$MES     MAKE FILEX
.R MACRO
*BIN:FILEX,LST:FILEX=SRC:FILEX/C
.R LINK
*BIN:FILEX,LST:FILEX=BIN:FILEX
$EOJ
```

4.15 BUILDING SRCCOM

```
$JOB/RT11/TIME   !SRCCOM.BAT
        TTYIO
$MES     MAKE SRCCOM
.R MACRO
*BIN:SRCCOM,LST:SRCCOM=SRC:SRCCOM/C
.R LINK
*BIN:SRCCOM,LST:SRCCOM=BIN:SRCCOM
$EOJ
```

4.16 BUILDING DUMP

```
$JOB/RT11/TIME   !DUMP.BAT
        TTYIO
$MES     MAKE DUMP
.R MACRO
*BIN:DUMP,LST:DUMP=SRC:DUMP/C
.R LINK
*BIN:DUMP,LST:DUMP=BIN:DUMP
$EOJ
```

4.17 BUILDING THE SYSTEM LIBRARY (SYSF4.OBJ)

```
$JOB/RT11        !SYSF4.BAT
        TTYIO
$MES    MAKE SYSF4
.R MACRO
*BIN:CHAIN.SBJ,LST:CHAIN/N:SYM=SRC:CHAIN
*BIN:CLOSEC.SBJ,LST:CLOSEC/N:SYM=SRC:CLOSEC
*BIN:CMPLT.SBJ,LST:CMPLT=SRC:CMPLT/C
*BIN:CONCAT.SBJ,LST:CONCAT/N:SYM=SRC:CONCAT
*BIN:CVTTIM.SBJ,LST:CVTTIM/N:SYM=SRC:CVTTIM
*BIN:DEVICE.SBJ,LST:DEVICE/N:SYM=SRC:DEVICE
*BIN:DIV60.SBJ,LST:DIV60/N:SYM=SRC:DIV60
*BIN:GTIM.SBJ,LST:GTIM/N:SYM=SRC:GTIM
*BIN:GTJB.SBJ,LST:GTJB/N:SYM=SRC:GTJB
*BIN:GTLIN.SBJ,LST:GTLIN/N:SYM=SRC:GTLIN
*BIN:IADDR.SBJ,LST:IADDR/N:SYM=SRC:IADDR
*BIN:IASIGN.SBJ,LST:IASIGN=SRC:IASIGN/C
*BIN:ICDFN.SBJ,LST:ICDFN/N:SYM=SRC:ICDFN
*BIN:ICHCPY.SBJ,LST:ICHCPY/N:SYM=SRC:ICHCPY
*BIN:ICMKT.SBJ,LST:ICMKT/N:SYM=SRC:ICMKT
*BIN:ICSI.SBJ,LST:ICSI/N:SYM=SRC:ICSI
*BIN:ICSTAT.SBJ,LST:ICSTAT/N:SYM=SRC:ICSTAT
*BIN:IDELET.SBJ,LST:IDELET/N:SYM=SRC:IDELET
*BIN:IDSTAT.SBJ,LST:IDSTAT/N:SYM=SRC:IDSTAT
*BIN:IENTER.SBJ,LST:IENTER/N:SYM=SRC:IENTER
*BIN:IFETCH.SBJ,LST:IFETCH/N:SYM=SRC:IFETCH
*BIN:IFREEC.SBJ,LST:IFREEC/N:SYM=SRC:IFREEC
*BIN:IGETC.SBJ,LST:IGETC/N:SYM=SRC:IGETC
*BIN:IGETSP.SBJ,LST:IGETSP/N:SYM=SRC:IGETSP
*BIN:IJCVT.SBJ,LST:IJCVT/N:SYM=SRC:IJCVT
*BIN:ILUN.SBJ,LST:ILUN/N:SYM=SRC:ILUN
*BIN:INDEX.SBJ,LST:INDEX/N:SYM=SRC:INDEX
*BIN:INSERT.SBJ,LST:INSERT/N:SYM=SRC:INSERT
*BIN:INTSET.SBJ,LST:INTSET=SRC:INTSET/C
*BIN:IPEEK.SBJ,LST:IPEEK/N:SYM=SRC:IPEEK
*BIN:IPEEKB.SBJ,LST:IPEEKB/N:SYM=SRC:IPEEKB
*BIN:IQSET.SBJ,LST:IQSET/N:SYM=SRC:IQSET
*BIN:IRAD50.SBJ,LST:IRAD50/N:SYM=SRC:IRAD50
*BIN:IRCVD.SBJ,LST:IRCVD/N:SYM=SRC:IRCVD
*BIN:IRCVDF.SBJ,LST:IRCVDF/N:SYM=SRC:IRCVDF
*BIN:IREAD.SBJ,LST:IREAD/N:SYM=SRC:IREAD
*BIN:IREADF.SBJ,LST:IREADF/N:SYM=SRC:IREADF
*BIN:IRENAM.SBJ,LST:IRENAM/N:SYM=SRC:IRENAM
*BIN:IREOPN.SBJ,LST:IREOPN/N:SYM=SRC:IREOPN
*BIN:ISAVES.SBJ,LST:ISAVES/N:SYM=SRC:ISAVES
*BIN:ISCHED.SBJ,LST:ISCHED/N:SYM=SRC:ISCHED
*BIN:ISDAT.SBJ,LST:ISDAT/N:SYM=SRC:ISDAT
*BIN:ISDATF.SBJ,LST:ISDATF/N:SYM=SRC:ISDATF
*BIN:ISLEEP.SBJ,LST:ISLEEP/N:SYM=SRC:ISLEEP
*BIN:ISPFN.SBJ,LST:ISPFN/N:SYM=SRC:ISPFN
*BIN:ISPFNF.SBJ,LST:ISPFNF/N:SYM=SRC:ISPFNF
*BIN:ISPY.SBJ,LST:ISPY/N:SYM=SRC:ISPY
*BIN:ITLOCK.SBJ,LST:ITLOCK/N:SYM=SRC:ITLOCK
*BIN:ITTINR.SBJ,LST:ITTINR/N:SYM=SRC:ITTINR
*BIN:ITTOUR.SBJ,LST:ITTOUR/N:SYM=SRC:ITTOUR
*BIN:ITWAIT.SBJ,LST:ITWAIT/N:SYM=SRC:ITWAIT
*BIN:IUNTIL.SBJ,LST:IUNTIL/N:SYM=SRC:IUNTIL
*BIN:IWAIT.SBJ,LST:IWAIT/N:SYM=SRC:IWAIT
*BIN:IWRITE.SBJ,LST:IWRITE/N:SYM=SRC:IWRITE
*BIN:IWRITF.SBJ,LST:IWRITF/N:SYM=SRC:IWRITF
*BIN:JADD.SBJ,LST:JADD/N:SYM=SRC:JADD
*BIN:JCMP.SBJ,LST:JCMP/N:SYM=SRC:JCMP
*BIN:JDIV.SBJ,LST:JDIV/N:SYM=SRC:JDIV
*BIN:JFIX.SBJ,LST:JFIX/N:SYM=SRC:JFIX
*BIN:JFLT.SBJ,LST:JFLT/N:SYM=SRC:JFLT
*BIN:JICVT.SBJ,LST:JICVT/N:SYM=SRC:JICVT
*BIN:JJCVT.SBJ,LST:JJCVT/N:SYM=SRC:JJCVT
```

```
*BIN:JMOV.SBJ,LST:JMOV/N:SYM=SRC:JMOV
*BIN:JMUL.SBJ,LST:JMUL/N:SYM=SRC:JMUL
*BIN:JSUB.SBJ,LST:JSUB/N:SYM=SRC:JSUB
*BIN:JTIME.SBJ,LST:JTIME/N:SYM=SRC:JTIME
*BIN:LEN.SBJ,LST:LEN/N:SYM=SRC:LEN
*BIN:LOCK.SBJ,LST:LOCK/N:SYM=SRC:LOCK
*BIN:LOOKUP.SBJ,LST:LOOKUP/N:SYM=SRC:LOOKUP
*BIN:MRKT.SBJ,LST:MRKT/N:SYM=SRC:MRKT
*BIN:MTSET.SBJ,LST:MTSET/N:SYM=SRC:MTSET
*BIN:MWAIT.SBJ,LST:MWAIT/N:SYM=SRC:MWAIT
*BIN:PRINT.SBJ,LST:PRINT/N:SYM=SRC:PRINT
*BIN:PURGE.SBJ,LST:PURGE/N:SYM=SRC:PURGE
*BIN:R50ASC.SBJ,LST:R50ASC/N:SYM=SRC:R50ASC
*BIN:RCHAIN.SBJ,LST:RCHAIN/N:SYM=SRC:RCHAIN
*BIN:RCTRLO.SBJ,LST:RCTRLO/N:SYM=SRC:RCTRLO
*BIN:REPEAT.SBJ,LST:REPEAT/N:SYM=SRC:REPEAT
*BIN:RESUME.SBJ,LST:RESUME/N:SYM=SRC:RESUME
*BIN:SCCA.SBJ,LST:SCCA/N:SYM=SRC:SCCA
*BIN:SCOMP.SBJ,LST:SCOMP/N:SYM=SRC:SCOMP
*BIN:SCOPY.SBJ,LST:SCOPY/N:SYM=SRC:SCOPY
*BIN:SECNDS.SBJ,LST:SECNDS/N:SYM=SRC:SECNDS
*BIN:SETCMD.SBJ,LST:SETCMD/N:SYM=SRC:SETCMD
*BIN:STRPAD.SBJ,LST:STRPAD/N:SYM=SRC:STRPAD
*BIN:SUBSTR.SBJ,LST:SUBSTR/N:SYM=SRC:SUBSTR
*BIN:SUSPND.SBJ,LST:SUSPND/N:SYM=SRC:SUSPND
*BIN:SYSLBV.SBJ,LST:SYSLBV/N:SYM=SRC:SYSLBV
*BIN:TIMASC.SBJ,LST:TIMASC/N:SYM=SRC:TIMASC
*BIN:TIME.SBJ,LST:TIME/N:SYM=SRC:TIME
*BIN:TIME1.SBJ,LST:TIME1/N:SYM=SRC:TIME1
*BIN:TIMSUB.SBJ,LST:TIMSUB/N:SYM=SRC:TIMSUB
*BIN:TRANSL.SBJ,LST:TRANSL/N:SYM=SRC:TRANSL
*BIN:TRIM.SBJ,LST:TRIM/N:SYM=SRC:TRIM
*BIN:UNLOCK.SBJ,LST:UNLOCK/N:SYM=SRC:UNLOCK
*BIN:VERIFY.SBJ,LST:VERIFY/N:SYM=SRC:VERIFY
.R FORTRA
*BIN:PUTSTR.SBJ,LST:PUTSTR=SRC:PUTSTR/I:THR/S
*BIN:GETSTR.SBJ,LST:GETSTR=SRC:GETSTR/I:THR/S
.R PIP
*BIN:SYSF4.OBJ=BIN:*.SBJ/B/U
$EOJ
```

4.18 BUILDING SYE AND PSE

```
$JOB/RT11/TIME   !SYE.BAT
        TTYIO
$MES    MAKE SYE,PSE
.R FORTRA
*BIN:SYMAN2,LST:SYMAN2=SRC:SYMAN2
*BIN:SYINPT,LST:SYINPT=SRC:SYINPT
*BIN:SYCOML,LST:SYCOML=SRC:SYCOML
*BIN:SYLIN1,LST:SYLIN1=SRC:SYLIN1
*BIN:SYREGN,LST:SYREGN=SRC:SYREGN
*BIN:SYSEQH,LST:SYSEQH=SRC:SYSEQH
*BIN:SYDEV,LST:SYDEV=SRC:SYDEV
*BIN:SYPAR,LST:SYPAR=SRC:SYPAR
*BIN:SYSUM,LST:SYSUM=SRC:SYSUM
.R MACRO
*BIN:SYGRAB,LST:SYGRAB=SRC:SYGRAB/C
*BIN:PSE,LST:PSE=SRC:PSE/C
.R LINK
*BIN:SYE,LST:SYE=BIN:SYMAN2,SYREGN,SYLIN1,SYGRAB/W//
*BIN:SYDEV/O:1
*BIN:SYPAR/O:1
*BIN:SYINPT,SYCOML/O:1
*BIN:SYSEQH/O:1
*BIN:SYSUM/O:1//
*BIN:PSE,LST:PSE=BIN:PSE
$EOJ
```

4.19 BUILDING SYSGEN

```
$JOB/RT11/TIME   !SYSGEN.BAT
        TTYIO
$MES    BUILD SYSGEN
.R FORTRA
*BIN:SYSGEN,LST:SYSGEN=SRC:SYSGEN/I:THR/E/B/W/N:4/V/O/S
.R LINK
*BIN:SYSGEN,LST:SYSGEN=BIN:SYSGEN/W
$EOJ
```

4.20 BUILDING PATCH AND PAT

```
$JOB/RT11/TIME   !PATCH.BAT
        TTYIO
$MES    MAKE PATCH & PAT
.R MACRO
*BIN:PATCH,LST:PATCH=SRC:PATCH/C
*BIN:PAT,LST:PAT=SRC:PAT/C
.R LINK
*BIN:PATCH,LST:PATCH=BIN:PATCH
*BIN:PAT,LST:PAT=BIN:PAT
$EOJ
```

4.21 BUILDING ODT

