RT-11
Installation Guide
AA-H376C-TC
RT-11
Installation Guide
AA-H376C-TC

July 1984

This manual tells you how to install the RT-11 operating system. It provides information necessary to bootstrap and preserve the distribution volume(s); create the working system; and customize, preserve, and test the working system.


Operating System: RT-11 Version 5.1
The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

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

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


Printed in U.S.A.

A postage-paid READER'S COMMENTS form is included on the last page of this document. Your comments will assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

- DEC
- DECMate
- DECSYSTEM-10
- DECSYSTEM-20
- DECUS
- DECWri
ter
- DIBOL
- MASSBUS
- PDP
- P/OS
- Professional
- Rainbow
- RSTS
- RSX
- RT-11
- UNIBUS
- VAX
- VMS
- VT
- Work Processor

M33700
Contents

Preface

Chapter 1 Introduction

1.1 Identifying Your Needs ........................................ 1-1
  1.1.1 What Is Your Hardware Configuration? ...................... 1-3
  1.1.2 Which Monitor(s) Do You Need? .............................. 1-3
    1.1.2.1 Base-Line Single-Job Monitor .......................... 1-6
    1.1.2.2 Single-Job Monitor .................................... 1-6
    1.1.2.3 Foreground/Background Monitor ........................ 1-6
    1.1.2.4 Extended Memory Monitor ............................... 1-7
  1.1.3 Which Features Do You Need? ................................ 1-7
    1.1.3.1 Is Installation All You Need to Do? .................... 1-8
    1.1.3.2 Do You Need to Perform the System Generation Procedures? 1-14

1.2 Choosing a Reading Path ........................................ 1-17

Chapter 2 Preparing for Installation

2.1 Surveying Installation Procedures ............................. 2-1
2.2 Studying Software Kit Contents ................................ 2-2
2.3 Selecting Components for Your Working System ................ 2-8
  2.3.1 Monitor ................................................. 2-10
  2.3.2 SWAP.SYS File ........................................... 2-10
  2.3.3 System Device Handler ................................... 2-11
  2.3.4 Other Device Handlers .................................... 2-11
  2.3.5 Default System Library .................................. 2-11
  2.3.6 Exercises ............................................... 2-12
  2.3.7 Help Package ............................................. 2-12
  2.3.8 Line Printer Handlers .................................... 2-13
  2.3.9 MACRO–11 Assembler ...................................... 2-13
  2.3.10 Line Printer Utilities .................................... 2-13
  2.3.11 Source Files ............................................. 2-14
  2.3.12 Start-Up Command File ................................. 2-14
  2.3.13 Text Editors ............................................. 2-14
  2.3.14 Utility Programs ....................................... 2-15
2.4 Planning the Arrangement of Components ............................................. 2–15
  2.4.1 Assigning the Default Device to the Data Device ....................... 2–15
  2.4.2 Creating a Separate Utilities Volume ........................................ 2–16
  2.4.3 Creating Bad Blocks on TU58 DECtape II Cartridges ....................... 2–17
  2.4.4 Creating Several System Volumes ............................................ 2–18
  2.4.5 Creating Volumes for Use with Introduction to RT–11 ....................... 2–18
  2.4.6 Limiting Components on the System Volume ................................... 2–20

2.5 Installing Software Updates .......................................................... 2–20

2.6 Acquiring Sufficient Media ............................................................ 2–21

2.7 Choosing Software Customizations ................................................. 2–21
  2.7.1 Changing Characters That Indicate Insertion/Deletion .................... 2–22
  2.7.2 Changing Default Output Device from Line Printer to Terminal ........ 2–23
  2.7.3 Changing Location of VT11/VS60 Floating Vectors ........................ 2–23
  2.7.4 Changing the Number of Directory Columns ................................... 2–24
  2.7.5 Changing the Default Order of Directory Listings ......................... 2–24
  2.7.6 Changing the Number of /Q Program Sections LINK Allows ................ 2–25
  2.7.7 Changing the Size of LINK's Library Module List .......................... 2–25
  2.7.8 Changing the Size of the QUEUE Work File .................................. 2–26
  2.7.9 Modifying EDIT ................................................................. 2–27
     2.7.9.1 Size of the Text Window ............................................... 2–27
     2.7.9.2 Terminals with Nonstandard ESCAPE Codes ............................ 2–27
  2.7.10 Extracting the Overlay Handlers from SYSLIB .............................. 2–28
  2.7.11 Installing Other Devices ....................................................... 2–29
     2.7.11.1 CSR and Vector Addresses for Line Printers, DECTape II, and MSCP Disks ..................................................... 2–30
     2.7.11.2 Hardware Magtape Support ............................................. 2–31
     2.7.11.3 Magtape Parity and Density ........................................... 2–33
     2.7.11.4 RP03 Support ............................................................... 2–33
  2.7.12 Modifying BATCH to Save Space .............................................. 2–33
  2.7.13 Modifying LINK to Change the Default SYSLIB Device ...................... 2–34
  2.7.14 Modifying the Help Text ....................................................... 2–35
  2.7.15 Preventing Fatal System Errors from Causing a Reset ................. 2–37
  2.7.16 Running RT–11 in Less Memory Than Is Available ....................... 2–37
  2.7.17 Setting VTCOM Default Dial String ......................................... 2–39
  2.7.18 Setting Upper Limit on a File Size ......................................... 2–39
  2.7.19 Specifying 50-Cycle Instead of 60-Cycle Clock Rate ..................... 2–40
  2.7.20 Using CAPS–11 to Load Files ............................................... 2–40
  2.7.21 Setting Upper Limit on Memory Size ....................................... 2–41
  2.7.22 Suppressing the Bootstrap Message ......................................... 2–41
  2.7.23 Suppressing the Start-Up Indirect Command File .......................... 2–41
  2.7.24 Suppressing the Start-Up Indirect Command File Echo .................... 2–42
  2.7.25 Changing the Bootstrap Message ............................................. 2–42
  2.7.26 Changing the Default Device for Indirect Command Files ................ 2–42
  2.7.27 Changing the Default File Type for Indirect Command Files .............. 2–43
  2.7.28 Changing the Default Device for the FRUN Command ........................ 2–43
  2.7.29 Changing the Default File Type for the FRUN Command .................... 2–44
  2.7.30 Changing the Default Device for the EDIT Command ......................... 2–44
  2.7.31 Changing the Default File Name for the EDIT Command .................... 2–45
  2.7.32 Using Examine and Deposit Above the Background Job ..................... 2–45
  2.7.33 Changing the Default Device for QUEMAN ................................... 2–45
Chapter 3 Installing a System Distributed on RX01 to Run on a Small Device

3.1 Bootstrapping the Distribution Volume ........................................ 3-2
3.2 Preserving the Distribution Volumes ........................................... 3-3
3.3 Installing Software Updates ..................................................... 3-7
3.4 Creating the Working System from Chosen Components ................... 3-7
3.5 Installing the Bootstrap on Any Volumes That Need to Be Bootable ...... 3-9
3.6 Customizing the System .......................................................... 3-9
3.7 Compressing Each Volume ....................................................... 3-10
3.8 Preserving the Working System ................................................ 3-10
3.9 Testing the Working System .................................................... 3-11

3.9.1 Preparing the Background Demonstration Program ....................... 3-12
  3.9.1.1 Edit the Background Demonstration Program ....................... 3-12
  3.9.1.2 Assemble the Background Demonstration Program ................. 3-12
  3.9.1.3 Link the Background Demonstration Program ...................... 3-14
  3.9.1.4 Run the Background Demonstration Program ...................... 3-14

3.9.2 Preparing the Foreground Demonstration Program ........................ 3-14
  3.9.2.1 Assemble the Foreground Demonstration Program .................. 3-14
  3.9.2.2 Link the Foreground Demonstration Program ...................... 3-15
  3.9.2.3 Run the Foreground and Background Demonstration Programs ...... 3-15

3.10 Performing the System Generation Process ................................ 3-16

Chapter 4 Installing a System Distributed on RX01 to Run on a Hard Disk

4.1 Bootstrapping the Distribution Volume ...................................... 4-2
4.2 Copying the Distribution Volumes to the Disk ............................ 4-3
4.3 Preserving the Distribution Volumes ........................................ 4-5
Chapter 5 Installing a System Distributed on Hard Disk to Run on Hard Disk

5.1 Bootstrapping the Distribution Disk ........................................ 5–3
5.2 Preserving the Distribution Disk .............................................. 5–4
5.3 Installing Software Updates ..................................................... 5–6
5.4 Creating the Working System from Chosen Components .......... 5–6
5.5 Installing the Bootstrap on the Disk .......................................... 5–7
5.6 Customizing the System .......................................................... 5–8
5.7 Compressing the Disk .............................................................. 5–8
5.8 Preserving the Working System ................................................ 5–9
5.9 Testing the Working System ....................................................... 5–10

5.9.1 Preparing the Background Demonstration Program .............. 5–11
5.9.1.1 Edit the Background Demonstration Program ................. 5–11
5.9.1.2 Assemble the Background Demonstration Program ......... 5–11
5.9.1.3 Link the Background Demonstration Program ............... 5–12
5.9.1.4 Run the Background Demonstration Program ............... 5–12

5.9.2 Preparing the Foreground Demonstration Program ................ 5–13
5.9.2.1 Assemble the Foreground Demonstration Program .......... 5–13
5.9.2.2 Link the Foreground Demonstration Program ............... 5–13
5.9.2.3 Run the Foreground and Background Demonstration Programs .. 5–14

5.10 Performing the System Generation Process .............................. 5–15

Chapter 6 Installing a System Distributed on Hard Disk to Run on a Small Device

6.1 Bootstrapping the Distribution Disk ........................................ 6–2
6.2 Preserving the Distribution Disk .............................................. 6–3
6.3 Installing Software Updates ........................................ 6–5
6.4 Creating the Working System from Chosen Components .......... 6–6
  6.4.1 Initializing RX50 Diskettes ..................................... 6–6
  6.4.2 Formatting and Initializing RX01/RX02 Diskettes ............. 6–7
  6.4.3 Copying the Selected Files onto Diskette ...................... 6–7
6.5 Installing the Bootstrap on Any Diskettes That Need to Be Bootable . 6–8
6.6 Customizing the System ............................................ 6–9
6.7 Compressing Each Diskette ........................................ 6–9
6.8 Preserving the Working System ................................... 6–10
6.9 Testing the Working System ....................................... 6–11
  6.9.1 Preparing the Background Demonstration Program ............. 6–12
    6.9.1.1 Edit the Background Demonstration Program .................. 6–12
    6.9.1.2 Assemble the Background Demonstration Program .......... 6–12
    6.9.1.3 Link the Background Demonstration Program .............. 6–14
    6.9.1.4 Run the Background Demonstration Program ............... 6–14
  6.9.2 Preparing the Foreground Demonstration Program ............. 6–14
    6.9.2.1 Assemble the Foreground Demonstration Program .......... 6–15
    6.9.2.2 Link the Foreground Demonstration Program .............. 6–15
    6.9.2.3 Run the Foreground and Background Demonstration Programs ....... 6–15
6.10 Performing the System Generation Process ........................ 6–16

Chapter 7 Installing a System Distributed on RX02 to Run on RX02

  7.1 Bootstrapping the Distribution Diskette .......................... 7–2
  7.2 Installing Software Updates ..................................... 7–4
  7.3 Creating the Working System from Chosen Components ............ 7–4
  7.4 Installing the Bootstrap on Any Diskettes That Need to Be Bootable . 7–6
  7.5 Customizing the System ........................................... 7–7
  7.6 Compressing Each Diskette ....................................... 7–7
  7.7 Preserving the Working System ................................... 7–8
  7.8 Testing the Working System ...................................... 7–8
  7.8.1 Preparing the Background Demonstration Program ............ 7–10
    7.8.1.1 Edit the Background Demonstration Program .................. 7–10
    7.8.1.2 Assemble the Background Demonstration Program ............ 7–10
    7.8.1.3 Link the Background Demonstration Program .............. 7–11
    7.8.1.4 Run the Background Demonstration Program ............... 7–11
  7.8.2 Preparing the Foreground Demonstration Program ............ 7–12
    7.8.2.1 Assemble the Foreground Demonstration Program .......... 7–12
    7.8.2.2 Link the Foreground Demonstration Program .............. 7–12
    7.8.2.3 Run the Foreground and Background Demonstration Programs ....... 7–12
  7.9 Performing the System Generation Process ........................ 7–14

Chapter 8 Installing a System Distributed on Magtape to Run on Hard Disk

  8.1 Bootstrapping the First Distribution Magtape .................... 8–2
  8.2 Preserving Both Distribution Magtapes ........................... 8–4
  8.3 Installing Software Updates ..................................... 8–6
Chapter 9 Installing RT-11 on a MICRO/PDP-11 Processor

9.1 Bootstrapping the System Diskette ........................................... 9-2
9.2 Installing Software Updates ..................................................... 9-3
9.3 Copying the System Diskette onto the System Disk ....................... 9-3
9.4 Installing the Bootstrap on the System Disk ............................... 9-4
9.5 Copying the Distribution Diskettes onto the System Disk ............... 9-4
9.6 Creating the Working System from Chosen Components .................. 9-5
9.7 Customizing the System .......................................................... 9-5
9.8 Compressing the System ......................................................... 9-6
9.9 Preserving the Working System ............................................... 9-6
9.10 Testing the Working System .................................................... 9-8
9.10.1 Preparing the Background Demonstration Program .................... 9-9
  9.10.1.1 Edit the Background Demonstration Program ......................... 9-9
  9.10.1.2 Assemble the Background Demonstration Program .................. 9-9
  9.10.1.3 Link the Background Demonstration Program ......................... 9-10
  9.10.1.4 Run the Background Demonstration Program ......................... 9-10
9.10.2 Preparing the Foreground Demonstration Program ..................... 9-11
  9.10.2.1 Assemble the Foreground Demonstration Program .................. 9-11
  9.10.2.2 Link the Foreground Demonstration Program ......................... 9-12
  9.10.2.3 Run the Foreground and Background Demonstration Programs .... 9-12
9.11 Performing the System Generation Process ................................ 9-13

Chapter 10 Installing RT-11 on a Professional 325 Computer

10.1 Bootstrapping the System Diskette ......................................... 10-2
10.2 Installing Software Updates .................................................. 10-3
10.3 Creating the Working System from Chosen Components .................. 10-3
10.4 Installing the Bootstrap on Diskettes that Must Be Bootable .......... 10-5
10.5 Customizing the System .................................. 10–6
10.6 Preserving the Working System ...................... 10–7
10.7 Testing the Working System .......................... 10–8
10.7.1 Preparing the Background Demonstration Program .................. 10–9
  10.7.1.1 Edit the Background Demonstration Program .............. 10–9
  10.7.1.2 Assemble the Background Demonstration Program ...... 10–9
  10.7.1.3 Link the Background Demonstration Program ......... 10–10
  10.7.1.4 Run the Background Demonstration Program ...... 10–11

10.7.2 Preparing the Foreground Demonstration Program .......... 10–11
  10.7.2.1 Assemble the Foreground Demonstration Program ... 10–11
  10.7.2.2 Link the Foreground Demonstration Program ....... 10–12

10.7.3 Run the Foreground and Background Demonstration Programs ............. 10–12

10.8 Performing the System Generation Procedures .............. 10–13

Chapter 11 Installing RT–11 on a Professional 350 Computer

11.1 Bootstrapping the System Diskette ...................... 11–2
11.2 Installing Software Updates .......................... 11–3
11.3 Copying the System Diskette onto the System Disk .......... 11–4
11.4 Installing the Bootstrap on the System Disk .............. 11–4
11.5 Copying the Distribution Diskettes onto the System Disk ...... 11–5
11.6 Creating the Working System from Chosen Components ....... 11–5
11.7 Customizing the System .............................. 11–5
11.8 Compressing the System Disk .......................... 11–5
11.9 Preserving the Working System ........................ 11–6
11.10 Testing the Working System ........................... 11–7

11.10.1 Preparing the Background Demonstration Program ............ 11–9
  11.10.1.1 Edit the Background Demonstration Program .......... 11–9
  11.10.1.2 Assemble the Background Demonstration Program ... 11–9
  11.10.1.3 Link the Background Demonstration Program ....... 11–10
  11.10.1.4 Run the Background Demonstration Program ....... 11–10

11.10.2 Preparing the Foreground Demonstration Program .......... 11–11
  11.10.2.1 Assemble the Foreground Demonstration Program .... 11–11
  11.10.2.2 Link the Foreground Demonstration Program ....... 11–11

11.10.3 Run the Foreground and Background Demonstration Programs .... 11–12

11.11 Performing the System Generation Procedures .............. 11–13

Appendix A Modifying MDUP to Recognize a Device Handler

Appendix B Loading Software Bootstraps

  B.1 Loading the RK11 (RK05) DECPack Bootstrap .............. B–1
  B.2 Loading the TC11 DECTape Bootstrap .................... B–2
  B.3 Loading the RX11 Bootstrap ........................... B–4
B.4 Loading the TJU16 or TM11 Magtape Bootstrap ................. B–5
B.5 Loading the RK06/RK07 DECpack Bootstrap .................. B–7
B.6 Loading the RL01/RL02 Disk Bootstrap ....................... B–8
B.7 Loading the RX211 Bootstrap ............................... B–10
B.8 Loading the TSV05 Bootstrap Using MICRO-ODT .......... B–12
B.9 Loading the TSV05 Bootstrap Using the Switch Register ... B–13

Index

Figures

1–1 RT–11 Installation Process ................................ 1–2
2–1 Block Locations on DECTape II ............................... 2–18
2–2 Installation Worksheet ........................................ 2–58
3–1 Sample Backup Volumes ...................................... 3–2
3–2 DEMOBG Assembly Listing ................................... 3–13
4–1 Sample Backup Disks .......................................... 4–2
4–2 DEMOBG Assembly Listing ................................... 4–12
5–1 Sample Backup Disks (RL01, RL02, RK05) ................. 5–2
5–2 Sample Backup Disks (RC25) ................................. 5–2
5–3 DEMOBG Assembly Listing ................................... 5–12
6–1 Sample Backup Disk and Diskettes ......................... 6–2
6–2 DEMOBG Assembly Listing ................................... 6–13
7–1 Sample Backup Diskettes ..................................... 7–2
7–2 DEMOBG Assembly Listing ................................... 7–11
8–1 Sample Backup Magtapes ...................................... 8–2
8–2 DEMOBG Assembly Listing ................................... 8–13
9–1 Sample Backup Diskettes ..................................... 9–2
9–2 DEMOBG Assembly Listing ................................... 9–10
10–1 Sample Backup Diskettes .................................... 10–2
10–2 DEMOBG Assembly Listing .................................. 10–10
11–1 Sample Backup Diskettes .................................... 11–2
11–2 DEMOBG Assembly Listing .................................. 11–10