```
$JOB/RT11/TIME   !ODT.BAT
        TTYIO
$MES MAKE ODT
.R MACRO
*BIN:ODT,LST:ODT=SRC:ODT/C
$EOJ
```

4.22 BUILDING THE VT11 LIBRARY AND PLOT55

```
$JOB/RT11/TIME   !VTLIB.BAT
        TTYIO
$MES    MAKE VTLIB & PLOT55
.R MACRO
*BIN:VTCAL1,LST:VTCAL1=SRC:VTCAL1/C
*BIN:VTCAL2,LST:VTCAL2=SRC:VTCAL2/C
*BIN:VTCAL3,LST:VTCAL3=SRC:VTCAL3/C
*BIN:VTCAL4,LST:VTCAL4=SRC:VTCAL4/C
*BIN:VTBASE,LST:VTBASE=SRC:VTBASE/C
*BIN:PLOT55,LST:PLOT55=SRC:PLOT55/C
.R PIP
*BIN:VTHDLR.OBJ=BIN:VTCAL1.OBJ,VTCAL2.OBJ,VTCAL3.OBJ,VTCAL4.OBJ,VTBASE.OBJ/B/U
$EOJ
```

4.23 BUILDING FORMAT

```
$JOB/RT11/TIME   !FORMAT.BAT
        TTYIO
$MES    MAKE FORMAT
.R MACRO
*BIN:FORMAT,LST:FORMAT=SRC:FORMAT/C
*BIN:FMTDY,LST:FMTDY=SRC:FMTDY/C
*BIN:FMTRK,LST:FMTRK=SRC:FMTRK/C
.R LINK
*BIN:FORMAT,LST:FORMAT=SRC:FORMAT,FMTDY,FMTRK/W
$EOJ
```

4.24 BUILDING RESORC

```
$JOB/RT11/TIME   !RESORC.BAT
        TTYIO
$MES    MAKE RESORC
.R MACRO
*BIN:RESORC,LST:RESORC/C=SRC:RESORC
*BIN:RESRC1,LST:RESRC1/C=SRC:DIRMAC/M,RESRC1
.R LINK
*BIN:RESORC,LST:RESORC=BIN:RESORC,RESRC1/U:100
*PATCH
$EOJ
```

4.25 BUILDING SWAP

```
$JOB/RT11/TIME   !SWAP.BAT
        TTYIO
$MES    BUILD SWAP.SYS
.R MACRO
*BIN:SWAP=SRC:MEND
.R LINK
*BIN:SWAP.SYS=BIN:SWAP/B:30000
$EOJ
```

4.26 BUILDING TECO

```
$JOB/RT11/TIME   !TECO.BAT
        TTYIO
$MES    MAKE TECO
.R MACRO
*BIN:TIO,LST:TIO=SRC:TIO/C
.R LINK
*BIN:TECO,LST:TECO=BIN:TIO,SRC:TECO/W
$EOJ
```

APPENDIX A

EXAMPLE SYSGEN'S

This appendix contains the actual console terminal output generated during three RT-11 system generations. The first uses the expanded form of dialogue to generate a base-line SJ monitor for RX01 diskette, the second uses the short form to build SJ, FB, and XM monitors for RL01 with almost all functionality included. The third example shows the complete procedure for undertaking a SYSGEN on a system with two RX01 diskette drives.

A.1 BASE-LINE RX01 DISKETTE SJ GENERATION

This example shows the actual SYSGEN dialogue and the resulting SYCND file.

```
.R SYSGEN (RET)


RT-11 SYSTEM GENERATION PROGRAM V02.11

Do you want the expanded form of dialogue [Y/N] (Y)? (RET)


All answers requiring a choice are indicated by [Y/N]
and require you to type either 'Y' for yes or 'N'
for no followed by a carriage return.  Answers requiring
numeric input are indicated by [D] for decimal input and
[O] for octal input.  Answers requiring a device name
response are indicated by [DD]; answers requiring a
device name followed by a unit number are indicated
by [DDU]. If you enter no response (just a CARRIAGE RETURN),
your answer will default to the response indicated in
parentheses before the question mark (e.g. [DD] (RK)?).
Your responses are checked for validity wherever possible.

The outputs of this SYSGEN program are two conditional
files (SYCND.MAC and SYSTBL.MAC) and three indirect
command files (SYSBLD.COM, MONBLD.COM and DEVBLD.COM)
which are used to assemble and link the monitor(s) and
device handlers for the target system. The assembly
and linking process requires a machine 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.
```

EXAMPLE SYSGEN'S

```
The following dialogue asks questions about
the configuration for which the system is being
generated, and about monitor and device support
options you may desire.  You should have read the
RT-11 System Generation Manual before proceeding.
You can terminate SYSGEN at any time by typing CTRL/C.

The RT-11 system includes a variety of monitors, allowing
you to select the monitor best suited to your
application.  The monitors are customized to your configuration
by this SYSGEN process, and, while you may build any number of
monitors, you should realize that SYSGEN places a large
demand on time and data storage.

If you elect to build the base-line
single-job (SJ) monitor, no other monitors
may be built during the same SYSGEN pass.

The base-line single-job (SJ) monitor is the minimal monitor
necessary to execute standard system functions. It is intended
for systems with only 8K words of memory, or for larger
systems that require the lowest possible monitor overhead.
Device support is limited, and some features of the
SJ monitor usable in larger configurations are omitted,
such as BATCH and VT11 support.


Do you want the base-line single-job monitor [Y/N] (N)? Y RET


The following options apply to all monitors.

RT-11 assumes a 60 Hz power line frequency, but can
optionally support a 50 Hz frequency.


Do you want the optional 50 Hz clock support [Y/N] (N)? RET


The bootstrap optionally executes an indirect file when
starting the system.


Do you want the startup indirect file [Y/N] (Y)? RET


The following questions ask about the
peripheral devices that are to be supported by the
generated system. By selecting support for a specific
device, you make the device known to the monitor's
device tables, and you cause a device handler
to be generated for the device. If you do not select
a specific device, you will have to create the
device handler separately and use the monitor
INSTALL command when device access is desired.

RT-11 supports the following devices as system devices:
        RK   RK05 Cartridge Disk
        DT   DECtape
        DX   Flexible Diskette

You must select one as your system device; use the
two character device name as indicated above.
```

```
What is the name of the system device [DD] (RK)? DX RET


The RX11 flexible diskette subsystem supports only two
drives for each controller. If your system contains four
diskette drives, there is a second controller present
supporting the third and fourth units.


Do you want support for a second RX11 controller [Y/N] (N)? RET


What is the CSR address for the
   first RX11 controller [0] (177170)? RET


What is the vector address for the first RX11 controller [0] (264)? RET


Do you want RK05/RK05F disk support [Y/N] (N)? RET


Do you want TC11 DECtape support [Y/N] (N)? RET


Do you want line printer support [Y/N] (Y)? RET


The line printer vector address is normally 200
and the CSR address is normally 177514. If your printer
is installed at another vector or CSR address, you
must specify the correct values.


Does your printer have a non-standard vector or CSR address [Y/N] (N)? RET


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.


Do you want PC11 high-speed paper tape reader/punch support [Y/N] (N)? RET


Do you want PR11 high-speed paper tape reader support [Y/N] (N)? RET


The null handler is a mechanism that allows logical
input and output to take place without actual physical
data transfers. It is most useful in debugging applications.


Do you want the null handler [Y/N] (Y)? RET

This completes the device query. Space in the monitor is allocated
only for the devices specified, but you may 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.
```

```
How many extra device slots do you want [D] (0)? RET


The indirect command files generated by the SYSGEN program
contain commands that will delete the object modules
from which the system is built when they are
no longer needed. This measure serves to conserve disk
space while the system is being built. It is often useful,
however, to keep the object modules for later use in
patching the system. If your output device
has sufficient free space,you can elect to retain all
system object modules for future use.


Do you want to retain the system OBJ's [Y/N] (N)? RET


The SYSGEN assembling and linking process requires a
source input 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
Section 3.6.1 of the RT-11 System Generation Manual.


What is the PHYSICAL name and unit of the
   source input device [DDU] (RK1)? DX1 RET


What is the PHYSICAL name and unit of the
   binary output device [DDU] (RK0)? DX0 RET


What is the PHYSICAL name and unit of the
   map output device [DDU] (TT)? RET


To build an entire system, mount the source and
binary disks, copy the files SYCND.MAC and SYSTBL.MAC
to the source disk, and type "@SYSBLD". To build just the monitors,
type "@MONBLD". To build just the device handlers, type
"@DEVBLD". For more information, read Section 3.6 of the
RT-11 System Generation Manual.


END OF SYSGEN PROGRAM--Don't forget to copy DK:SYCND.MAC and
        DK:SYSTBL.MAC to the source disk before executing the
        command file(s). Remember that SWAP.SYS must
        be on all bootable volumes.
```

```
.SBTTL  SYSTEM CONDITIONAL FILE
        ; SYCND.MAC--SYSTEM CONDITIONAL FILE
        ;
        ; COPYRIGHT (C)    1978
        ;
        ; DIGITAL EQUIPMENT CORPORATION,  MAYNARD,   MASS.

SYSG$N  = 1                ;INDICATE SYSGENED MONITOR
 ; BASE-LINE SJ MONITOR
TIME$R  = 0                ;NO TIMER SUPPORT
MPT$Y   = 0                ;NO MEMORY PARITY SUPPORT
BATC$H  = 0                ;NO BATCH SUPPORT
ERL$G   = 0                ;NO ERROR LOG SUPPORT
ESC$P   = 0                ;NO ESCAPE SEQUENCE SUPPORT
RDF$L   = 0                ;NO MESSAGE ON SYS I/O ERRORS (SJ)
CLOCK   = 60.              ;POWER LINE FREQUENCY (SJ,FB,XM)
STAR$T  = 1                ;STARTUP COMMAND FILE (SJ,FB,XM)
$DXSYS  = 1                ;SYSTEM DEVICE
DX$CSR  = 177170           ;CSR OF FIRST RX11
DX$VEC  = 264              ;VECTOR OF FIRST RX11
```

A.2 RL01, SJ, FB, AND XM GENERATION

This example shows the actual SYSGEN dialogue and the resulting files: SYCND, SYSBLD, MONBLD, and DEVBLD.

```
.R SYSGEN RET


RT-11 SYSTEM GENERATION PROGRAM V02.11

Do you want the expanded form of dialogue [Y/N] (Y)? YES RET
```

```
All answers requiring a choice are indicated by [Y/N]
and require you to type either 'Y' for yes or 'N'
for no followed by a carriage return.  Answers requiring
numeric input are indicated by [D] for decimal input and
[O] for octal input.  Answers requiring a device name
response are indicated by [DD]; answers requiring a
device name followed by a unit number are indicated
by [DDU]. If you enter no response (just a CARRIAGE RETURN),
your answer will default to the response indicated in
parentheses before the question mark (e.g. [DD] (RK)?).
Your responses are checked for validity wherever possible.

The outputs of this SYSGEN program are two conditional
files (SYCND.MAC and SYSTBL.MAC) and three indirect
command files (SYSBLD.COM, MONBLD.COM and DEVBLD.COM)
which are used to assemble and link the monitor(s) and
device handlers for the target system. The assembly
and linking process requires a machine 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.


The following dialogue asks questions about
the configuration for which the system is being
generated, and about monitor and device support
options you may desire.  You should have read the
RT-11 System Generation Manual before proceeding.
You can terminate SYSGEN at any time by typing CTRL/C.

The RT-11 system includes a variety of monitors, allowing
you to select the monitor best suited to your
application.  The monitors are customized to your configuration
by this SYSGEN process, and, while you may build any number of
monitors, you should realize that SYSGEN places a large
demand on time and data storage.


If you elect to build the base-line
single-job (SJ) monitor, no other monitors
may be built during the same SYSGEN pass.

The base-line single-job (SJ) monitor is the minimal monitor
necessary to execute standard system functions. It is intended
for systems with only 8K words of memory, or for larger
systems that require the lowest possible monitor overhead.
Device support is limited, and some features of the
SJ monitor usable in larger configurations are omitted,
such as BATCH and VT11 support.

Do you want the base-line single-job monitor [Y/N] (N)? NO (RET)


The single-job (SJ) monitor provides an environment suitable
for the development of simple applications in FORTRAN or
BASIC.  It provides a subset of the foreground/background
(FB) features and offers maximum ability for minimum size.
Some additional features may be added during SYSGEN, but at
the cost of increased monitor size.
```

```
Do you want the single-job (SJ) monitor [Y/N] (Y)? YES RET


The foreground/background (FB) monitor offers an extended
set of facilities which are helpful for applications
involving real-time or data acquisition. These
include timer support, serialized asynchronous I/O, and
support for a foreground job.


Do you want the foreground/background (FB) monitor [Y/N] (Y)? YES RET


The extended memory (XM) monitor is a version of the FB
monitor which supports up to 128K words of memory. It is
useful for advanced applications that require an extended
addressing space.  The XM monitor is larger than FB and
requires a resident USR and drivers.


Do you want the extended memory (XM) monitor [Y/N] (N)? YES RET


There are a number of monitor options you may select:


The following options apply only to the SJ monitor.

The SJ monitor normally does not have timer support.
You may optionally include the .MRKT and .CMKT requests.


Do you want timer support in the SJ monitor [Y/N] (N)? YES RET


The device time-out option permits device drives to issue
mark time requests.  This feature is currently not used
by any RT-11 drivers, but is necessary for DECNET support.
Select this option if you are planning to use DECNET
with your monitor.