Tables

1–1 Features Available in Distributed Monitors .................. 1–3
1–2 Base-Line Monitor Versus Single-Job Monitor ............... 1–6
1–3 Features Available Through Simple Software Customizations 1–8
1–4 Features Available Only Through the System Generation Procedures ........................................ 1–15
1–5 Chapters You Should Read ..................................... 1–18
2–1 RT–11 Software Components ................................. 2–2
2–2 Device Size .................................................... 2–9
2–3 File-Structured and Hardware Magtape Handlers for SJ and FB Monitors ........................................... 2–33
2–4 File-Structured and Hardware Magtape Handlers for XM Monitors ........................................... 2–33
B–1 RK11 (RK05) Bootstrap Loader .............................. B–2
B–2 TC11 Bootstrap Loader ........................................ B–3
B–3 RX11 Bootstrap Loader ........................................ B–4
B–4 TJU16 Bootstrap Loader ....................................... B–6
B–5 TM11 Bootstrap Loader ........................................ B–7
B–6 RK06/RK07 Bootstrap Loader ................................. B–8
B–7 RL01/RL02 Bootstrap Loader ................................ B–9
B–8 RX211 Bootstrap Loader ........................................ B–11
B–9 TSV05 Bootstrap Loader ........................................ B–13
Preface

How to Use This Manual

This manual introduces you to the RT–11 Version 5 software kit and tells you how to install your operating system. The modular structure of this manual allows you to read only the chapters that you need for your particular combination of hardware and software.

Chapter 1 of the manual introduces you to installation and system generation concepts and helps you choose a reading path through the other chapters.

Installation concepts — both planning activities and actual installation procedures — are covered in Chapters 2 through 11. You should read Chapter 2 to plan your installation, then choose the chapter (3, 4, 5, 6, 7, 8, 9, 10 or 11) that describes the installation of a system most closely resembling your own configuration.

Additional information available in the appendixes includes BATCH streams to build system programs and toggle-in software bootstraps.

If you discover that you need to perform the system generation process, refer to the *RT–11 System Generation Guide*.

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

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

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

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

2. Unless the manual indicates otherwise, all commands or command strings end with a carriage return. The symbol \( \text{\textasciitilde} \) represents a carriage return, \( \text{\textasciicircum} \) a line feed, \( \text{\textasciitilde} \) a space, \( \text{\textasciitilde} \) an ESCAPE or ALTMODE, and \( \text{\textasciitilde} \) a tab.

3. To produce certain characters in system commands, you must type a combination of keys concurrently. For example, while holding down the CTRL key, type C to produce the CTRL/C character. Key combinations such as this are documented as \( \text{\textasciitilde} \), \( \text{\textasciitilde} \), and so forth.

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

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

   Letter O:    0
   Number 0:    0

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

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

7. A decimal point (.) follows a number to indicate that it is a decimal number. A number without a decimal point is an octal number. For example, 128. is 128 (decimal) and 126 is 126 (octal).
Chapter 1
Introduction

Now that your PDP–11 processor or Professional 300 series computer has been installed and your software distribution kit has been delivered, you are ready to get started with RT–11. To begin, you must install RT–11. Installation requires bootstrapping and preserving the distribution volumes, installing software updates if necessary, creating the working system from selected components, and customizing, preserving, and testing the working system. In most cases, your system is ready to run once you have installed and tested it. However, if you need certain special features, for example, multiterminal support, you must perform the system generation process. In this case, you must run the SYSGEN program, answer its questions, and assemble your own generated monitor(s) and device handlers. Figure 1–1 summarizes the flow of the installation process. Refer to the RT–11 System Generation Guide for specific information about generating specialized monitors.

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

1.1 Identifying Your Needs

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

- What is your hardware configuration?
- Which monitor(s) do you need?
- Which features do you need?
1.1.1 What Is Your Hardware Configuration?

Your hardware configuration is an important consideration for three reasons:

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

Chapters 3 through 11 contain detailed procedures for installing RT–11 systems on certain common hardware configurations. Unless you have an unusual configuration, read only the chapter intended for your configuration. (See Table 1–5 to identify your reading path.)

1.1.2 Which Monitor(s) Do You Need?

You have to decide which monitor or monitors you need in order to know what steps to follow in getting started. Your distribution kit includes a base-line single-job monitor (BL), a single-job monitor (SJ), a foreground/background monitor (FB), and an extended memory monitor (XM). Sections 1.1.2.1 through 1.1.2.4 describe the various monitors, and Table 1–1 summarizes the features available in the distributed monitors. The sections and the table give you the information you need to make a decision about your monitor requirements.

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Monitor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BL</td>
</tr>
<tr>
<td>Memory limits</td>
<td>16K to</td>
</tr>
<tr>
<td></td>
<td>30K words</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Size of resident monitor</td>
<td>1.8K words</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Size of disk image</td>
<td>71 blocks (approx.)</td>
</tr>
<tr>
<td>FG job support</td>
<td>no</td>
</tr>
<tr>
<td>System job support</td>
<td>no</td>
</tr>
<tr>
<td>Timer support (.MRKT and .CMKT — including midnight date/time rollover)</td>
<td>no</td>
</tr>
<tr>
<td>Error messages on system I/O errors</td>
<td>no*</td>
</tr>
</tbody>
</table>

* Feature enabled/disabled by system generation.
<table>
<thead>
<tr>
<th>Characteristics</th>
<th>BL</th>
<th>SJ</th>
<th>FB</th>
<th>XM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiterminal support</td>
<td>no*</td>
<td>no*</td>
<td>no*</td>
<td>no*</td>
</tr>
<tr>
<td>60Hz clock frequency</td>
<td>yes*</td>
<td>yes*</td>
<td>yes*</td>
<td>yes*</td>
</tr>
<tr>
<td>50Hz clock frequency instead of 60Hz</td>
<td>no*</td>
<td>no*</td>
<td>no*</td>
<td>no*</td>
</tr>
<tr>
<td>Programmable clock instead of line as system clock</td>
<td>no</td>
<td>no*</td>
<td>no*</td>
<td>no*</td>
</tr>
<tr>
<td>Start-up command file</td>
<td>yes*</td>
<td>yes*</td>
<td>yes*</td>
<td>yes*</td>
</tr>
<tr>
<td>Memory parity support</td>
<td>no</td>
<td>no*</td>
<td>no*</td>
<td>yes*</td>
</tr>
<tr>
<td>Month and year date rollover support</td>
<td>no</td>
<td>no*</td>
<td>no*</td>
<td>no*</td>
</tr>
<tr>
<td>Fetchable device handlers</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes*</td>
</tr>
<tr>
<td>User command linkage support</td>
<td>no</td>
<td>yes*</td>
<td>yes*</td>
<td>yes*</td>
</tr>
<tr>
<td>Concise command language</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>High-speed ring buffer</td>
<td>no</td>
<td>no*</td>
<td>no*</td>
<td>no*</td>
</tr>
<tr>
<td>Contains all keyboard monitor commands</td>
<td>yes*</td>
<td>yes*</td>
<td>yes*</td>
<td>yes*</td>
</tr>
<tr>
<td>Floating point support</td>
<td>no</td>
<td>yes*</td>
<td>yes*</td>
<td>yes*</td>
</tr>
<tr>
<td>Error message on powerfail</td>
<td>no</td>
<td>no*</td>
<td>no*</td>
<td>no*</td>
</tr>
<tr>
<td>BATCH</td>
<td>no</td>
<td>no*</td>
<td>no*</td>
<td>no*</td>
</tr>
<tr>
<td>Error logging</td>
<td>no</td>
<td>no*</td>
<td>no*</td>
<td>no*</td>
</tr>
<tr>
<td>MSCP devices</td>
<td>yes†</td>
<td>yes†</td>
<td>yes†</td>
<td>yes†</td>
</tr>
<tr>
<td>RD50/RD51</td>
<td>no</td>
<td>no</td>
<td>yes†</td>
<td>yes†</td>
</tr>
<tr>
<td>RK11/RK05</td>
<td>yes†</td>
<td>yes†</td>
<td>yes†</td>
<td>yes†</td>
</tr>
<tr>
<td>RL01/RL02</td>
<td>yes†</td>
<td>yes†</td>
<td>yes†</td>
<td>yes†</td>
</tr>
<tr>
<td>(2 drives)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RJS03/RJS04</td>
<td>yes†</td>
<td>yes†</td>
<td>yes†</td>
<td>no‡</td>
</tr>
<tr>
<td>(RJS03)</td>
<td></td>
<td></td>
<td></td>
<td>(RJS03)</td>
</tr>
<tr>
<td>RK06/RK07</td>
<td>yes†</td>
<td>yes†</td>
<td>yes†</td>
<td>yes†</td>
</tr>
<tr>
<td>RF11</td>
<td>yes†</td>
<td>yes‡</td>
<td>yes‡</td>
<td>no‡</td>
</tr>
<tr>
<td>RP11/RPR02/RP03</td>
<td>yes†</td>
<td>yes‡</td>
<td>yes‡</td>
<td>no‡</td>
</tr>
</tbody>
</table>

* Feature enabled/disabled by system generation.  
† Refer to Section 2.7.11.  
‡ RT-11 no longer supports this device; however, the handler is included in the software kit.

(continued on next page)
<table>
<thead>
<tr>
<th>Characteristics</th>
<th>BL</th>
<th>SJ</th>
<th>FB</th>
<th>XM</th>
</tr>
</thead>
<tbody>
<tr>
<td>RX01</td>
<td>yes† (2 drives)</td>
<td>yes† (2 drives)</td>
<td>yes† (2 drives)</td>
<td>yes† (2 drives)</td>
</tr>
<tr>
<td>RX02</td>
<td>yes† (2 drives)</td>
<td>yes† (2 drives)</td>
<td>yes† (2 drives)</td>
<td>yes† (2 drives)</td>
</tr>
<tr>
<td>RX50</td>
<td>no</td>
<td>no</td>
<td>yes†</td>
<td>yes†</td>
</tr>
<tr>
<td>DECTape</td>
<td>yes‡</td>
<td>yes‡</td>
<td>yes‡</td>
<td>no‡</td>
</tr>
<tr>
<td>DECTape II</td>
<td>yes†</td>
<td>yes†</td>
<td>yes†</td>
<td>yes†</td>
</tr>
<tr>
<td>PDT–11 intelligent terminal</td>
<td>yes‡</td>
<td>yes‡</td>
<td>yes‡</td>
<td>no</td>
</tr>
<tr>
<td>Virtual memory handler</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes (as data device only)</td>
</tr>
<tr>
<td>File-structured magtape — TM11</td>
<td>yes (2 units)</td>
<td>yes (2 units)</td>
<td>yes (2 units)</td>
<td>yes (2 units)</td>
</tr>
<tr>
<td>File-structured magtape — TJU16</td>
<td>yes (2 units)</td>
<td>yes (2 units)</td>
<td>yes (2 units)</td>
<td>yes (2 units)</td>
</tr>
<tr>
<td>File-structured magtape — TS11</td>
<td>yes (2 units)</td>
<td>yes (2 units)</td>
<td>yes (2 units)</td>
<td>yes (2 units)</td>
</tr>
<tr>
<td>Hardware magtape TM11</td>
<td>no§</td>
<td>no§</td>
<td>no§</td>
<td>no§</td>
</tr>
<tr>
<td>Hardware magtape TJU16</td>
<td>no§</td>
<td>no§</td>
<td>no§</td>
<td>no§</td>
</tr>
<tr>
<td>Hardware magtape TS11</td>
<td>no§</td>
<td>no§</td>
<td>no§</td>
<td>no§</td>
</tr>
<tr>
<td>TA11 cassette</td>
<td>yes‡</td>
<td>yes‡</td>
<td>yes‡</td>
<td>no</td>
</tr>
<tr>
<td>VT11/VS60</td>
<td>no</td>
<td>no*</td>
<td>no*</td>
<td>no</td>
</tr>
<tr>
<td>Line printer</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Serial printer</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>High-speed paper tape reader/punch</td>
<td>yes‡</td>
<td>yes‡</td>
<td>yes‡</td>
<td>no‡</td>
</tr>
<tr>
<td>CR11 card reader</td>
<td>yes‡</td>
<td>yes‡</td>
<td>yes‡</td>
<td>no</td>
</tr>
<tr>
<td>Null handler</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Logical disk handler</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Single-line editor</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Extra device slots</td>
<td>0*</td>
<td>0*</td>
<td>0*</td>
<td>4*</td>
</tr>
</tbody>
</table>

* Feature enabled/disabled by system generation.
† Refer to Section 2.7.11.
‡ RT–11 no longer supports this device; however, the handler is included in the software kit.
§ Refer to Section 2.7.11.2.
1.1.2.1 Base-Line Single-Job Monitor — The base-line single-job monitor (BL) is a stripped-down version of the single-job monitor (SJ). The BL monitor does not support optional monitor and device functions. It has the smallest residency requirement of any RT–11 monitor and is intended for use in applications that require small monitor size and minimal executive support.

Table 1–2 summarizes the differences between the distributed BL and SJ monitors.

Table 1–2: Base-Line Monitor Versus Single-Job Monitor

<table>
<thead>
<tr>
<th>BL Monitor</th>
<th>SJ Monitor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does not support graphic display.</td>
<td>Supports graphic display terminal (VT11/VS60) as ASCII console terminal.</td>
</tr>
<tr>
<td>System halts on power failure or system device I/O error.</td>
<td>System prints error message on power failure or system device I/O error.</td>
</tr>
<tr>
<td>Supports only one terminal.</td>
<td>Can be generated to support more than one terminal.</td>
</tr>
<tr>
<td>Minimal memory and system device requirements for monitor.</td>
<td>Requires slightly larger memory and system device for monitor.</td>
</tr>
<tr>
<td>Does not include floating point support.</td>
<td>Includes floating point support.</td>
</tr>
</tbody>
</table>

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

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

The SJ monitor supports all hardware devices (except memory management hardware) and runs all the system utilities except QUEUE or SPOOL. It runs in any supported configuration, except Professional 300 series processors, with at least 16K words of memory but cannot make use of more than 28K words of memory (30K on an LSI–11 processor).

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

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

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

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

The minimum memory requirement for the FB monitor is 24K words. The FB monitor resides in 8K or fewer words of memory and cannot use more than 28K words of memory (30K on an LSI-11 processor).

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

1.1.2.4 Extended Memory Monitor — The extended memory monitor (XM) is the largest and most powerful of the RT-11 executives. It has all the facilities of the FB monitor; in addition, it can support up to 128K words of memory in systems supporting 18-bit architecture and up to 2 megawords of memory in Q-bus systems supporting 22-bit architecture. The XM monitor provides a set of programmed requests that allow advanced applications to make use of additional memory above 28K (using the PDP-11 memory management unit).

The XM monitor is significantly larger than the FB monitor. The XM monitor's requirement that the USR be resident adds even more to the monitor memory overhead. Consequently, if you have no more than 48K words of memory, choose one of the other monitors. Also, if your application involves program development or execution of an end-user application that does not make use of extended memory facilities, you may opt for the FB or SJ monitor.

The XM monitor requires 32K words of memory, a memory management unit (KT11 hardware), and the extended instruction set (EIS) option to operate.

1.1.3 Which Features Do You Need?

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

1.1.3.1 Is Installation All You Need to Do? — You need to decide whether installation is all you need to do or whether you will also need to perform the system generation process. Table 1-3 summarizes the features that you can add to your system by performing simple customizations during installation. Section 2.7 describes these software customizations in detail and provides instructions for implementing them. Many users will not need to make any of these customizations.

Table 1-3: Features Available Through Simple Software Customizations

<table>
<thead>
<tr>
<th>Feature</th>
<th>Section</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adding subroutines to SYSLIB.OBJ</td>
<td></td>
<td>To add subroutines to the default system library, SYSLIB.OBJ, use the librarian. The RT-11 linker searches SYSLIB to resolve undefined globals at the end of a link operation. Normally, you should customize SYSLIB to contain the system subroutines, FORTRAN OTS routines, and application-specific subroutines.</td>
</tr>
<tr>
<td>Changing characters SRCCOM uses to indicate insertions and deletions</td>
<td>2.7.1</td>
<td>Normally, when you use the SRCCOM utility or DIFFERENCES/CHANGEBAR command to compare two files, SRCCOM uses the vertical bar character to indicate insertions and the bullet (alphabetic o) character to indicate deletions. You can modify SRCCOM to use different characters.</td>
</tr>
<tr>
<td>Changing default output device from line printer to terminal</td>
<td>2.7.2</td>
<td>If your configuration does not include a line printer, you can change the default output device (which certain monitor commands use) from line printer to terminal. To make this change, you can edit the start-up command file.</td>
</tr>
<tr>
<td>Changing location of VT11/VS60 floating vectors</td>
<td>2.7.3</td>
<td>If you add certain other devices to a configuration that includes a VT11 or VS60, you must modify the monitor to move the VT11 or VS60 vector addresses.</td>
</tr>
<tr>
<td>Changing number of directory columns</td>
<td>2.7.4</td>
<td>When you use the DIRECTORY command, the directory prints in two columns. When you use the DIRECTORY/BRIEF command, the directory contains five columns with less information. You can modify DIR to change the default number of columns.</td>
</tr>
<tr>
<td>Changing default order of directory listings</td>
<td>2.7.5</td>
<td>By default, DIR lists a directory's contents by physical position. You can modify DIR to change the default to any one of five orders: creation date, file name, file size, file type, or position.</td>
</tr>
</tbody>
</table>