Do you want device time-out support [Y/N] (N)? NO RET


The SJ monitor will normally halt at the "system halt"
if there is a fatal system I/O error. This halt can
be replaced with a system error message at the cost of
about 30 words additional code in the resident
monitor. This option is highly recommended if the system
is going to be used by anyone other than the expert.


Do you want an error message on system I/O errors [Y/N] (Y)? YES RET


The following options apply only to the FB and XM monitors.

The FB and XM monitors' scheduler idle loop will display
a moving light pattern in the display register of any
PDP-11/45, 11/50, 11/55 or 11/70 processor.
```

```
Do you want idle loop light pattern [Y/N] (N)? NO (RET)


RT-11 normally supports a single terminal interfaced through
the console KL11 or DL11, which is shared by both background
and foreground jobs. You may optionally select multi-
terminal support of up to 16 terminals interfaced through
a choice of DL11 and DZ11 interfaces.


Do you want support of more than one terminal [Y/N] (N)? YES (RET)


The following options apply to all monitors.

RT-11 assumes a 60 Hz power line frequency, but can
optionally support a 50 Hz frequency.


Do you want the optional 50 Hz clock support [Y/N] (N)? NO (RET)


RT-11 normally uses a line clock for the system clock.
You may substitute the KW11-P programmable clock if you wish,
in which case the KW11-P is not available for program use.


Do you want to use the KW11-P clock as the system clock [Y/N] (N)? NO (RET)


The bootstrap optionally executes an indirect file when
starting the system.


Do you want the startup indirect file [Y/N] (Y)? YES (RET)


RT-11 supports parity memory and processes memory
parity errors.  If error logging is present, parity errors
are also logged.


Do you want memory parity support [Y/N] (N)? NO (RET)


The monitor normally halts on power recovery startup
after a power failure.  You may elect to have the monitor
print a message explaining that power failure caused the halt.


Do you want power failure messages [Y/N] (Y)? YES (RET)


RT-11 supports BATCH as an option. Indirect command
file support is separately available in all monitors,
and BATCH support need not be selected to obtain
indirect command file support.


Do you want BATCH support [Y/N] (N)? YES (RET)
```

```
The RT-11 terminal service is capable of recognizing
ANSI or DEC standard escape sequences and processing
those sequences according to application instructions.
This support is not necessary for normal system operation;
if your application does not make special use of escape
sequences, do not select this support.


Do you want escape sequence processing [Y/N] (N)? NO (RET)


Selection of error logging creates the error logging
driver EL and incorporates error logging into the device
handlers. When EL is loaded and initialized, logging of
disk, DECtape and memory parity errors will occur.


Do you want error logging [Y/N] (N)? NO (RET)


The following questions ask about the
peripheral devices that are to be supported by the
generated system. By selecting support for a specific
device, you make the device known to the monitor's
device tables, and you cause a device handler
to be generated for the device. If you do not select
a specific device, you will have to create the
device handler separately and use the monitor
INSTALL command when device access is desired.


RT-11 supports the following devices as system devices:
        RK   RK05 Cartridge Disk
        DL   RL01 Cartridge Disk
        RF   RF11 Fixed-head Disk
        DT   DECtape
        DX   RX01 Flexible Diskette
        DY   RX02 Double Density Diskette
        DS   RJS03/4 Massbus Fixed-head Disk
        DP   RP11 Disk Pack
        DM   RK06/RK07 Cartridge Disk

You must select one as your system device; use the
two character device name,as indicated above.


What is the name of the system device [DD] (RK)? DL (RET)


Do you want RJS03 or RJS04 disk support [Y/N] (N)? NO (RET)


Do you want RF11 fixed-head disk support [Y/N] (N)? NO (RET)


Do you want RP11/RPR02 or RP11/RP03 disk support [Y/N] (N)? NO (RET)


Do you want RX01/RX11 flexible diskette support [Y/N] (N)? YES (RET)


The RX11 flexible diskette subsystem supports only two
drives for each controller. If your system contains four
diskette drives, there is a second controller present
supporting the third and fourth units.
```

```
Do you want support for a second RX11 controller [Y/N] (N)? YES (RET)

What is the CSR address for the
   first RX11 controller [O] (177170)? 177170 (RET)

What is the vector address for the first RX11 controller [O] (264)? 264 (RET)

What is the CSR address for the
   second RX11 controller [O] (177150)? 176200 (RET)

What is the vector address for the second RX11 controller [O] (270)? 320 (RET)

Do you want RX02 diskette support [Y/N] (N)? NO (RET)

How many RL01 units are to be supported [D] (2)? 2 (RET)

Do you want RK05/RK05F disk support [Y/N] (N)? NO (RET)

Do you want TC11 DECtape support [Y/N] (N)? NO (RET)

Do you want RK06/RK07 disk support [Y/N] (N)? NO (RET)

Do you want magnetic tape support [Y/N] (N)? YES (RET)

RT-11 magtape support is available in two forms. The standard
form is file-structured magtape, which supports a subset of
ANSI file format. The file-structured handler may be used to
store and retrieve data in a file format usable with most
RT-11 system programs. The second form of magtape support
omits file structuring and reads and writes data directly in variable
length records. This type of support is called the "hardware
handler", and while this handler is significantly smaller than
the file-structured handler, it can not be used with any
system programs. The file-structured handler is also
capable of performing direct hardware functions.


How many extra device slots do you want [D] (0)? 1 (RET)


The indirect command files generated by the SYSGEN program
contain commands that will delete the object modules
from which the system is built when they are
no longer needed. This measure serves to conserve disk
space while the system is being built. It is often useful,
however, to keep the object modules for later use in
patching the system. If your output device
has sufficient free space,you can elect to retain all
system object modules for future use.
```

```
Do you want the file-structured magtape handler [Y/N] (Y)? YES (RET)

The file-structured 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.

How many magtape units are to be supported [D] (2)? 2 (RET)

Do you want TM11 (UNIBUS) magtape support [Y/N] (Y)? YES (RET)

Do you want TA11 cassette support [Y/N] (N)? NO (RET)

Do you want VT11 or VS60 graphics support [Y/N] (N)? NO (RET)

Do you want line printer support [Y/N] (Y)? YES (RET)

The line printer vector address is normally 200
and the CSR address is normally 177514. If your printer
is installed at another vector or CSR address, you
must specify the correct values.

Does your printer have a non-standard vector or CSR address [Y/N] (N)? NO (RET)

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.

Do you want PC11 high-speed paper tape reader/punch support [Y/N] (N)? NO (RET)

Do you want PR11 high-speed paper tape reader support [Y/N] (N)? NO (RET)

Do you want card reader support [Y/N] (N)? NO (RET)

The null handler is a mechanism that allows logical
input and output to take place without actual physical
data transfers. It is most useful in debugging applications.

Do you want the null handler [Y/N] (Y)? YES (RET)

This completes the device query. Space in the monitor is allocated
only for the devices specified, but you may 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.
```

```
Do you want to retain the system OBJ's [Y/N] (N)? YES (RET)


The following questions ask about 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. Physical unit numbers of the DL11 lines are assigned
first to local lines and then to remote lines. This assignment
is permanent and its order cannot be changed.


After the dialogue has established the number of lines of each
type, the CSR and vector addresses for each line must be
established. The first 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 response applies to the first remote
line, the second to the second remote line, etc. until all
remote lines are accounted for.


How many local DL11's, including the console [D] (1)? 1 (RET)


How many remote DL11 lines [D] (0)? 0 (RET)


What is the CSR address for the
   first (console) DL11 [O] (177560)? 177560 (RET)


What is the vector address for the first (console) DL11 [O] (60)? 60 (RET)


RT-11 supports one DZ11-A or -B eight line multiplexor
with an additional DZ11-C or -D eight line multiplexor
for a maximum of 16 lines,or one DZV11 four line
multiplexor.  Modem support is available on
remote lines, but is limited to Bell 103 type modems or
equivalent with the "common clear to send and carrier"
option installed to operate the modem in an auto-answer mode.


Do you want DZ11 or DZV11 multiplexor support [Y/N] (N)? YES (RET)


Do you want DZ11 multiplexor support [Y/N] (Y)? YES (RET)


Do you want support for two DZ11's [Y/N] (N)? NO (RET)


The lines must be initialized to a specific baud
rate. The choices are 300, 150, and 110 baud.


Do you want the lines initialized to 300 baud [Y/N] (Y)? YES (RET)
```

```
The vectors and CSR addresses are assigned to the floating
device region and will vary with each installation.


What is the CSR address for the first DZ11 [O] (160010)? 160100 (RET)


What is the vector address for the first DZ11 [O] (300)? 300 (RET)


The following dialogue will ask 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. Physical unit numbers of the DZ11 or DZV11 lines
are assigned first to local lines and then to remote
lines. This assignment is permanent and its order cannot
be changed.


How many DZ11 lines are local terminals [D] (0)? 4 (RET)


How many remote DZ11 lines [D] (0)? 2 (RET)


The terminal service requires a set of input and output
ring buffers for each terminal supported. The output ring
buffer default size is 40 characters. The input ring size
depends on terminal width and amount of type-ahead
expected. The default size is 82 characters.


Size of the output buffers in characters [D] (40)? 40 (RET)


Size of the input buffers in characters [D] (82)? 82 (RET)


Terminal status may be obtained by doing an .MTGET request
but some applications require asynchronous notification of
changes in terminal status. RT-11 supports asynchronous
terminal status as an option to the .MTATCH
request and provides information on buffer and modem
status as well as double CTRL/C.


Do you want asynchronous terminal status [Y/N] (Y)? YES (RET)


Time-out support will cause the monitor to reset at regular
intervals any terminal that may have gone off-line. This
helps to minimize the impact of static (and similar)
problems, and is recommended if maximum availability
is important to the application.


Do you want multi-terminal time-out support [Y/N] (Y)? YES (RET)


The SYSGEN assembling and linking process requires a
source input 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
Section 3.6.1 of the RT-11 System Generation Manual.
```

```
What is the PHYSICAL name and unit of the
   source input device [DDU] (RK1)? DL0 RET


What is the PHYSICAL name and unit of the
   binary output device [DDU] (RK0)? DL0 RET


What is the PHYSICAL name and unit of the
   map output device [DDU] (TT)? LP RET


To build an entire system, mount the source and
binary disks, copy the files SYCND.MAC and SYSTBL.MAC
to the source disk, and type "@SYSBLD". To build just the monitors,
type "@MONBLD". To build just the device handlers, type
"@DEVBLD". For more information, read Section 3.6 of the
RT-11 System Generation Manual.


END OF SYSGEN PROGRAM--Don't forget to copy DK:SYCND.MAC and
         DK:SYSTBL.MAC to the source disk before executing the
         command file(s). Remember that SWAP.SYS must
         be on all bootable volumes.


.SBTTL  SYSTEM CONDITIONAL FILE
        ; SYCND.MAC--SYSTEM CONDITIONAL FILE
        ;
        ; COPYRIGHT (C)    1978
        ;
        ; DIGITAL EQUIPMENT CORPORATION,  MAYNARD,   MASS.

SYSG$N  = 1             ;INDICATE SYSGENED MONITOR
TIME$R  = 1             ;ADD TIMER SUPPORT (SJ)
RDF$L   = 1             ;ADD SYSTEM I/O ERROR MESSAGE (SJ)
MTT$Y   = 1             ;MULTIPLE TERMINAL SUPPORT (FB,XM)
CLOCK   = 60.           ;POWER LINE FREQUENCY (SJ,FB,XM)
STAR$T  = 1             ;STARTUP COMMAND FILE (SJ,FB,XM)
PWF$L   = 1             ;POWER FAIL (SJ,FB,XM)
BATC$H  = 1             ;BATCH SUPPORT (SJ,FB,XM)
ESC$P   = 0             ;NO ESCAPE SEQUENCES (SJ,FB,XM)
$DLSYS  = 1             ;SYSTEM DEVICE
DX$CSR  = 177170        ;CSR OF FIRST RX11
DX$VEC  = 264           ;VECTOR OF FIRST RX11
DXT$O   = 1             ;SECOND RX11 CONTROLLER
DX$CS2  = 176200        ;CSR OF SECOND RX11
DX$CV2  = 320           ;VECTOR OF SECOND RX11
DL$UN   = 2             ;NUMBER OF RL01 UNITS
FILES   = 1             ;MAGTAPE FILES SUPPORT
DRIVEN  = 2.            ;NUMBER OF MAGTAPE UNITS
DL11$L  = 1.            ;# OF LOCAL DL11 INTERFACES (MULTI-TERM SUPPORT)
DL11$M  = 0.            ;# OF REMOTE DL11-E LINES (MULTI-TERM SUPPORT)
DL11$N  = 1. + 0.       ;TOTAL # OF DL11 LINES (MULTI-TERM SUPPORT)
DLC$0   = 177560        ;CSR OF 1ST (CONSOLE) DL11 (MULTI-TERM SUPPORT)
DLV$0   = 60           .;VECTOR OF 1ST (CONSOLE) DL11 (MULTI-TERM SUPPORT)
DZ11$N  = 1             ;# OF DZ11 MULTIPLEXORS (MULTI-TERM SUPPORT)
DZSP$D  = 2400          ;INIT TO 300 BAUD (MULTI-TERM SUPPORT)
DZST$P  = 0             ;1 STOP UNIT (MULTI-TERM SUPPORT)
DZVC$0  = 300           ;VECTOR OF FIRST DZ11 (MULTI-TERM SUPPORT)
DZCS$0  = 160100        ;CSR OF FIRST DZ11 (MULTI-TERM SUPPORT)
```