(continued on next page)
Table 1–3: **Features Available Through Simple Software Customizations** (Cont.)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Section</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changing number of /Q p-sects LINK allows</td>
<td>2.7.6</td>
<td>You can modify LINK to change the number of absolute base-address p-sects that LINK allows. Normally, LINK's /Q option lets you specify the absolute base addresses of up to eight p-sects in your program.</td>
</tr>
<tr>
<td>Changing size of LINK's library module list</td>
<td>2.7.7</td>
<td>To change the default size of LINK's list of library modules, you can modify LINK. LINK normally creates a list of 252 (octal) modules to be included from libraries during the link operation. If you are a DIBOL user, you must customize the software.</td>
</tr>
<tr>
<td>Changing size of QUEUE work file</td>
<td>2.7.8</td>
<td>You can modify QUEUE to change the size of the work file QUEUE uses to queue files to be sent to a device. If you increase the size of the work file, QUEUE can handle more files at one time.</td>
</tr>
<tr>
<td>Modifying EDIT</td>
<td>2.7.9</td>
<td>In certain applications using VT11/VS60 terminals, the text window overflows onto the scrolled editing commands, making a portion of the screen difficult to read. You can alleviate this problem by changing EDIT to reduce the size of the text window.</td>
</tr>
<tr>
<td>Extracting overlay handlers from SYSLIB.OBJ</td>
<td>2.7.10</td>
<td>If you intend to use overlaid programs, you need the overlay handlers that are included in SYSLIB. If you are a MACRO-only user and need only the overlay handlers, you can extract them from SYSLIB for use as a separate library.</td>
</tr>
<tr>
<td>Installing other devices</td>
<td>2.7.11</td>
<td>When you boot the system device, the bootstrap installs the devices in your hardware configuration if the appropriate device handler is on the system device and if there are enough device slots. If you want to install a device that the bootstrap does not install, use the REMOVE and INSTALL commands. You can also:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Change LP, LS, DD, RK, DX, DY, DL, DM, or DU addresses:</em> If any of these devices are installed at nonstandard CSR and vector addresses, you can use the monitor SET command to change the addresses.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Install hardware magtape:</em> If you need hardware magtape support, you must perform a system generation and then rename the hardware handler to MT.SYS, MM.SYS, or MS.SYS.</td>
</tr>
</tbody>
</table>

(continued on next page)
<table>
<thead>
<tr>
<th>Feature</th>
<th>Section</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installing other devices (cont.)</td>
<td></td>
<td><em>Set magtape parity and density:</em> If you need to set magtape parity or density (other than the standard 800 bits/in, odd parity), you can use the monitor SET command for TM11 or TJU16 magtape (but not TS11).</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Change RP02 support to RP03:</em> To use RP03 disk drives, you must think of each RP03 drive as two logical units of 40,000 blocks.</td>
</tr>
<tr>
<td>Modifying BATCH to save space</td>
<td>2.7.12</td>
<td>If you need to save disk space, you can store certain system programs on DK: rather than on SY: and cause BATCH to access them there. Modify BATCH so that it invokes system programs with the monitor RUN command (which assumes DK: as the default) rather than with the R command (which always uses SY:).</td>
</tr>
<tr>
<td>Modifying LINK to change default SYSLIB device</td>
<td></td>
<td>By customizing the linker, you can change the device on which the default system library, SYSLIB.OBJ, resides.</td>
</tr>
<tr>
<td>Modifying help text</td>
<td>2.7.14</td>
<td>To change the help text that prints when you use the HELP command, you must create your own help text file, process that text file with LIBR, and copy the resulting library and the file HELP.EXE to the same volume.</td>
</tr>
<tr>
<td>Preventing fatal system errors from causing a reset</td>
<td>2.7.15</td>
<td>You can customize the monitor to prevent it from performing a hard reset of errors when a fatal system error occurs. The normal reset stops I/O, protecting media from being corrupted. Some hardware errors may be more easily diagnosed without this reset.</td>
</tr>
<tr>
<td>Running RT-11 in less memory than is available</td>
<td>2.7.16</td>
<td>You can customize your software if your application requires that RT-11 run in less memory than is available.</td>
</tr>
<tr>
<td>Setting VTCOM default dial string</td>
<td>2.7.17</td>
<td>You can modify VTCOM to set a default string for the DIAL command.</td>
</tr>
<tr>
<td>Setting upper limit on file size</td>
<td>2.7.18</td>
<td>If your application requires an upper limit to the size of a file, you can change the monitor to set the limit you require.</td>
</tr>
<tr>
<td>Specifying 50-cycle clock rate</td>
<td>2.7.19</td>
<td>To use a 50-cycle clock rate rather than the 60-cycle clock of the standard monitors, you must modify the monitor.</td>
</tr>
</tbody>
</table>

(continued on next page)
Table 1–3: **Features Available Through Simple Software Customizations (Cont.)**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Section</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using CAPS–11 to load files</td>
<td>2.7.20</td>
<td>If you need to use CAPS–11 to load RT–11 files, you must modify the cassette handler, CT.SYS, to alter header records.</td>
</tr>
<tr>
<td>Setting upper limit on memory size</td>
<td>2.7.21</td>
<td>If your PDP–11 does not generate a bus timeout trap, RT–11 does not load into memory properly. You can alter the monitor to cause the bootstrap to never look for more than 28K words of memory.</td>
</tr>
<tr>
<td>Suppressing bootstrap message</td>
<td>2.7.22</td>
<td>If you want to prevent the monitor identification message from printing when you bootstrap a monitor, you can modify that monitor.</td>
</tr>
<tr>
<td>Suppressing start-up indirect command file</td>
<td>2.7.23</td>
<td>If you want to prevent the start-up command file from executing when you bootstrap a monitor, you can modify that monitor.</td>
</tr>
<tr>
<td>Suppressing start-up indirect command file echo</td>
<td>2.7.24</td>
<td>If you want the start-up indirect command file to execute when you bootstrap a monitor but you do not want the command lines in the file to echo (appear) on the terminal, you can modify the monitor.</td>
</tr>
<tr>
<td>Changing bootstrap message</td>
<td>2.7.25</td>
<td>If you want to change the monitor identification message that appears when you bootstrap a monitor, you can modify that monitor.</td>
</tr>
<tr>
<td>Changing default device for indirect command files</td>
<td>2.7.26</td>
<td>If you want to change the default device for indirect command files, you can modify the monitor. Normally, the monitor looks for the command file on DK:.</td>
</tr>
<tr>
<td>Changing default file type for indirect command files</td>
<td>2.7.27</td>
<td>If you want to change the default file type for indirect command files, you can modify the monitor. Normally, indirect command files have the default file type .COM.</td>
</tr>
<tr>
<td>Changing default device for FRUN command</td>
<td>2.7.28</td>
<td>If you want to change the default device for the FRUN command, you can modify the monitor. Normally, the monitor looks for the foreground program on device DK: when you type FRUN filnam.</td>
</tr>
<tr>
<td>Changing default file type for FRUN command</td>
<td>2.7.29</td>
<td>If you want to change the default file type for the FRUN command, you can modify the monitor. Normally, the default file type for foreground programs is .REL.</td>
</tr>
<tr>
<td>Changing default device for EDIT command</td>
<td>2.7.30</td>
<td>If you want to change the default device for the EDIT command, you can modify the monitor. Normally, the monitor looks for EDIT.SAV on device SY:.</td>
</tr>
</tbody>
</table>

(continued on next page)
<table>
<thead>
<tr>
<th>Feature</th>
<th>Section</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changing default file name for EDIT command</td>
<td>2.7.31</td>
<td>If you want the monitor to run an editor other than the default editor when you type the EDIT command, you can modify the monitor to change the file name of the default editor.</td>
</tr>
<tr>
<td>Using Examine and Deposit above background job</td>
<td>2.7.32</td>
<td>If you want to be able to examine and modify the monitor and the I/O page, you can customize the monitor to remove a restriction on the use of the examine (E) and deposit (D) commands. Normally, the monitor allows you to examine and modify only locations inside the background job's area.</td>
</tr>
<tr>
<td>Changing default device for QUEMAN</td>
<td>2.7.33</td>
<td>If you want to send a file to a device other than LP: when you use the PRINT command in an FB system (with QUEUE running as a foreground job), you can modify the monitor to change the default. This modification also changes the default for the DELETE/ENTRY command.</td>
</tr>
<tr>
<td>Changing indirect command file nesting depth</td>
<td>2.7.34</td>
<td>If you want to change the depth to which you can nest indirect command files, you can modify the monitor. Normally, RT-11 allows you to nest indirect command files three deep.</td>
</tr>
<tr>
<td>Changing threshold for resuming output-stalled jobs</td>
<td>2.7.35</td>
<td>If you have a foreground job that produces much terminal output and stalls frequently (waiting for room in the terminal output buffer), you can modify the monitor to improve throughput.</td>
</tr>
<tr>
<td>Changing default number of directory segments</td>
<td>2.7.36</td>
<td>If you want DUP to use a different default number of directory segments when you initialize volumes, you can modify DUP. DUP normally uses the number of directory segments specified in a table in DUP.</td>
</tr>
<tr>
<td>Changing width of banner pages from 132 to 80 columns</td>
<td>2.7.37</td>
<td>If you want to change the width of banner page printouts from 132 to 80 columns, you can modify QUEUE.</td>
</tr>
<tr>
<td>Modifying listing page length in LINK</td>
<td>2.7.38</td>
<td>If your line printer uses paper that is not 10.5 inches long, or if you do not have a line printer, you can modify LINK to change the listing page length from the standard 60 lines.</td>
</tr>
<tr>
<td>Assigning help file</td>
<td>2.7.39</td>
<td>You can modify the monitor to assign your HELP file to a file or device other than SY:HELP.SAV.</td>
</tr>
</tbody>
</table>

(continued on next page)
<table>
<thead>
<tr>
<th>Feature</th>
<th>Section</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changing device from which IND is run</td>
<td>2.7.40</td>
<td>You can modify the monitor to change the device from which IND is run.</td>
</tr>
<tr>
<td>Replacing bad blocks</td>
<td>2.7.41</td>
<td>You must customize DUP if your user-written device handler supports bad block replacement. One change is required to replace all bad blocks, another to replace only bad sectors.</td>
</tr>
<tr>
<td>Including user-written magtape handler</td>
<td>2.7.42</td>
<td>You must customize DUP, PIP, and MDUP if you use a user-written magtape handler.</td>
</tr>
<tr>
<td>Producing 1600 bits/in bootable magtape</td>
<td>2.7.43</td>
<td>You can edit DISMT1.COM to replace module MBOOT.BOT with MBOT16.BOT, a primary bootstrap for producing 1600 bits/in, phase-encoded bootable magtape.</td>
</tr>
<tr>
<td>Changing default file type of logical disk files</td>
<td>2.7.44</td>
<td>You can change the default file type of logical disk files from .DSK by modifying either the LD.SYS file or the conditional file SYSGEN.CND.</td>
</tr>
<tr>
<td>Changing QUEUE to allow first form feed</td>
<td>2.7.45</td>
<td>You can modify QUEUE to never suppress the initial form feed in a file.</td>
</tr>
<tr>
<td>Changing listing page length in MACRO and CREF</td>
<td>2.7.46</td>
<td>You can change the default (60 lines) page listing length for MACRO and CREF.</td>
</tr>
<tr>
<td>Changing default command file processor to IND</td>
<td>2.7.47</td>
<td>You can change the monitor command file processor from KMON to IND.</td>
</tr>
<tr>
<td>Limiting amount of memory KEX requests</td>
<td>2.7.48</td>
<td>You can reduce memory requested by KEX from the default 32KW to your requirements by using the .SETTOP programmed request.</td>
</tr>
<tr>
<td>Changing MACRO's default .LIST/.NLIST options</td>
<td>2.7.49</td>
<td>You can change the default options for the .LIST/.NLIST macros.</td>
</tr>
</tbody>
</table>

(continued on next page)
Table 1–3: Features Available Through Simple Software Customizations (Cont.)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Section</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changing MACRO's default .ENABLE/.DISABLE options</td>
<td>2.7.50</td>
<td>You can change the default options for the .ENABLE/.DISABLE macros.</td>
</tr>
<tr>
<td>Modifying KED default file type</td>
<td>2.7.51</td>
<td>You can change the KED default file type.</td>
</tr>
<tr>
<td>Changing SPOOL's work file size</td>
<td>2.7.52</td>
<td>You can change the size of SPOOL's work file.</td>
</tr>
<tr>
<td>Changing SPOOL's output device</td>
<td>2.7.53</td>
<td>You can change SPOOL's default output device.</td>
</tr>
<tr>
<td>Changing LINK's default map and table output to 132 columns</td>
<td>2.7.54</td>
<td>You can change LINK's default link map and global cross-reference table output to 132 columns.</td>
</tr>
</tbody>
</table>

1.1.3.2 Do You Need to Perform the System Generation Procedures? — You must perform the system generation procedures if you want to add to your system any features listed in Table 1–4. You may also need to perform the system generation procedures if you want to remove features to reduce the size of the monitor and improve response time. Refer to the system generation dialog and the descriptive text in the *RT–11 System Generation Guide* for further information about these features.

You should not attempt to perform the system generation procedures unless your hardware configuration meets certain requirements. DIGITAL supports automatic system generation (under license) only on a system with at least 16K words of memory and 2000 free blocks of disk storage. The minimum configuration that DIGITAL recommends for system generation is a system with at least two disk drives and 24K words of memory. DIGITAL also supports system generation on dual-diskette systems (with at least 28K words of memory) if you use the manual method described in the *RT–11 System Generation Guide*. However, DIGITAL does not recommend this very lengthy procedure.
Table 1–4: Features Available Only Through the System Generation Procedures

<table>
<thead>
<tr>
<th>Feature</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asynchronous terminal status</td>
<td>Provides a program with the updated status of terminals in multiterminal systems. MU BASIC–11 requires this support.</td>
</tr>
<tr>
<td>BATCH support</td>
<td>Job control language allows RT–11 to operate unattended. All monitors except BL support BATCH.</td>
</tr>
<tr>
<td>Device I/O timeout</td>
<td>Permits device handlers to issue a mark-time programmed request. DECnet applications require this support, but RT–11 does not.</td>
</tr>
<tr>
<td>Disable .FETCH request under XM</td>
<td>Disables the use of the .FETCH request under XM, requiring device handlers to be resident in memory.</td>
</tr>
<tr>
<td>Double-density only RX02</td>
<td>Permits you to use only double-density RX02 diskettes on the system. This feature makes the RX02 handler smaller.</td>
</tr>
<tr>
<td>DZ11 up to 9600 baud Error logging</td>
<td>Permits you to initialize lines at specific baud rates up to 9600. Keeps a statistical record of device, memory parity, and memory cache errors. An error logging job is created when you select this support. The EL job retrieves information that is later available to you in summary report format. All monitors except BL support error logging.</td>
</tr>
<tr>
<td>Extra device slots</td>
<td>Permits you to add devices to the system after it is built. The number of logical assignments you can make is equal to the number of devices plus the empty device slots in the system.</td>
</tr>
<tr>
<td>Global SCCA</td>
<td>When global SCCA is turned on (enabled), all CTRL/C characters sent from the terminal to the background are ignored, protecting the program from being aborted with double CTRL/Cs. Support must be chosen to use IND directives .ENABLE ABORT and .DISABLE ABORT.</td>
</tr>
<tr>
<td>Graphics support</td>
<td>Enables the commands GT ON and GT OFF.</td>
</tr>
<tr>
<td>High-speed ring buffer</td>
<td>Causes character processing and interpretation to be performed at the fork level, allowing short bursts of characters transmitted at very high rates to be received. Use of this feature is recommended with PDTs. All monitors except BL support this feature.</td>
</tr>
<tr>
<td>Keyboard monitor command subsets</td>
<td>Allows you to choose one, two, or three subsets of the keyboard monitor commands instead of all the commands.</td>
</tr>
<tr>
<td>Memory parity</td>
<td>Causes the system to print an error message when a memory parity error occurs if your configuration includes memory parity hardware. If you have this hardware but do not select this support, the system halts when memory errors occur. All monitors except BL support this feature; it is available in the distributed XM monitor.</td>
</tr>
</tbody>
</table>

(continued on next page)
<table>
<thead>
<tr>
<th>Feature</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Month and year rollover</td>
<td>Adds support that automatically rolls over the date at the end of the month and the end of the year. Normally, you must reset the date and time. This support is useful for applications that run continuously and over a long period of time. All monitors except BL support this feature.</td>
</tr>
<tr>
<td>Multiterminal support</td>
<td>Permits you to use two or more terminals with the SJ, FB, or XM monitor. MU BASIC-11 requires this support.</td>
</tr>
<tr>
<td>Multiterminal timeout</td>
<td>Causes the monitor to reset at regular intervals any terminal that goes off line. This support minimizes the impact of static in multiterminal systems. MU BASIC-11 requires this support.</td>
</tr>
<tr>
<td>Programmable clock as system clock</td>
<td>Allows you to substitute as the system clock the KW11-P programmable clock for the usual line clock. However, the programmable clock would not then be available for program use. All monitors except BL support this feature.</td>
</tr>
<tr>
<td>Ring buffer size</td>
<td>Allows you to change the size of the input and output ring buffers. The input ring is a buffer in the monitor that holds characters you type at a terminal until a program requests them. The output ring is a buffer in the monitor that holds characters until the terminal can print them. The default input ring buffer size is 134 (decimal) characters, and the default output ring buffer size is 40 (decimal) characters.</td>
</tr>
<tr>
<td>Second RX01, RX02, or TU58 controller</td>
<td>Adds support for a second RX01, RX02, or TU58 controller, allowing a total of four units in the configuration instead of the usual two.</td>
</tr>
<tr>
<td>SJ message on system I/O errors</td>
<td>Causes the SJ monitor to issue an error message instead of simply halting. This feature helps to reduce confusion when an error occurs. The FB and XM monitors always issue error messages.</td>
</tr>
<tr>
<td>SJ timer</td>
<td>Configures the SJ monitor to support mark-time and cancel mark-time programmed requests. Otherwise, only the FB and XM monitors support these requests, which provide timer capabilities.</td>
</tr>
<tr>
<td>.SPCPS programmed request</td>
<td>The save/set main-line PC and PS (.SPCPS) programmed request changes the flow of control of main-line code by saving the PC and PS and changing the main-line PC to a new value. This support can be generated for only the FB and XM monitors. .SPCPS is especially useful to control the switching context among users in multiuser applications.</td>
</tr>
</tbody>
</table>
Table 1–4: Features Available Only Through the System Generation Procedures (Cont.)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>System jobs</td>
<td>Assembles the FB or XM monitor to support as many as eight simultaneously active jobs instead of the usual two. The error logging subsystem, the device queue program (QUEUE), the transparent spooler (SPOOL), and the communication package (VTCOM) can run as system jobs. This feature is available in the distributed XM monitor.</td>
</tr>
<tr>
<td>User command linkage</td>
<td>Permits users to create their own commands in a format similar to supplied DCL commands. All monitors except BL support this feature.</td>
</tr>
</tbody>
</table>

1.2 Choosing a Reading Path

The modular organization of this manual limits your reading to those chapters that answer your needs. Table 1–5 gives you the information you need to select these chapters. To use the table, find your distribution medium and your target system device (the device that you will want to bootstrap and that will contain the monitors and system components). Then find in the rightmost column the relevant chapters for your configuration. If you must perform a system generation, refer to the RT–11 System Generation Guide after you have installed your system.