```
DZ11$L   = 4.             ;# OF LOCAL DZ11 LINES (MULTI-TERM SUPPORT)
DZ11$M   = 2.             ;# OF REMOTE DZ11 LINES (MULTI-TERM SUPPORT)
TIM$IT   = 1              ;GENERAL DEVICE TIME OUT (MULTI-TERM SUPPORT)
TTYOUT   = 40.            ;SIZE OF OUTPUT RING BUFFERS(MULTI-TERM SUPPORT)
TTYIN    = 82.            ;SIZE OF INPUT RING BUFFERS (MULTI-TERM SUPPORT)
MAT$S    = 1              ;ASYNCHRONOUS TERMINAL STATUS (MULTI-TERM SUPPORT)
MTI$M    = 1              ;TERMINAL TIME OUT (MULTI-TERM SUPPORT)
```

```
!        MONBLD.COM
!
ASS DLO SRC
ASS DLO BIN
ASS LP MAP
MACRO/OBJ:BIN:KMSJ SRC:(SJ+SYCND+EDTGBL+KMON+KMOVLY)
MACRO/OBJ:BIN:RMSJ SRC:(SJ+SYCND+EDTGBL+USR+RMONSJ)
MACRO/OBJ:BIN:SYSJ SRC:(SJ+SYCND+SYSDEV+DL)
MACRO/OBJ:BIN:TBSJ SRC:(SJ+SYCND+SYSTBL)
MACRO/OBJ:BIN:BTSJ SRC:(SJ+SYCND+BSTRAP)
LINK/EXE:BIN:DLMNSJ.SYG/BOU:1000/MAP:MAP:DLMNSJ
BIN:(BT,RM,KM,TB,SY)SJ
OVLYO
MACRO/OBJ:BIN:KMFB SRC:(FB+SYCND+EDTGBL+KMON+KMOVLY)
MACRO/OBJ:BIN:RMFB SRC:(FB+SYCND+EDTGBL+USR+RMONFB)
MACRO/OBJ:BIN:SYFB SRC:(FB+SYCND+SYSDEV+DL)
MACRO/OBJ:BIN:TBFB SRC:(FB+SYCND+SYSTBL)
MACRO/OBJ:BIN:BTFB SRC:(FB+SYCND+BSTRAP)
MACRO/OBJ:BIN:MEFB SRC:(FB+SYCND+MTTEMT)
MACRO/OBJ:BIN:MIFB SRC:(FB+SYCND+MTTINT)
LINK/EXE:BIN:DLMNFB.SYG/BOU:1000/PROMPT/MAP:MAP:DLMNFB BIN:BTFB
BIN:RMFB,KMFB,MEFB,MIFB,TBFB,SYFB//
OVLYO
MACRO/OBJ:BIN:KMXM SRC:(XM+SYCND+EDTGBL+KMON+KMOVLY)
MACRO/OBJ:BIN:RMXM SRC:(XM+SYCND+EDTGBL+USR+RMONFB)
MACRO/OBJ:BIN:SYXM SRC:(XM+SYCND+SYSDEV+DL)
MACRO/OBJ:BIN:TBXM SRC:(XM+SYCND+SYSTBL)
MACRO/OBJ:BIN:BTXM SRC:(XM+SYCND+BSTRAP)
MACRO/OBJ:BIN:MEXM SRC:(XM+SYCND+MTTEMT)
MACRO/OBJ:BIN:MIXM SRC:(XM+SYCND+MTTINT)
LINK/EXE:BIN:DLMNXM.SYG/BOU:1000/PROMPT/MAP:MAP:DLMNXM BIN:BTXM
BIN:RMXM,KMXM,MEXM,MIXM,TBXM,SYXM//
OVLYO
```

```
!        DEVBLD.COM
!
ASS DLO SRC
ASS DLO BIN
MACRO/OBJ:BIN:TT SRC:(SYCND+TT)
LINK/EXE:BIN:TT.SYG BIN:TT
MACRO/OBJ:BIN:BA SRC:(SYCND+BA)
LINK/EXE:BIN:BA.SYG BIN:BA
MACRO/OBJ:BIN:BAX SRC:(XM+SYCND+BA)
LINK/EXE:BIN:BAX.SYG BIN:BAX
MACRO/OBJ:BIN:LP SRC:(SYCND+LP)
LINK/EXE:BIN:LP.SYG BIN:LP
MACRO/OBJ:BIN:LPX SRC:(XM+SYCND+LP)
LINK/EXE:BIN:LPX.SYG BIN:LPX
MACRO/OBJ:BIN:DL SRC:(SYCND+DL)
LINK/EXE:BIN:DL.SYG BIN:DL
MACRO/OBJ:BIN:DLX SRC:(XM+SYCND+DL)
LINK/EXE:BIN:DLX.SYG BIN:DLX
MACRO/OBJ:BIN:DX SRC:(SYCND+DX)
LINK/EXE:BIN:DX.SYG BIN:DX
MACRO/OBJ:BIN:DXX SRC:(XM+SYCND+DX)
```

```
LINK/EXE:BIN:DXX.SYG BIN:DXX
MACRO/OBJ:BIN:NL SRC:(SYCND+NL)
LINK/EXE:BIN:NL.SYG BIN:NL
MACRO/OBJ:BIN:NLX SRC:(XM+SYCND+NL)
LINK/EXE:BIN:NLX.SYG BIN:NLX
MACRO/OBJ:BIN:MT SRC:(SYCND+TM)
MACRO/OBJ:BIN:FSM SRC:(SYCND+FSM)
LINK/EXE:BIN:MT.SYG BIN:(MT,FSM)
MACRO/OBJ:BIN:MTX SRC:(XM+SYCND+TM)
MACRO/OBJ:BIN:FSMX SRC:(XM+SYCND+FSM)
LINK/EXE:BIN:MTX.SYG BIN:(MTX,FSMX)
```

A.3 DISKETTE SYSGEN PROCEDURE

If you find that none of the distributed RT-11 monitors meets the needs of your application and no customization patches are suitable either, you must perform a system generation to create the appropriate monitor and device handlers. If your system has two RX01 diskette drives and no disk device of larger capacity, you can still perform a system generation. However, the process is not completely straightforward, and you should read and study this entire section before attempting to generate your system.

Because of the severe space constraints of the dual diskette system, it is possible to generate only one monitor and its associated handlers at a time. You should answer the SYSGEN dialogue questions appropriately for all the monitors you need to create. However, you can assemble only one monitor at a time. DIGITAL recommends that you use the base-line single-job monitor when you perform the system generation. You can use the single-job monitor, but the base-line monitor makes more free space available on your system diskette.

The following procedure is an example of one method you can use to organize files on three diskettes and successfully assemble an RT-11 system.

First, create a working system diskette from the files on the distribution kit, following the procedures outlined in Chapter 2. Your system disk must include the following files:

```
monitor
SWAP.SYS
TT.SYS
LP.SYS - if a line printer available
SYSGEN.SAV
SYSGEN.CND
SYSTBL.CND
DIR.SAV
PIP.SAV
DUP.SAV
```

Remember to copy the bootstrap onto this system disk by using the COPY/BOOT command. Bootstrap the system and run from this system diskette.

EXAMPLE SYSGEN'S

Following the directions in Chapter 3, invoke the SYSGEN dialogue program. Read each question carefully and type the appropriate response for your application. When the dialogue program completes, the following new files should appear in the directory of your system diskette:

```
SYCND.MAC
SYSTBL.MAC
SYSBLD.COM
MONBLD.COM
DEVBLD.COM
```

Delete SYSBLD.COM from the system diskette.

```
Type:   DELETE SYSBLD.COM/NOQUERY  RET
```

Copy the files MONBLD.COM and DEVBLD.COM to the terminal, or to the line printer if your system has one.

```
Type: TYPE MONBLD.COM,DEVBLD.COM RET  (if no line printer)
      PRINT MONBLD.COM,DEVBLD.COM RET (if line printer available)
```

Study these two files. They contain the names of all the files you need in order to perform the assemblies, as well as the keyboard monitor commands to invoke the assemblies and the link. Because the required files do not all fit on two diskettes at once, you must arrange the files so that you can copy them to your system disk a few at a time and perform the component assemblies separately.

The following sample dialogue creates an RX01 foreground/background monitor with multi-terminal support.

```
.R SYSGEN RET


RT-11 SYSTEM GENERATION PROGRAM V02.11

Do you want the expanded form of dialogue [Y/N] (Y)? N RET


Do you want the base-line single-job monitor [Y/N] (N)? N RET


Do you want the single-job (SJ) monitor [Y/N] (Y)? N RET


Do you want the foreground/background (FB) monitor [Y/N] (Y)? Y RET
```

EXAMPLE SYSGEN'S

```
Do you want the extended memory (XM) monitor [Y/N] (N)? N RET

Do you want idle loop light pattern [Y/N] (N)? N RET

Do you want support of more than one terminal [Y/N] (N)? Y RET

Do you want the optional 50 Hz clock support [Y/N] (N)? N RET

Do you want to use the KW11-P clock as the system clock [Y/N] (N)? N RET

Do you want the startup indirect file [Y/N] (Y)? Y RET

Do you want memory parity support [Y/N] (N)? N RET

Do you want power failure messages [Y/N] (Y)? Y RET

Do you want BATCH support [Y/N] (N)? N RET

Do you want escape sequence processing [Y/N] (N)? N RET

Do you want error logging [Y/N] (N)? N RET

What is the name of the system device [DD] (RK)? DX RET

Do you want RJS03 or RJS04 disk support [Y/N] (N)? N RET

Do you want RF11 fixed-head disk support [Y/N] (N)? N RET

Do you want RP11/RPR02 or RP11/RP03 disk support [Y/N] (N)? N RET

Do you want support for a second RX11 controller [Y/N] (N)? N RET

What is the CSR address for the
   first RX11 controller [O] (177170)? 177170 RET
```

```
What is the vector address for the first RX11 controller [O] (264)? 264 RET

Do you want RX02 diskette support [Y/N] (N)? N RET

Do you want RL01 disk support [Y/N] (N)? N RET

Do you want RK05/RK05F disk support [Y/N] (N)? N RET

Do you want TC11 DECtape support [Y/N] (N)? N RET

Do you want RK06/RK07 disk support [Y/N] (N)? N RET

Do you want magnetic tape support [Y/N] (N)? N RET

Do you want TA11 cassette support [Y/N] (N)? N RET

Do you want VT11 or VS60 graphics support [Y/N] (N)? N RET

Do you want line printer support [Y/N] (Y)? Y RET

Does your printer have a non-standard vector or CSR address [Y/N] (N)? N RET

Do you want PC11 high-speed paper tape reader/punch support [Y/N] (N)? N RET

Do you want PR11 high-speed paper tape reader support [Y/N] (N)? N RET

Do you want card reader support [Y/N] (N)? N RET

Do you want the null handler [Y/N] (Y)? Y RET

How many extra device slots do you want [D] (0)? 2 RET

Do you want to retain the system OBJ's [Y/N] (N)? N RET

How many local DL11's, including the console [D] (1)? 2 RET

How many remote DL11 lines [D] (0)? 0 RET

What is the CSR address for the
   first (console) DL11 [O] (177560)? 177560 RET

What is the vector address for the first (console) DL11 [O] (60)? 60 RET
What is the CSR address for the next (local) DL11 [O] (176500)? 176500 (RET)

What is the vector address for the next (local) DL11 [O] (300)? 300 (RET)

Do you want DZ11 or DZV11 multiplexor support [Y/N] (N)? N (RET)

Size of the output buffers in characters [D] (40)? 40 (RET)

Size of the input buffers in characters [D] (82)? 82 (RET)

Do you want asynchronous terminal status [Y/N] (Y)? Y (RET)

Do you want multi-terminal time-out support [Y/N] (Y)? Y (RET)

What is the PHYSICAL name and unit of the
   source input device [DDU] (RK1)? DX0 (RET)

What is the PHYSICAL name and unit of the
   binary output device [DDU] (RK0)? DX1 (RET)

What is the PHYSICAL name and unit of the
   map output device [DDU] (TT)? LP (RET)

END OF SYSGEN PROGRAM--Don't forget to copy DK:SYCND.MAC and
        DK:SYSTBL.MAC to the source disk before executing the
        command file(s). Remember that SWAP.SYS must
        be on all bootable volumes.
```

The system conditional file produced by the sample dialogue is as follows:

```
 .SBTTL  SYSTEM CONDITIONAL FILE
         ; SYCND.MAC--SYSTEM CONDITIONAL FILE
         ;
         ; COPYRIGHT (C)    1978
         ;
         ; DIGITAL EQUIPMENT CORPORATION,  MAYNARD,   MASS.

SYSG$N   = 1                ;INDICATE SYSGENED MONITOR
MTT$Y    = 1                ;MULTIPLE TERMINAL SUPPORT (FB,XM)
```

EXAMPLE SYSGEN'S

```
CLOCK    = 60.           ;POWER LINE FREQUENCY (SJ,FB,XM)
STAR$T   = 1             ;STARTUP COMMAND FILE (SJ,FB,XM)
PWF$L    = 1             ;POWER FAIL (SJ,FB,XM)
ESC$P    = 0             ;NO ESCAPE SEQUENCES (SJ,FB,XM)
$DXSYS   = 1             ;SYSTEM DEVICE
DX$CSR   = 177170        ;CSR OF FIRST RX11
DX$VEC   = 264           ;VECTOR OF FIRST RX11
DL11$L   = 2.            ;# OF LOCAL DL11 INTERFACES (MULTI-TERM SUPPORT)
DL11$M   = 0.            ;# OF REMOTE DL11-E LINES (MULTI-TERM SUPPORT)
DL11$N   = 2. + 0.       ;TOTAL # OF DL11 LINES (MULTI-TERM SUPPORT)
DLC$0    = 177560        ;CSR OF 1ST (CONSOLE) DL11 (MULTI-TERM SUPPORT)
DLV$0    = 60            ;VECTOR OF 1ST (CONSOLE) DL11 (MULTI-TERM SUPPORT)
DLC$1    = 176500        ;CSR OF 2ND (LOCAL) DL11 (MULTI-TERM SUPPORT)
DLV$1    = 300           ;VECTOR OF 2ND (LOCAL) DL11 (MULTI-TERM SUPPORT)
DZ11$N   = 0             ;NO DZ11 OR DZV11 SUPPORT (MULTI-TERM SUPPORT)
TTYOUT   = 40.           ;SIZE OF OUTPUT RING BUFFERS(MULTI-TERM SUPPORT)
TTYIN    = 82.           ;SIZE OF INPUT RING BUFFERS (MULTI-TERM SUPPORT)
MAT$S    = 1             ;ASYNCHRONOUS TERMINAL STATUS (MULTI-TERM SUPPORT)
MTI$M    = 1             ;TERMINAL TIME OUT (MULTI-TERM SUPPORT)
TIM$IT   = 1             ;GENERAL DEVICE TIME OUT (MULTI-TERM SUPPORT)
```

The MONBLD.COM and DEVBLD.COM files produced by the sample dialogue are as follows:

```
!       MONBLD.COM
!
ASS DX0 SRC
ASS DX1 BIN
ASS LP MAP
MACRO/OBJ:BIN:KMFB SRC:(FB+SYCND+EDTGBL+KMON+KMOVLY)
MACRO/OBJ:BIN:RMFB SRC:(FB+SYCND+EDTGBL+USR+RMONFB)
MACRO/OBJ:BIN:SYFB SRC:(FB+SYCND+SYSDEV+DX)
MACRO/OBJ:BIN:TBFB SRC:(FB+SYCND+SYSTBL)
MACRO/OBJ:BIN:BTFB SRC:(FB+SYCND+BSTRAP)
MACRO/OBJ:BIN:MEFB SRC:(FB+SYCND+MTTEMT)
MACRO/OBJ:BIN:MIFB SRC:(FB+SYCND+MTTINT)
LINK/EXE:BIN:DXMNFB.SYG/BOU:1000/PROMPT/MAP:MAP:DXMNFB BIN:BTFB
BIN:RMFB,KMFB,MEFB,MIFB,TBFB,SYFB//
OVLY0
DELETE/NOQ BIN:(BT,RM,ME,MI,TB,SY)FB.OBJ
DELETE/NOQ BIN:KMFB.OBJ
```

```
!       DEVBLD.COM
!
ASS DX0 SRC
ASS DX1 BIN
MACRO/OBJ:BIN:LP SRC:(SYCND+LP)
LINK/EXE:BIN:LP.SYG BIN:LP
DELETE/NOQ BIN:LP.OBJ
MACRO/OBJ:BIN:DX SRC:(SYCND+DX)
LINK/EXE:BIN:DX.SYG BIN:DX
DELETE/NOQ BIN:DX.OBJ
MACRO/OBJ:BIN:NL SRC:(SYCND+NL)
LINK/EXE:BIN:NL.SYG BIN:NL
DELETE/NOQ BIN:NL.OBJ
```

To assemble and store the system components, you need three diskettes. The first diskette will serve as your system volume during the assembly procedure. It must contain the following files (and only these files):

DXMNSJ.SYS	(This is the base-line SJ monitor)
SWAP.SYS	
TT.SYS	
MACRO.SAV	
SYCND.MAC	
PIP.SAV	
SYSMAC.SML	
xx.MAC	(xx represents the monitor you are building, SJ, FB, or XM)
EDTGBL.MAC	
KMON.MAC	
KMOVLY.MAC	

The second diskette will serve as work file space during the assembly procedure. It will also store the .OBJ files that are produced as a result of the assemblies, and will become the bootable system diskette when the SYSGEN procedure is completed. This diskette must have the following files on it:

SWAP.SYS	
DIR.SAV	
PIP.SAV	
DUP.SAV	
TT.SYS	(If you are creating an SJ monitor)
LP.SYS	(If you want link maps to print on a line printer)

The third diskette will store the .MAC source files for the assemblies until they are needed on the system diskette. It must contain the following files:

RMONxx.MAC	(xx is SJ, FB, or XM)
USR.MAC	
SYSDEV.MAC	
DX.MAC	
SYSTBL.MAC	
BSTRAP.MAC	
MTTINT.MAC	(Only if multi-terminal support is required)
MTTEMT.MAC	(Only if multi-terminal support is required)
dd.MAC	(dd represents the device name for any device support you requested in addition to the system device; LP, NL, etc.)
LINK.SAV	
DUP.SAV	

The files listed above are located either on your system disk as a result of the SYSGEN dialogue, or on the distribution diskettes.

Initialize the three diskettes:

Type: `INITIALIZE/BADBLOCKS/SEGMENTS:1 DX1:` RET

Response: `DX1:/Init are you sure?`

Type: `Y` RET

Now copy the files described above to the appropriate diskettes. When the files are all stored on the correct diskettes, use the SQUEEZE command on each volume to make sure it is not fragmented.

```
Type:      SQUEEZE DX1:(RET)

Response:  DX1:/Squeeze are you sure?

Type:      Y (RET)
```

Now place the new system diskette in drive 0 and boot the system.

```
Response:  RT-11SJ     V03B-xx
           ?KMON-F-Command file not found
```

Enter the date and time:

```
Type:      DATE dd-mmm-yy(RET)
           TIME hh:mm:ss (RET)
```

Assign the logical name WF to DX1:.

```
Type:      ASSIGN DX1 WF(RET)

Response:  .
```

Mount the second diskette, the one that is to store the .OBJ files, in drive 1.

Now you are ready to perform the first assembly. Refer to MONBLD.COM for the component to be assembled first. Type the following commands:

```
Type:      R MACRO (RET)

Response:  *

Type:      DX1:KMxx=xx,SYCND,EDTGBL,KMON,KMOVLY (RET)
```

where xx is SJ, FB, or XM.

```
Response:  ERRORS DETECTED:   0
           *

Type:      (CTRL/C)
```

The first assembly is quite lengthy and can take a considerable amount of time. The more memory there is on your system, the faster the assembly runs.

The demonstration listing is as follows:

```
.R MACRO
*DX1:KMFB=FB,SYCND,EDTGBL,KMON,KMOVLY
ERRORS DETECTED:  0
*^C
```

When the assembly completes, you can delete two source files from your system diskette.

```
Type:      DELETE DX0:(KMON,KMOVLY).MAC (RET)
```

Respond by typing Y followed by a carriage return to delete each file when the system queries you.

Now remove the .OBJ diskette and mount the third diskette, the one that contains the .MAC source files, in drive 1. Copy the next two source files onto your system disk.

```
Type:      COPY DX1:(USR,RMONxx).MAC DX0:(RET)
```

where xx is SJ, FB, or XM.

Remove the .MAC diskette and mount the .OBJ diskette in drive 1.

Invoke MACRO to assemble the next system component.

Type: `R MACRO` (RET)

Response: `*`

Type: `DX1:RMxx=xx,SYCND,EDTGBL,USR,RMONxx` (RET)

where xx is SJ, FB, or XM.

Response:
```
ERRORS DETECTED:  0
*
```

Type: (CTRL/C)

The demonstration listing is as follows:

```
.R MACRO
*DX1:RMFB=FB,SYCND,EDTGBL,USR,RMONFB
ERRORS DETECTED:  0
*
```

Next, delete the two source files from your system diskette:

Type: `DELETE DX0:(USR,RMONxx).MAC` (RET)

where xx is SJ, FB, or XM.

Respond by typing Y followed by a carriage return to delete each file when the system queries you.

Remove the .OBJ diskette and mount the .MAC diskette in drive 1. Copy the next three source files onto your system diskette.

Type: `COPY DX1:(SYSDEV,DX,SYSTBL).MAC DX0:` (RET)

Remove the .MAC diskette and mount the .OBJ diskette in drive 1.

Invoke MACRO to assemble the next two system components.

Type: `R MACRO` (RET)

Response: `*`

Type: `DX1:SYxx=xx,SYCND,SYSDEV,DX` (RET)

where xx is SJ, FB, or XM.

Response:
```
ERRORS DETECTED:  0
*
```

Type: `DX1:TBxx=xx,SYCND,SYSTBL` (RET)

where xx is SJ, FB, or XM.

Response:
```
ERRORS DETECTED:  0
*
```

Type: (CTRL/C)

The demonstration listing is as follows:

```
        .R MACRO
        *DX1:SYFB=FB,SYCND,SYSDEV,DX
        ERRORS DETECTED:  0
        *DX1:TBFB=FB,SYCND,SYSTBL
        ERRORS DETECTED:  0
        *^C
```

Now delete two of those three source files from your system diskette, leaving the DX.MAC file there.

```
Type:     DELETE DX0:(SYSDEV,SYSTBL).MAC RET
```

Respond by typing Y followed by a carriage return to delete each file when the system queries you.

Remove the .OBJ diskette and mount the .MAC diskette in drive 1 and copy the next group of source files onto your system diskette.

```
Type:     COPY DX1:(BSTRAP,MTTEMT,MTTINT,zz).MAC DX0: RET
```

where zz represents the name of any extra devices you requested in the SYSGEN dialogue; LP, NL, etc. MTTEMT and MTTINT are required only for multi-terminal systems.

Copy LINK, DUP, and any remaining handler files to the system diskette.

```
Type:     COPY DX1:(LINK,DUP).SAV,DX1:(zz).MAC DX0: RET
```

where zz represents the name of any other extra devices.

Now remove the .MAC diskette from drive 1 and mount the .OBJ diskette. Invoke MACRO to perform the next group of assemblies.

```
Type:     R MACRO RET

Response: *

Type:     DX1:  BTxx=xx,SYCND,BSTRAP RET
```

where xx is SJ, FB, or XM.

```
Response: ERRORS DETECTED:  0
          *
```

The next assembly is required only for multi-terminal systems.

```
Type:     DX1:MExx=xx,SYCND,MTTEMT RET
```

where xx is SJ, FB, or XM.

```
Response: ERRORS DETECTED:  0
          *
```

The following assembly is required only for multi-terminal systems.

```
Type:     DX1:MIxx=xx,SYCND,MTTINT RET
```

where xx is SJ, FB, or XM.

```
Response: ERRORS DETECTED:  0
          *
```

The demonstration listing is as follows:

```
.R MACRO
*DX1:BTFB=FB,SYCND,BSTRAP
ERRORS DETECTED:  0
*DX1:MEFB=FB,SYCND,MTTEMT
ERRORS DETECTED:  0
*DX1:MIFB=FB,SYCND,MTTINT
ERRORS DETECTED:  0
*
```

The following command lines to MACRO were derived from the file DEVBLD.COM. Study the listing of DEVBLD.COM before proceeding.

The first command assembles the system device handler.

Type: `DX1:DX=SYCND,DX` RET

Response:
```
ERRORS DETECTED:  0
*
```

Next, assemble any other handlers for those devices you requested in the SYSGEN dialogue.

Type: `DX1:xx=SYCND,xx` RET

where xx represents the name of a device for which you requested support.

The demonstration listing assembles DX, NL, and LP:

```
.R MACRO
*DX1:DX=SYCND,DX
ERRORS DETECTED:  0
*DX1:NL=SYCND,NL
ERRORS DETECTED:  0
*DX1:LP=SYCND,LP
ERRORS DETECTED:  0
*^C
```

Now you are ready to link the .OBJ files. If you want the link maps to print on the line printer, copy the LP handler from the .OBJ diskette onto the system diskette.

Type: `COPY DX1:LP.SYS/SYSTEM DX0:` RET

The system prints the ?PIP-W-Reboot message. You should now reboot the system diskette, and enter the date and time.

You can now link the .OBJ files to create the RX01 monitor and its associated device handlers. Be careful to save the link map that results from linking the monitor. If you ever need to submit an SPR to DIGITAL for a monitor created by the system generation process, you must include the monitor link map with the SPR.

Type: `R LINK` RET

Response: *

Type: `DX1:DXMNxx.SYS,map=DX1:BTxx,RMxx,KMxx,MExx,MIxx/Y:1000/C` RET

where map represents the destination device or file for the link map. Use TT: for the terminal, LP: for the line printer, and DX1: filnam.typ to store the map in a file.

Response: *

Type: `DX1:TBxx,SYxx` RET

Response: Boundary section?

Type: OVLYO (RET)

Response: *

Type: DX1:DX.SYS=DX1:DX (RET)

Response: *

Type: DX1:xx.SYS=DX1:xx (RET)

where xx represents the name of any other handlers you requested in the SYSGEN dialogue.

Response: *

When you have linked all the extra handlers, exit from LINK.

Type: (CTRL/C)

The demonstration listing is as follows:

```
.R LINK
*DX1:DXMNFB.SYS,LP:=DX1:BTFB,RMFB,KMFB,MEFB,MIFB/Y:1000/C
*DX1:TBFB,SYFB
Boundary section? OVLYO
*DX1:DX.SYS=DX1:DX
*DX1:LP.SYS=DX1:LP
*DX1:NL.SYS=DX1:NL
*^C
```

Compress the files on the .OBJ diskette.