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

NOTE

If you are unfamiliar with RT–11 software, you should probably use the distributed monitors uncustomized for a while. Once you have gained familiarity with the system, you will be better equipped to perform customizations or system generation procedures. Be aware, however, that some customizations may be essential for your particular application, and, consequently, you may experience problems when you use software without needed support. If you are an inexperienced user, be sure to study Tables 1–3 and 1–4, but you may want to skip Section 2.7 (Software Customizations) until you have gained some experience.
<table>
<thead>
<tr>
<th>Distribution Medium</th>
<th>System Device in Target System</th>
<th>Chapters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-Density Diskette RX01</td>
<td>Single-Density Diskette or DECTapeII</td>
<td>1, 2, 3</td>
</tr>
<tr>
<td>Hard Disk</td>
<td>Hard Disk</td>
<td>1, 2, 4</td>
</tr>
<tr>
<td>RL01</td>
<td>RL01</td>
<td>1, 2, 5</td>
</tr>
<tr>
<td>RL02</td>
<td>RL02</td>
<td></td>
</tr>
<tr>
<td>RK05</td>
<td>RK05</td>
<td></td>
</tr>
<tr>
<td>RC25</td>
<td>RK06</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RK07</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RC25</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Single-Density or Double-Density Diskette</td>
<td>1, 2, 6</td>
</tr>
<tr>
<td>Double-Density Diskette RX02</td>
<td>Double-Density Diskette</td>
<td>1, 2, 7</td>
</tr>
<tr>
<td></td>
<td>RX02</td>
<td></td>
</tr>
<tr>
<td>Magtape</td>
<td>Hard Disk</td>
<td>1, 2, 8</td>
</tr>
<tr>
<td>TM11</td>
<td>RL01</td>
<td></td>
</tr>
<tr>
<td>TJU16</td>
<td>RL02</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RK05</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RK06</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RK07</td>
<td></td>
</tr>
<tr>
<td>Diskette RX50</td>
<td>RD51 (MICRO/PDP-11)</td>
<td>1, 2, 9</td>
</tr>
<tr>
<td></td>
<td>RD51 (Professional 350)</td>
<td>1, 2, 11</td>
</tr>
<tr>
<td></td>
<td>Diskette</td>
<td>1, 2, 10</td>
</tr>
<tr>
<td></td>
<td>RX50</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 2
Preparing for Installation

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

1. Survey the general installation procedures.
2. Study the contents of your software kit.
3. Select the components you need in your working system.
4. Plan the arrangement of components.
5. Install any software updates.
6. Acquire sufficient media.
7. Choose the software customizations you need to make.

You can use the worksheet (Figure 2–2) at the end of this chapter to record the components you select, the arrangement of files you plan, and any other information you need in order to perform the installation procedures.

2.1 Surveying Installation Procedures

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

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

Familiarity with the contents of the software kit you received helps to make the installation procedures go smoothly. You should learn which files are included in the kit and what each file does.

Study Table 2–1 for a summary of the RT–11 components distributed in the kits, and refer to the *RT–11 System Utilities Manual* for a thorough description of each utility. RT–11 is distributed on RC25, RL01, RL02, and RK05 disks, on RX01, RX02, and RX50 diskettes, and on magtape. The organization of files on volumes depends on the distribution medium.

Table 2–1: RT–11 Software Components

<table>
<thead>
<tr>
<th>Type of Software</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitors</td>
<td></td>
</tr>
<tr>
<td>RT11AL.SYS</td>
<td>Automatic installation monitor</td>
</tr>
<tr>
<td>RT11BL.SYS</td>
<td>Base-line single-job monitor</td>
</tr>
<tr>
<td>RT11SJ.SYS</td>
<td>Single-job monitor</td>
</tr>
<tr>
<td>RT11FB.SYS</td>
<td>Foreground/Background monitor</td>
</tr>
<tr>
<td>RT11XM.SYS</td>
<td>Extended memory monitor</td>
</tr>
<tr>
<td>RT11PL.SYS</td>
<td>Automatic installation monitor for Professional 300 series</td>
</tr>
<tr>
<td>Device Handlers</td>
<td></td>
</tr>
<tr>
<td>CR.SYS</td>
<td>Card reader handler for SJ, FB monitors</td>
</tr>
<tr>
<td>CT.SYS</td>
<td>TA11 cassette handler for SJ, FB monitors</td>
</tr>
<tr>
<td>DD.SYS</td>
<td>DECtape II handler for SJ, FB monitors</td>
</tr>
<tr>
<td>DDX.SYS</td>
<td>DECtape II handler for XM monitor</td>
</tr>
<tr>
<td>DL.SYS</td>
<td>RL11/RL01/RL02 handler for SJ, FB monitors</td>
</tr>
<tr>
<td>DLX.SYS</td>
<td>RL11/RL01/RL02 handler for XM monitor</td>
</tr>
<tr>
<td>DM.SYS</td>
<td>RK611/RK06/RK07 handler for SJ, FB monitors</td>
</tr>
<tr>
<td>DMX.SYS</td>
<td>RK611/RK06/RK07 handler for XM monitor</td>
</tr>
<tr>
<td>DP.SYS</td>
<td>RP11/RP02/RP03 handler for SJ, FB monitors</td>
</tr>
<tr>
<td>DS.SYS</td>
<td>RJS03/04 handler for SJ, FB monitors</td>
</tr>
<tr>
<td>DT.SYS</td>
<td>DECtape handler for SJ, FB monitors</td>
</tr>
<tr>
<td>DU.SYS</td>
<td>MSCP device handler for SJ, FB monitors</td>
</tr>
<tr>
<td>DUX.SYS</td>
<td>MSCP device handler for XM monitor</td>
</tr>
<tr>
<td>DW.SYS</td>
<td>Winchester disk handler for FB monitor</td>
</tr>
<tr>
<td>DWX.SYS</td>
<td>Winchester disk handler for XM monitor</td>
</tr>
<tr>
<td>DX.SYS</td>
<td>RX11/RX01 single-density diskette handler for SJ, FB monitors</td>
</tr>
</tbody>
</table>

(continued on next page)
<table>
<thead>
<tr>
<th>Type of Software</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXX.SYS</td>
<td>RX11/RX01 single-density diskette handler for XM monitor</td>
</tr>
<tr>
<td>DY.SYS</td>
<td>RX211/RX02 handler for SJ, FB monitors</td>
</tr>
<tr>
<td>DYS.SYS</td>
<td>RX211/RX02 handler for XM monitor</td>
</tr>
<tr>
<td>DZ.SYS</td>
<td>Diskette handler for FB monitor</td>
</tr>
<tr>
<td>DZX.SYS</td>
<td>Diskette handler for XM monitor</td>
</tr>
<tr>
<td>LD.SYS</td>
<td>Logical disk subsetting handler for SJ, FB monitors</td>
</tr>
<tr>
<td>LDX.SYS</td>
<td>Logical disk subsetting handler for XM monitor</td>
</tr>
<tr>
<td>LP.SYS</td>
<td>Line printer handler for SJ, FB monitors</td>
</tr>
<tr>
<td>LPX.SYS</td>
<td>Line printer handler for XM monitor</td>
</tr>
<tr>
<td>LS.SYS</td>
<td>Serial printer handler for SJ, FB monitors</td>
</tr>
<tr>
<td>LSX.SYS</td>
<td>Serial printer handler for XM monitor</td>
</tr>
<tr>
<td>MM.SYS</td>
<td>File-structured TJU16 handler for SJ, FB monitors</td>
</tr>
<tr>
<td>MMX.SYS</td>
<td>File-structured TJU16 handler for XM monitor</td>
</tr>
<tr>
<td>MS.SYS</td>
<td>File-structured TS11 handler for SJ, FB monitors</td>
</tr>
<tr>
<td>MSX.SYS</td>
<td>File-structured TS11 handler for XM monitor</td>
</tr>
<tr>
<td>MTL.SYS</td>
<td>File-structured TM11 handler for SJ, FB monitors</td>
</tr>
<tr>
<td>MTX.SYS</td>
<td>File-structured TM11 handler for XM monitor</td>
</tr>
<tr>
<td>NL.SYS</td>
<td>Null handler for SJ, FB monitors</td>
</tr>
<tr>
<td>NLX.SYS</td>
<td>Null handler for XM monitor</td>
</tr>
<tr>
<td>PC.SYS</td>
<td>High-speed paper tape handler for SJ, FB monitors</td>
</tr>
<tr>
<td>PD.SYS</td>
<td>PDT-11 skeleton handler for SJ, FB monitors</td>
</tr>
<tr>
<td>PI.SYS</td>
<td>Professional interface handler for video, clock and keyboard for FB monitor</td>
</tr>
<tr>
<td>PIX.SYS</td>
<td>Professional interface handler for video, clock and keyboard for XM monitor</td>
</tr>
<tr>
<td>RF.SYS</td>
<td>RF11 handler for SJ, FB monitors</td>
</tr>
<tr>
<td>RK.SYS</td>
<td>RK11/RK05 handler for SJ, FB monitors</td>
</tr>
<tr>
<td>RKX.SYS</td>
<td>RK11/RK05 handler for XM monitor</td>
</tr>
<tr>
<td>SP.SYS</td>
<td>Transparent print spooler handler for FB monitor</td>
</tr>
<tr>
<td>SPX.SYS</td>
<td>Transparent print spooler handler for XM monitor</td>
</tr>
<tr>
<td>TT.SYS</td>
<td>Terminal handler for SJ monitor</td>
</tr>
<tr>
<td>VM.SYS</td>
<td>Virtual disk handler for SJ, FB monitors</td>
</tr>
<tr>
<td>VMX.SYS</td>
<td>Virtual disk handler for XM monitor</td>
</tr>
</tbody>
</table>