```
.SQUEEZE DX1:(RET)
DX1:/Squeeze are you sure?Y'(RET)
```

Copy the bootstrap onto the .OBJ diskette.

Type: COPY/BOOT DX1:DXMNxx DX1:(RET)

where xx is SJ, FB, or XM.

Use the BOOT command to boot the .OBJ diskette, which is now the new system diskette.

```
.BOOT DX1:
RT-11FB (S)V03B-02
?KMON-F-Command file not found
```

The (S) in the boot message means that the monitor was created through the system generation process.

Move the .OBJ diskette to drive 0 and perform a hardware bootstrap.

```
RT-11FB (S)V03B-02
?KMON-F-Command file not found
```

Enter the date and time, and copy the .OBJ files from the new system diskette to another diskette for safe keeping. You can then delete them from the system diskette. Keeping the .OBJ files will help you later if it becomes necessary to add a patch to the system. Patches are distributed in source form. If, for example, only one component of the system, such as KMON, needs a patch, you can assemble just that one component. Then you can link the new .OBJ file with the old .OBJ files from your original SYSGEN. This saves your having to do a complete SYSGEN in order to install a patch.

Your system diskette should now be ready to store more files and to function appropriately in your application environment.

APPENDIX B

SYSGEN SCRIPT LANGUAGE DESCRIPTION

B.1 INTRODUCTION

SYSGEN is a FORTRAN program that uses a script file to drive the terminal dialogue and output information to command and conditional files. This appendix describes the script language, allowing you to understand the system script file and use SYSGEN for application generations of your own. Note, however, that DIGITAL only supports SYSGEN for use with the script files distributed with the system; this description is for your convenience only. For the discussion that follows, the script file distributed with the system serves as a good example; you should have a listing at your disposal while reading this appendix.

The SYSGEN script file, which must be named SYSGEN.CND and reside on device DK:, is an ASCII input file consisting of SYSGEN command lines interspersed with the actual text lines that are to be output to the output files.

Directed by commands in the script file, the SYSGEN program interacts with a user at the terminal. It manipulates internal variables called "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. Figure B-1 illustrates the flow of information through the SYSGEN program.

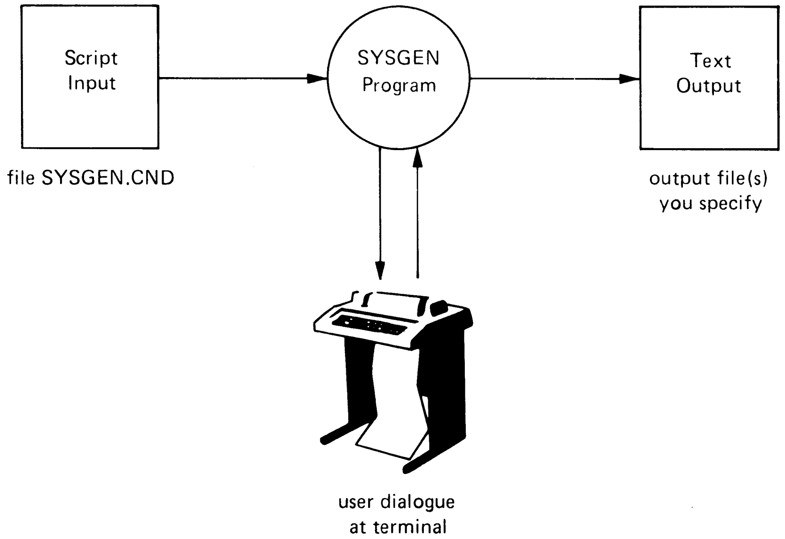


Figure B-1 SYSGEN Program Information Flow

B.2 COMMAND SYNTAX

Each line in a SYSGEN script file must be less than or equal to 80 characters and falls into one of the three following categories:

1. If the line begins with a semicolon (;) character, SYSGEN ignores it. Use lines beginning with semicolon (;) to comment the script file.

2. If the command line begins with

   TAB #

   SYSGEN interprets the remaining portion of the line as a command line.

3. In all other cases, SYSGEN outputs the entire text line to the current open output file after it has made any substitutions in effect (see SUBS command in Section B.3.9). If you need to output a line that begins with a semicolon (;) to the output file, precede the semicolon (;) with a space or tab to differentiate it from a script comment.

Example:

```
;THIS IS A SCRIPT EXAMPLE
TAB #COMMAND1 TAB ARG1
TAB ;CONDITIONAL FILE COMMENT
TAB #COMMAND2 TAB ARG1
COND1=1            ;CONDITIONAL
COND2=2            ;CONDITIONAL
```

The example shown above results in the current output file containing the following lines after COMMAND1 and COMMAND2 execute:

```
TAB ;CONDITIONAL FILE COMMENT
COND1=1            ;CONDITIONAL
COND2=2            ;CONDITIONAL
```

Each command must begin with a TAB followed by a # character. You must terminate the command name by a second TAB or RET LF (if no argument is present). If the first character is not a TAB or the second is not a #, SYSGEN treats the line as a text line. If the first character is a TAB and the second a # character and the command does not match a legal command name, SYSGEN reports an error and treats the line as a text line.

B.2.1 Script Variables

The script commands manipulate script variables. Script variables have names of from one to six alphanumeric characters. Script variables are initially undefined; you establish their values either by a SET command (see Section B.3) or by using the variable of interest in an #ASK command (see Section B.3.6). Each script variable has two values associated with it: a "character value" equal to the actual characters used in a #SET or #ASK command, and a "numeric value" equal to the numeric value of the characters to which the variable was associated during an #ASK command. #IF, #IFN, and #SUB commands use the character value. #IFGT and #DECR commands use the numeric value. #ASK 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 (see Section B.3.9) does not take effect until the next SUBS statement.

B.3 COMMANDS

Following are the legal commands in the SYSGEN script language.

B.3.1 FILE

```
TAB #FILE TAB filnam.typ
```

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.

Example:

```
#FILE SYCND.MAC
#FILE SY1.COM
```

B.3.2 NAME

```
TAB #NAME TAB segnam
```

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. This is helpful in logically dividing the script file into discrete sections. The segment name consists of one to six alphanumeric characters.

Example:

```
#NAME SYCND
#NAME PART1
```

B.3.3 IF,IFN,IFGT

```
TAB #IF TAB variable
TAB #IFN TAB variable
TAB #IFGT TAB variable
```

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"; #IFN tests for "NO". #IFGT 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. (See #DECR, Section B.3.13.)

You can nest #IF, #IFN, and #IFGT commands up to 10 levels, but you must pair each #IF, #IFN, or #IFGT command with a corresponding #ENDC command.

Example:

```
             #SET A=Y
             #SET B=N
             #IF A
     A IS YES
             #ENDC
             #IFN A
     A IS NO
             #ENDC
             #IF B
     B IS YES
             #ENDC
             #IFN B
     B IS NO
             #ENDC
```

The preceding script writes the following lines in the output file.

```
     A IS YES
     B IS NO
```

B.3.4 IFT, IFF, IFTF

```
     TAB #IFT
     TAB #IFF
     TAB #IFTF
```

You use the #IFT, #IFF and #IFTF commands within an existing #IF, #IFN, or #IFGT conditional block to:

1. Include an alternate body of text and command lines when the condition of the block tests false

2. Include a non-contiguous 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.

Thus, a revision of the previous example is as follows:

```
        #SET A=Y
        #SET B=N
        #IF A
A IS YES
        #IFF
A IS NO
        #ENDC
        #IFN B
        #IFF
B IS YES
        #IFT
B IS NO
        #ENDC
```

B.3.5 ENDC

TAB #ENDC TAB [variable name]

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.

You can optionally follow the #ENDC command with a script variable name. SYSGEN does not use the variable, but it helps to mark which block is being terminated in nesting situations.

Example:

```
#IF     COND1
   .
   .
   .
#ENDC   COND1
#IFN    COND2
   .
   .
   .
#IF     COND3
   .
   .
   .
#IFF
   .
   .
   .
#IFT
   .
   .
   .
#ENDC   COND3
#ENDC   COND2
```

B.3.6 ASK

TAB #ASK TAB [-]question?variable/ $\left\{\begin{matrix}A\\D\\O\\Y\end{matrix}\right\}$ =default[,lowlimit][,highlimit]]

You use the #ASK command to print dialogue on the terminal and to obtain an acceptable user response. The #ASK command then sets the specified script variable's character value and numeric value according to the user's response.

The #ASK command prints the text (question) indicated between the tab and the ? character on the terminal and waits for the user to respond.

The question, question mark (?), variable name, slash character (/), type letter (A, D, O, or Y), equal sign (=), and default value are all required. The low and high limit values, which are optional and apply only to numeric responses, establish a valid range for acceptable responses. The type letter codes (A, D, O, and Y) establish validity checking for the following data types:

/A The answer must be an alphanumeric device mnemonic of the form XX RET or XXd RET , where X is a letter and d 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 YES RET or NO RET .

The optional hyphen (-) is a continuation character. When it is the first character following TAB #ASK TAB , SYSGEN assumes the question portion of the command to be the second line of a two line question. Use #PRINT to display the first line of the question.

If the response to the question is RET , SYSGEN uses the default response. (Note that SYSGEN does not check the default answer for validity.) SYSGEN equates the variable's character value to the characters in the user response and sets the variable's numeric value equal to the value of the numeric response, if any. It then proceeds to the next command.

SYSGEN checks all terminal responses for validity according to the type letter code (A, D, O, or Y) specified in the #ASK command. In addition, it checks numeric responses against the low and high limits, if present. SYSGEN prints an error message if the response is not valid, and it then repeats the question.

Examples:

```
    #ASK HOW OLD ARE YOU?  AGE/D = 21, 18, 65

    #PRINT THIS IS THE FIRST LINE
    #ASK -  OF A TWO LINE QUESTION?  ANS/Y=N
```

The PRINT/ASK combination prints in the dialogue as follows:

```
    THIS IS THE FIRST LINE
      OF A TWO LINE QUESTION?
```

B.3.7 SET

TAB #SET TAB variable name=answer

The #SET command equates the character value of the specified variable to the characters in the specified answer.

Examples:

```
#SET A=1
#SET COND1=N
#SET COND2=Y
```

B.3.8 PRINT

TAB #PRINT TAB text

The #PRINT command prints the specified text on the terminal preceded by a LF and followed by a RET .

NOTE

> To #PRINT a blank line, you must include the succeeding TAB in the command.

Examples:

```
#PRINT MESSAGE1
#PRINT TAB
#PRINT MESSAGE2
```

This example prints the following lines on the terminal:

```
MESSAGE1

MESSAGE2
```

B.3.9 SUBS

TAB #SUBS TAB <delimiter>pattern<delimiter>variable name

The #SUBS command substitutes the character value of the specified script variable for each occurrence of the pattern in all script lines occurring between the #SUBS command and the next #ENDS commands.

The delimiter can be any character that does not appear in the pattern.

You can use the #SUBS command to modify the command lines as well as text lines in the script file.

Examples:

```
        SET COND1=3
        #SUBS "$XXX$"COND1
        #PRINT THE VALUE OF COND1 IS $XXX$
COND1=$XXX$
        #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
```

Note that 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.

B.3.10 **ENDS**

(TAB) #ENDS

The #ENDS command terminates the substitution activities of all active #SUBS commands. Only one is needed, regardless of the number of SUBS statements that are active.

B.3.11 **EXIT**

(TAB) #EXIT

The #EXIT command terminates the SYSGEN run without closing the active output file.

B.3.12 **CALL**

(TAB) #CALL (TAB) filnam.typ

The #CALL command causes SYSGEN to temporarily use the file specified by filnam.typ as the 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) where it left off.

You cannot nest #CALL commands; only one CALL command can be active at a time.

There is no default file type; you must explicitly enter the file type.

Example:

```
    #CALL  SYSTBL.CND
```

B.3.13 **DECR**

(TAB) #DECR (TAB) variable name

The #DECR command decrements the numeric value associated with the specified variable. SYSGEN sets the numeric value of the variable equal to the numeric value of the user response to a dialogue question when you use the variable in a SET or ASK command. #DECR alters only the numeric value; it does not alter the answer.

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, this combination is used in SYSGEN.CND to ask for n DL11 vectors, where n is the number of DL11's specified as a response to the dialogue question.

B.4 SCRIPT FILE ERROR MESSAGES

SYSGEN detects errors in the script file where possible. Following are the errors that SYSGEN detects.

SYSGEN script file errors print in the form:

```
     error message in segment name
     script line
```

where 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.

Message	Meaning
?SYSGEN-W-Bad #ASK in	An #ASK command is missing the ? delimiter.
?SYSGEN-W-Bad #SET in	A #SET command is missing the = delimeter.
?SYSGEN-W-Bad substitute pattern in	The delimiters do not match in a #SUBS command.
?SYSGEN-W-Missing variable in	You issued an #IF, #IFN, or #IFGT command without a script variable.
?SYSGEN-W-Missing file name in	You issued a #FILE command without a file name.
?SYSGEN-W-Nested #CALL in	You issued a #CALL command from a script file other than the original.
?SYSGEN-W-Too few #ENDC's in	The number of #IF, #IFN, and #IFGT statements exceeds the number of #ENDC commands.
?SYSGEN-W-Too many variables in	You defined more than 100 script variables.
?SYSGEN-W-Too many #ENDC's in	The number of #ENDC commands exceeds the number of #IF, #IFN, and #IFGT commands.
?SYSGEN-W-Too many nested #IFs or #SUBs in	The number of nested #IF, #IFN and #IFGT commands exceeds 10; the number of active #SUB commands exceeds five.
?SYSGEN-W-Undefined command in	The printed script line began with TAB #, but it is an undefined command.
?SYSGEN-W-Undefined variable in	The variable name you used in an #IF, #IFN, #IFGT, or #SUBS command is undefined.

APPENDIX C

INSTRUCTIONS FOR LOADING SOFTWARE BOOTSTRAPS

C.1 RK11(RK05) DECPACK BOOTSTRAP LOADER

1. Deposit the basic RK11 disk bootstrap loader into memory as follows:

   a. Set the ENABLE/HALT switch to HALT, then set the first address, 001000, in the Switch Register. (Set switch 9 to the up (1) position and all others to the down (0) position.)

   b. Press the LOAD ADDR switch.

   c. Set the proper contents for Table C-1 in the Switch Register and lift the DEP Switch.

   d. Repeat Step c until all the instructions have been deposited.

2. Verify that the bootstrap program has been deposited properly as follows:

   a. Set the starting address in the Switch Register as in Step 1a above.

   b. Press the LOAD ADDR switch.

   c. Display the contents of that address in the Data Register by pressing the EXAM switch.

   d. Compare the number in the Data Register with the value in Table C-1.

   e. If they are the same, repeat Step 2c until all words have been examined.

   f. If not the same, repeat Step 1.

3. Set the starting address, 001000, in the Switch Register as in Step 1a above, then press the LOAD ADDR switch.

4. Set the ENABLE/HALT switch to ENABLE, then press the START switch.

Table C-1
RK11 Bootstrap Loader

Location	Contents	Like This
001000	012700	Set switches 12, 10, 8, 7, and 6 to the up (1) position. Set all others to the down (0) position.
001002	177406	Set switches 15 through 8, 2, and 1 to the up (1) position. Set all others to the down (0) position.
001004	012710	Set switches 12, 10, 8, through 6, and 3 to the up (1) position. Set all others to the down (0) position.
001006	177400	Set switches 15 through 8 to the up (1) position. Set all others to the down (0) position.
001010	012740	Set switches 12, 10, and 8 through 5 to the up (1) position. Set all others to the down (0) position.
001012	000005	Set switches 2 and 0 to the up (1) position. Set all others to the down (0) position.
001014	105710	Set switches 15, 11, 9 through 6, and 3 to the up (1) position. Set all others to the down (0) position.
001016	100376	Set switches 15 and 7 through 1 to the up (1) position. Set all others to the down (0) position.
001020	005007	Set switches 11, 9, and 2 through 0 to the up (1) position. Set all others to the down (0) position.

C.2 TC11 DECTAPE BOOTSTRAP LOADER

1. Deposit the basic DECtape bootstrap loader into memory as follows:

   a. Set the ENABLE/HALT switch to HALT, then set the first address, 001000, in the Switch Register. (Set switch 9 to the up (1) position and all others to the down (0) position.

   b. Press the LOAD ADDR switch.

   c. Set the proper contents from Table C-2 in the Switch Register and lift the DEP switch.

   d. Repeat Step c until all the instructions have been deposited.

Table C-2
TC11 Bootstrap Loader

Location	Contents	Like This
001000	012700	Set switches 12, 10, 8, 7, and 6 to the up (1) position. Set all others to the down (0) position.
001002	177344	Set switches 15 through 9, 7, 6, 5, and 2 to the up (1) position. Set all others to the down (0) position.
001004	012710	Set switches 12, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.
001006	177400	Set switches 15 through 8 to the up (1) position. Set all others to the down (0) position.
001010	012740	Set switches 12, 10, 8, 7, 6, and 5 to the up (1) position. Set all others to the down (0) position.
001012	004002	Set switches 11 and 1 to the up (1) position. Set all others to the down (0) position.
001014	005710	Set switches 11, 9, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.
001016	100376	Set switches 15, and 7 through 1 to the up (1) position. Set all others to the down (0) position.
001020	012710	Set switches 12, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.
001022	000003	Set switches 1 and 0 to the up (1) position. Set all others to the down (0) position.
001024	105710	Set switches 15, 11, 9, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.
001026	100376	Set switches 15, and 7 through 1 to the up (1) position. Set all others to the down (0) position.
001030	012710	Set switches 12, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.
001032	000005	Set switches 2 and 0 to the up (1) position. Set all others to the down (0) position.

(continued on next page)

Table C-2 (Cont.)
TC11 Bootstrap Loader

Location	Contents	Like This
001034	105710	Set switches 15, 11, 9, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.
001036	100376	Set switches 15 and 7 through 1 to the up (1) position. Set all others to the down (0) position.
001040	005007	Set switches 11, 9, 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.

2. Verify that the bootstrap program has been deposited properly as follows:

   a. Set the starting address in the Switch Register as in Step 1a above.

   b. Press the LOAD ADDR switch.

   c. Display the contents of that address in the Data Register by pressing the EXAM switch.

   d. Compare the number in the Data Register with the value in Table C-2.

   e. If they are the same, repeat Step 2c until all words have been examined.

   f. If not the same, repeat Step 1.

3. Set the starting address 001000 in the Switch Register as in Step 1a above, then press the LOAD ADDR switch.

4. Set the ENABLE/HALT switch to ENABLE, then press the START switch.

C.3 RX11/RX01 DISKETTE BOOTSTRAP LOADER

1. If the computer is a PDP-11V/03, perform the following; if the computer is a PDP-11/03 or LSI-11, see the LSI-11, PDP-11/03 User's Manual (EK-LSI11-TM-002); otherwise, go to Step 2.

   a. Put all three switches in the up position.

   b. Move the DC ON/OFF switch down and up.

   c. Response: $
      Type: DX RET

The bootstrapping procedure is complete.

2. Deposit the basic diskette bootstrap loader into memory as follows:

   a. Set the ENABLE/HALT switch to HALT, then set the first address, 001000, in the Switch Register. (Set switch 9 to the up (1) position and all others to the down (0) position.)

   b. Press the LOAD ADDR switch.

   c. Set the proper contents from Table C-3 in the Switch Register and lift the DEP switch.

   d. Repeat Step 2c until all the instructions have been deposited.

Table C-3
RX11 Bootstrap Loader

Location	Contents	Like This
001000	012702	Set switches 12, 10, 8, 7, 6, and 1 to the up (1) position. Set all others to the down (0) position.
001002	1002n7 (n=4 for unit 0 and n=6 for unit 1)	Set switches 15, 7, 2, 1, and 0 to the up (1) position and all others to the down (0) position. For unit 0, set switch 5 to the up (1) position; for unit 1, set switches 5 and 4 to the up (1) position.
001004	012701	Set switches 12, 10, 8, 7, 6, and 0 to the up (1) position. Set all others to the down (0) position.
001006	177170	Set switches 15 through 9, and 6 through 3 to the up (1) position. Set all others to the down (0) position.
001010	130211	Set switches 15, 13, 12, 7, 3, and 0 to the up (1) position. Set all others to the down (0) position.
001012	001776	Set switches 9 through 1 to the up (1) position. Set all others to the down (0) position.
001014	112703	Set switches 15, 12, 10, 8, 7, 6, 1, and 0 to the up (1) position. Set all others to the down (0) position.
001016	000007	Set switches 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.
001020	010100	Set switches 12 and 6 to the up (1) position. Set all others to the down (0) position.

(continued on next page)

Table C-3 (Cont.)
RX11 Bootstrap Loader

Location	Contents	Like This
001022	010220	Set switches 12, 7, and 4 to the up (1) position. Set all others to the down (0) position.
001024	000402	Set switches 8 and 1 to the up (1) position. Set all others to the down (0) position.
001026	012710	Set switches 12, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.
001030	000001	Set switch 0 to the up (1) position. Set all others to the down (0) position.
001032	006203	Set switches 11, 10, 7, 1, and 0 to the up (1) position. Set all others to the down (0) position.
001034	103402	Set switches 15, 10, 9, 8, and 1 to the up (1) position. Set all others to the down (0) position.
001036	112711	Set switches 15, 12, 10, 8, 7, 6, 3, and 0 to the up (1) position. Set all others to the down (0) position.
001040	111023	Set switches 15, 12, 9, 4, 1, and 0 to the up (1) position. Set all others to the down (0) position.
001042	030211	Set switches 13, 12, 7, 3, and 0 to the up (1) position. Set all others to the down (0) position.
001044	001776	Set switches 9 through 1 to the up (1) position. Set all others to the down (0) position.
001046	100756	Set switches 15, 8, 7, 6, 5, and 3, 2, and 1 to the up (1) position. Set all others to the down (0) position.
001050	103766	Set switches 15, 10 through 4, 2, and 1 to the up (1) position. Set all others to the down (0) position.
001052	105711	Set switches 15, 11, 9 through 6, 3, and 0 to the up position. Set all others to the down (0) position.
001054	100771	Set switches 15, 8 through 3, and 0 to the up (1) position. Set all others to the down (0) position.

(continued on next page)

Table C-3 (Cont.)
RX11 Bootstrap Loader

Location	Contents	Like This
001056	005000	Set switches 11 and 9 to the up (1) position. Set all others to the down (0) position.
001060	022710	Set switches 13, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.
001062	000240	Set switches 7 and 5 to the up (1) position. Set all others to the down (0) position.
001064	001347	Set switches 9, 7, 6, 5, 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.
001066	122702	Set switches 15, 13, 10, 8, 7, 6, and 1 to the up (1) position. Set all others to the down (0) position.
001070	000247	Set switches 7, 5, 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.
001072	005500	Set switches 11, 9, 8, and 6 to the up (1) position. Set all others to the down (0) position.
001074	005007	Set switches 11, 9, 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.

3. Verify that the bootstrap program has been deposited properly as follows:

   a. Set the starting address in the Switch Register as in Step 2a above.

   b. Press the LOAD ADDR switch.

   c. Display the contents of that address in the Data Register by pressing the EXAM switch.

   d. Compare the number in the Data Register with the value in Table C-3.

   e. If they are the same, repeat Step 3c until all words have been examined.

   f. If not the same, repeat Step 2.

3. Set the starting address 001000 in the Switch Register as in Step 2a above, then press the LOAD ADDR switch.

4. Set the ENABLE/HALT switch to ENABLE, then press the START switch.

C.4 MAGTAPE BOOTSTRAP LOADERS

1. Deposit the basic magtape bootstrap into memory as follows:

   a. Set the ENABLE/HALT switch to HALT, then set the first address, 010000, in the Switch Register. (Set switch 12 to the up (1) position and all others to the down (0) position.)

   b. Press the LOAD ADDR switch.

   c. Set the proper contents for TJU16 magtape (Table C-4) or for TM11 magtape (Table C-5) in the Switch Register and lift the DEP switch.

   d. Repeat Step c until all the instructions have been deposited.

Table C-4
TJU16 Bootstrap Loader

Location	Contents	Like This
010000	012700	Set switches 12, 10, 8, 7, and 6 to the up (1) position. Set all others to the down (0) position.
010002	172440	Set switches 15 through 12, 10, 8, and 5 to the up (1) position. Set all others to the down (0) position.
010004	012710	Set switches 12, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.
010006	000021	Set switches 4 and 0 to the up (1) position. Set all others to the down (0) position.
010010	012760	Set switches 12, 10, and 8 through 4 to the up (1) position. Set all others to the down (0) position.
010012	001300	Set switches 9, 7, and 6 to the up (1) position. Set all others to the down (0) position.
010014	000032	Set switches 4, 3, and 1 to the up (1) position. Set all others to the down (0) position.
010016	012760	Set switches 12, 10, and 8 through 4 to the up (1) position. Set all others to the down (0) position.
010020	177777	Set all switches to the up (1) position.

(continued on next page)

Table C-4 (Cont.)
TJU16 Bootstrap Loader

Location	Contents	Like This
010022	000006	Set switches 2 and 1 to the up (1) position. Set all others to the down (0) position.
010024	012720	Set switches 12, 10, 8, 7, 6, and 4 to the up (1) position. Set all others to the down (0) position.
010026	000031	Set switches 4, 3, and 0 to the up (1) position. Set all others to the down (0) position.
010030	105760	Set switches 15, 11, and 9 through 4 to the up (1) position. Set all others to the down (0) position.
010032	000010	Set switch 3 to the up (1) position. Set all others to the down (0) position.
010034	100375	Set switches 15, 7 through 2, and 0 to the up (1) position. Set all others to the down (0) position.
010036	012710	Set switches 12, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.
010040	177000	Set switches 15 through 9 to the up (1) position. Set all others to the down (0) position.
010042	012740	Set switches 12, 10, and 8 through 5 to the up (1) position. Set all others to the down (0) position.
010044	000071	Set switches 5, 4, 3, and 0 to the up (1) position. Set all others to the down (0) position.
010046	032710	Set switches 13, 12, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.
010050	100200	Set switches 15 and 7 to the up (1) position. Set all others to the down (0) position.
010052	001775	Set switches 9 through 2 and 0 to the up (1) position. Set all others to the down (0) position.
010054	100007	Set switches 15, 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.

(continued on next page)

Table C-4 (Cont.)
TJU16 Bootstrap Loader

Locn	Contents	Like This
010056	022760	Set switches 13, 10, and 8 through 4 to the up (1) position. Set all others to the down (0) position.
010060	001000	Set switch 9 to the up (1) position. Set all others to the down (0) position.
010062	000014	Set switches 3 and 2 to the up (1) position. Set all others to the down (0) position.
010064	001403	Set switches 9, 8, 1, and 0 to the up (1) position. Set all others to the down (0) position.
010066	000005	Set switches 2 and 0 to the up (1) position. Set all others to the down (0) position.
010070	000167	Set switches 6, 5, 4, 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.
010072	177704	Set switches 15 through 6 and 2 to the up (1) position. Set all others to the down (0) position.
010074	005007	Set switches 11, 9, 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.

Table C-5
TM11 Bootstrap Loader

Location	Contents	Like This
010000	012700	Set switches 12, 10, 8, 7, and 6 to the up (1) position. Set all others to the down (0) position.
010002	172524	Set switches 15 through 12, 10, 8, 6, 4, and 2 to the up (1) position. Set all others to the down (0) position.
010004	005310	Set switches 11, 9, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.
010006	012740	Set switches 12, 10, and 8 through 5 to the up (1) position. Set all others to the down (0) position.

(continued on next page)

Table C-5 (Cont.)
TM11 Bootstrap Loader

Location	Contents	Like This
010010	060011	Set switches 14, 13, 3, and 0 to the up (1) position. Set all others to the down (0) position.
010012	105710	Set switches 15, 11, 9, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.
010014	100376	Set switches 15 and 7 through 1 to the up (1) position. Set all others to the down (0) position.
010016	005710	Set switches 11, 9, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.
010020	100767	Set switches 15, 8 through 4, 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.
010022	012710	Set switches 12, 10, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.
010024	060003	Set switches 14, 13, 1, and 0 to the up (1) position. Set all others to the down (0) position.
010026	105710	Set switches 15, 11, 9, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.
010030	100376	Set switches 15 and 7 through 1 to the up (1) position. Set all others to the down (0) position.
010032	005710	Set switches 11, 9, 8, 7, 6, and 3 to the up (1) position. Set all others to the down (0) position.
010034	100777	Set switches 15 and 8 through 0 to the up (1) position. Set all others to the down (0) position.
010036	005007	Set switches 11, 9, 2, 1, and 0 to the up (1) position. Set all others to the down (0) position.

2. Verify that the bootstrap program has been deposited properly as follows:

   a. Set the starting address, 010000, in the Switch Register as in Step 1a above.

   b. Press the LOAD ADDR switch.

   c. Display the contents of that address in the Data Register by pressing the EXAM switch.

d. Compare the number in the Data Register with the value in the appropriate table (C-4 or C-5).

e. If they are the same, repeat Step 2c until all words have been examined.

f. If not the same, repeat Step 1.

3. If TM11 magtape is being used, ensure that the magtape is positioned at the load point; if it is not, manually rewind the magtape.

4. Set the starting address, 010000, in the Switch Register as in Step 1a above, then press the LOAD ADDR switch.

5. Set the ENABLE/HALT switch to ENABLE, then press the START switch.

C.5 RK06 DECPACK BOOTSTRAP LOADER

1. Deposit the RK06 disk bootstrap loader into memory as follows:

   a. Set the ENABLE/HALT switch to HALT, then set the first address, 001000, in the Switch Register. (Set switch 9 to the up (1) position and all others to the down (0) position.)

   b. Press the LOAD ADDR switch.

   c. Set the proper contents for Table C-6 in the Switch Register and lift the DEP switch.

   d. Repeat Step c until all the instructions have been deposited.

2. Verify that the bootstrap program has been deposited properly as follows:

   a. Set the starting address in the Switch Register as in Step 1a above.

   b. Press the LOAD ADDR switch.

   c. Display the contents of that address in the Data Register by pressing the EXAM switch.

   d. Compare the number in the Data Register with the value in Table C-6.

   e. If they are the same, repeat Step 2c until all words have been examined.

   f. If not the same, repeat Step 1.

3. Set the starting address, 001000, in the Switch Register as in Step 1a above, then press the LOAD ADDR switch.

4. Set the ENABLE/HALT switch to ENABLE, then press the START switch.

Table C-6
RK06 Bootstrap Loader

Location	Contents	Like This
001000	012701	Set switches 12, 10, 8 through 6, and 0 to the up (1) position. Set all others to the down (0) position.
001002	177440	Set switches 15 through 8 and 5 to the up (1) position. Set all others to the down (0) position.
001004	012711	Set switches 12, 10, 8 through 6, 3, and 0 to the up (1) position. Set all others to the down (0) position. 001006 000003 Set switches 1 and 0 to the up (1) position. Set all others to the down (0) position.
001010	032711	Set switches 13, 12, 10, 8 through 6, 3, and 0 to the up (1) position. Set all others to the down (0) position.
001012	100200	Set switches 15 and 7 to the up (1) position. Set all others to the down (0) position.
001014	001775	Set switches 9 through 2, and 0 to the up (1) position. Set all others to the down (0) position.
001016	012761	Set switches 12, 10, 8 through 4, and 0 to the up (1) position. Set all others to the down (0) position.
001020	177400	Set switches 15 through 8 to the up (1) position. Set all others to the down (0) position.
001022	000002	Set switch 1 to the up (1) position. Set all others to the down (0) position.
001024	012711	Set switches 12, 10, 8 through 6, 3, and 0 to the up (1) position. Set all others to the down (0) position.
001026	000021	Set switches 4 and 0 to the up (1) position. Set all others to the down (0) position.
001030	032711	Set switches 13, 12, 10, 8 through 6, 3, and 0 to the up (1) position. Set all others to the down (0) position.

(continued on next page)

Table C-6 (Cont.)
RK06 Bootstrap Loader

Location	Contents	Like This
001032	100200	Set switches 15 and 7 to the up (1) position. Set all others to the down (0) position.
001034	001775	Set switches 9 through 2, and 0 to the up (1) position. Set all others to the down (0) position.
001036	005007	Set switches 11, 9, and 2 through 0 to the up (1) position. Set all others to the down (0) position.

APPENDIX D

RT-11 CONDITIONALS

The following listing contains all the conditionals used in the RT-11 system, including a brief explanation of each of the symbols.

```
BATC$H          ;BATCH Support
BF              ;FB Monitor

CLOCK           ;5Ø/6Ø Hertz clock support
CONT$N          ;Continuation file instead of monitor dot
CR$CSR          ;CSR for CR11 card reader
CR$VEC          ;Vector for CR11 card reader
CSIBFL          ;CSI line buffer size
CT$CSR          ;CSR for the TA11 cassette handler
CT$VEC          ;Vector for the TA11 cassette handler

DCLS$           ;
DCL$S           ;
DLC$Ø           ;CSR of first DL11
DLC$1           ;CSR of second DL11
DLC$2           ;CSR of third DL11
DLC$3           ;CSR of fourth DL11
DLC$4           ;CSR of fifth DL11
DLC$5           ;CSR of sixth DL11
DLC$6           ;CSR of seventh DL11
DLC$7           ;CSR of eighth DL11
DLV$Ø           :vector of first DL11
DLV$1           ;Vector of second DL11
DLV$2           ;Vector of third DL11
DLV$3           ;Vector of fourth DL11
DLV$4           ;Vector of fifth DL11
DLV$5           ;Vector of sixth DL11
DLV$6           ;Vector of seventh DL11
DLV$7           ;vector of eighth DL11
DL$CSR          ;CSR for RLØ1
DL$VEC          ;vector for RLØ1
DL11$L          ;# of local DL11 lines
DL11$M          ;# of remote DL11 lines
DL11$N          ;# of DL11's
DM$CSR          ;CSR for RKØ6/7
DM$VEC          ;vector for RKØ6/7
DP$CSR          ;CSR for RPØ2/3
DP$VEC          ;Vector for RPØ2/3
DRIVEN          ;Number of magtape units
DS$CSR          ;CSR for RJSØ3/4
DS$VEC          ;Vector for RJSØ3/4
DT$VEC          ;vector for TC11 DECtape
DT$CSR          ;CSR for TC11 DECtape
DX$CSR          ;CSR for the RXØ1
```

```
DX$CS2          ;CSR of second RXØ1 controller
DX$VEC          ;Vector for the RXØ1
DX$VC2          ;Vector of second RXØ1 controller
DYT$Ø           ;Second controller support for the RXØ2
DY$CSR          ;CSR for RXØ2
DY$CS2          ;CSR of second RXØ2 controller
DY$DD           ;RXØ2 Double density support only
DY$DS           ;RXØ2 handler attempts single density read first
DY$VC2          ;Vector of second RXØ2 controller
DY$VEC          ;Vector for RXØ2
DZ11$N          ;# of DZ11's
DZ11$L          ;# of local DZ11 lines
DZ11$M          ;# of remote DZ11 lines
DZST$P          ;DZ11 stop unit
DZSP$D          ;Baud rate for DZ11
DZVC$Ø          ;Vector of first DZ11
DZVC$1          ;Vector of second DZ11
DZCS$Ø          ;CSR of first DZ11
DZCS$1          ;CSR of second DZ11

ERL$A           ;Error log handler can write buffers on call
ERL$B           ;Number of blocks in error log buffer
ERL$G           ;Error logging support
ERL$U           ;Number of units to be logged
ERL$V           ;Number of status tables in  error log handler
ERL$W           ;Keep newest errors in error log handler
ESC$P           ;Escape sequence support

FILE$           ;Magtape files support
FPU$11          ;Floating point exception handling support

GT$NM           ;Getname routine resides in the USR or RMON

KW11$P          ;Programmable clock support

LIGH$T          ;Light routine support
LNK$IF          ;LINK overlay counters files
LP$CSR          ;CSR for the line printer
LP$VEC          ;Vector for the line printer

MAT$S           ;Multiterminal asynchronous status support
MMG$T           ;Memory management support
MM$VEC          ;Vector for the TJU16 magtape handler
MM$CSR          ;CSR for the TJU16 magtape handler
MPT$Y           ;Memory parity support
MTT$Y           ;Multi-terminal support
MTI$M           ;Multi-terminal time out support
MT$VEC          ;Vector for TM11 magtape handler
MT$CSR          ;CSR for TM11 magtape handler
MVEC$T          ;Multi-vector support for handlers
MXN$ST          ;Default to file nesting depth

NLKMVLY         ;LIST/NOLIST KMOVLY.MAC
NLKMON          ;LIST/NOLIST KMON.MAC
NLRMON          ;LIST/NOLIST RMON SJ(FB).MAC
NLUSR           ;LIST/NOLIST USR.MAC
```

RT-11 CONDITIONALS

```
PANI$C          ;Panic dump support in SJ
PATC$H          ;Size of PATCH space in RMON
PP$CSR          ;CSR for paper tape punch
PP$VEC          ;vector for paper tape punch
PR$CSR          ;CSR for paper tape reader
PR$VEC          ;vector for paper tape reader
PR11$X          ;Reader only option in the PC handler
PWF$L           ;Powerfail support

RDF$L           ;Default file nesting depth
RF$VEC          ;vector for the RF11
RF$CSR          ;CSR for the RF11
RJSØ$3          ;RJS disk is an RJSØ3 OR RJSØ4
RK$CSR          ;CSR for RKØ5 disk
RK$VEC          ;Vector for RKØ5 disk
RK6$B           ;Bad block support for RKØ6/7
RK6$S           ;Special function for RKØ6/7
RL1$UN          ;Number of units supported in the RLØ1 handler
RPØ$3           ;Specifies RPØ2 or RPØ3
RXT$C           ;Second controller support for the RXØ1

SCC$A           ;^C intercept support (.SCCA)
STAC$K          ;RMON Stack size
STAR$T          ;Start up command file
SYSGN$          ;SYSGENED monitor

TIME$R          ;Timer support in SJ
TIM$IT          ;Device time out support
TKS             ;Console input CSR
TKB             ;Console input data buffer
TPS             ;Console output CSR
TPB             ;Console output data buffer
TTYIN           ;Size of terminal input ring buffer
TTYOUT          ;Size of terminal output ring buffer

U.K.            ;United Kingdom remote line support parameter

V.TKB           ;Console input vector
V.TPS           ;Console output vector
VS6Ø$           ;VS6Ø Support
VT.CSR          ;CSR for VT11/VS6Ø
VT.VEC          ;Vector for VT11/VS6Ø
VT11$           ;VT11 Support
VT5ØHH          ;Set TT HOLD support

$DMSYS          ;RKØ6/7 is the system device
$DLSYS          ;RLØ1 is the system device
$DPSYS          ;RPØ2/RP11C is the system device
$DSSYS          ;RJSØ3/4 is the system device
$DTSYS          ;TA11 is the system device
$DXSYS          ;RXØ1 is the system device
$DYSYS          ;RXØ2 is the system device
$RFNUM          ;# of platters for RF11
$RFSYS          ;RF11/RS11 is the system device
$RKSYS          ;RKØ5 is the system device
```

INDEX

RT-11
System Generation Manual
Order No. AA-5283B-TC

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. Problems with software should be reported on a Software Performance Report (SPR) form. If you require a written reply and are eligible to receive one under SPR service, submit your comments on an SPR form.

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

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

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

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

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

Name ____________ Date ____________

Organization ____________

Street ____________

City ____________ State ____________ Zip Code or Country ____________

Please cut along this line.

-- Fold Here --

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

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

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

Postage will be paid by:

digital

Software Documentation
146 Main Street ML5-5/E39
Maynard, Massachusetts 01754