(continued on next page)
<table>
<thead>
<tr>
<th>Type of Software</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XC.SYS</td>
<td>Professional communication port modem handler for FB monitor</td>
</tr>
<tr>
<td>XCX.SYS</td>
<td>Professional communication port modem handler for XM monitor</td>
</tr>
<tr>
<td>XL.SYS</td>
<td>DL–11 communication port modem handler for FB monitor</td>
</tr>
<tr>
<td>XLX.SYS</td>
<td>DL–11 communication port modem handler for XM monitor</td>
</tr>
</tbody>
</table>

**Other System Files**

<table>
<thead>
<tr>
<th>Type of Software</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL.SYS</td>
<td>Single-line editor pseudohandler for SJ, FB monitors</td>
</tr>
<tr>
<td>SLX.SYS</td>
<td>Single-line editor pseudohandler for XM monitor</td>
</tr>
<tr>
<td>SWAP.SYS</td>
<td>External monitor swap blocks</td>
</tr>
</tbody>
</table>

**Utility Programs**

<table>
<thead>
<tr>
<th>Type of Software</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BINCOM.SAV</td>
<td>Binary compare utility</td>
</tr>
<tr>
<td>BUP.SAV</td>
<td>Backup utility</td>
</tr>
<tr>
<td>CREF.SAV</td>
<td>Cross-reference utility</td>
</tr>
<tr>
<td>DIR.SAV</td>
<td>Directory utility</td>
</tr>
<tr>
<td>DUMP.SAV</td>
<td>File dump utility</td>
</tr>
<tr>
<td>DUP.SAV</td>
<td>Disk maintenance utility</td>
</tr>
<tr>
<td>EDIT.SAV</td>
<td>Text editor</td>
</tr>
<tr>
<td>FILEX.SAV</td>
<td>Foreign file exchange utility</td>
</tr>
<tr>
<td>FORMAT.SAV</td>
<td>Disk formatting utility</td>
</tr>
<tr>
<td>HELP.SAV</td>
<td>Help utility</td>
</tr>
<tr>
<td>IND.SAV</td>
<td>Indirect control file processor</td>
</tr>
<tr>
<td>KED.SAV</td>
<td>Keypad editor for VT100 terminal</td>
</tr>
<tr>
<td>K52.SAV</td>
<td>Keypad editor for VT52 terminal</td>
</tr>
<tr>
<td>LIBR.SAV</td>
<td>Librarian</td>
</tr>
<tr>
<td>LINK.SAV</td>
<td>Linker</td>
</tr>
<tr>
<td>MACRO.SAV</td>
<td>MACRO assembler</td>
</tr>
<tr>
<td>MDUP.MM</td>
<td>Magtape bootstrap utility for TJU16</td>
</tr>
<tr>
<td>MDUP.MS</td>
<td>Magtape bootstrap utility for TS11</td>
</tr>
<tr>
<td>MDUP.MT</td>
<td>Magtape bootstrap utility for TM11</td>
</tr>
<tr>
<td>MDUP.SAV</td>
<td>Magtape utility</td>
</tr>
<tr>
<td>PAT.SAV</td>
<td>Patching utility for program object modules</td>
</tr>
<tr>
<td>PIP.SAV</td>
<td>File transfer utility</td>
</tr>
</tbody>
</table>

(continued on next page)
<table>
<thead>
<tr>
<th>Type of Software</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUEMAN.SAV</td>
<td>User interface with QUEUE utility</td>
</tr>
<tr>
<td>QUEUE.REL</td>
<td>Device queue utility</td>
</tr>
<tr>
<td>RESORC.SAV</td>
<td>System resource display utility</td>
</tr>
<tr>
<td>SETUP.SAV</td>
<td>VT100 and Professional terminal setup utility</td>
</tr>
<tr>
<td>SIPP.SAV</td>
<td>Save image patch utility</td>
</tr>
<tr>
<td>SLP.SAV</td>
<td>Source language patch utility</td>
</tr>
<tr>
<td>SPOOL.REL</td>
<td>Transparent print spooler utility</td>
</tr>
<tr>
<td>SRCCOM.SAV</td>
<td>Source compare utility</td>
</tr>
<tr>
<td>SYSMAC.SML</td>
<td>System macro library</td>
</tr>
<tr>
<td>TRANSF.SAV</td>
<td>File transfer program for VTCOM, to be installed on RT-11 or RTEM-11 host system</td>
</tr>
<tr>
<td>UCL.SAV</td>
<td>User command linkage executable image</td>
</tr>
<tr>
<td>VTCOM.REL</td>
<td>Virtual terminal communication and file transfer utility</td>
</tr>
<tr>
<td>VTCOM.SAV</td>
<td>Virtual version of VTCOM communication utility, for running under XM monitor</td>
</tr>
</tbody>
</table>

**Miscellaneous**

<table>
<thead>
<tr>
<th>Type of Software</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATCH.SAV</td>
<td>BATCH processor</td>
</tr>
<tr>
<td>DISMT1.COM</td>
<td>Copy and write bootstrap to distribution magtape 1</td>
</tr>
<tr>
<td>DISMT2.COM</td>
<td>Copy distribution magtape 2</td>
</tr>
<tr>
<td>ERROUT.SAV</td>
<td>Error log program</td>
</tr>
<tr>
<td>RTBL.MAP</td>
<td>Distributed BL monitor link map</td>
</tr>
<tr>
<td>RTSJ.MAP</td>
<td>Distributed SJ monitor link map</td>
</tr>
<tr>
<td>RTFB.MAP</td>
<td>Distributed FB monitor link map</td>
</tr>
<tr>
<td>RTXM.MAP</td>
<td>Distributed XM monitor link map</td>
</tr>
<tr>
<td>STARTF.COM</td>
<td>FB start-up command file</td>
</tr>
<tr>
<td>STARTS.COM</td>
<td>SJ start-up command file</td>
</tr>
<tr>
<td>STARTX.COM</td>
<td>XM start-up command file</td>
</tr>
<tr>
<td>V5USER.TXT</td>
<td>Distribution kit message text file</td>
</tr>
</tbody>
</table>

**System Generation Files**

<table>
<thead>
<tr>
<th>Type of Software</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSGEN.COM</td>
<td>SYSGEN command file</td>
</tr>
<tr>
<td>BL.ANS</td>
<td>Answer file to build distributed BL monitor</td>
</tr>
<tr>
<td>SJFB.ANS</td>
<td>Answer file to build distributed SJ, FB monitors</td>
</tr>
<tr>
<td>XM.ANS</td>
<td>Answer file to build distributed XM monitor</td>
</tr>
</tbody>
</table>

(continued on next page)
Table 2-1: RT-11 Software Components (Cont.)

<table>
<thead>
<tr>
<th>Type of Software</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic Installation Files</td>
<td></td>
</tr>
<tr>
<td>IVP.MAC</td>
<td>Automatic installation verification source file</td>
</tr>
<tr>
<td>STARTA.COM</td>
<td>Start-up file for automatic installation</td>
</tr>
<tr>
<td>TERMID.SAV</td>
<td>Console terminal identification program</td>
</tr>
<tr>
<td>VERIFY.COM</td>
<td>Automatic installation verification command file</td>
</tr>
<tr>
<td>Debuggers</td>
<td></td>
</tr>
<tr>
<td>ODT.OBJ</td>
<td>Debugging aid</td>
</tr>
<tr>
<td>VDT.OBJ</td>
<td>Debugging aid for virtual and multiterminal jobs</td>
</tr>
<tr>
<td>Graphics Software</td>
<td></td>
</tr>
<tr>
<td>VTHDLR.OBJ</td>
<td>VT11/VS60 display handler</td>
</tr>
<tr>
<td>VTMAC.MAC</td>
<td>Display handler macro file</td>
</tr>
<tr>
<td>Libraries and Subroutines</td>
<td></td>
</tr>
<tr>
<td>GETSTR.FOR</td>
<td>FORTRAN subroutine source</td>
</tr>
<tr>
<td>PUTSTR.FOR</td>
<td>FORTRAN subroutine source</td>
</tr>
<tr>
<td>SYSLIB.OBJ</td>
<td>System FORTRAN-callable subroutines</td>
</tr>
<tr>
<td>Bootstraps</td>
<td></td>
</tr>
<tr>
<td>MBOOT.BOT</td>
<td>Magtape primary bootstrap for 8-track tape</td>
</tr>
<tr>
<td>MBOT16.BOT</td>
<td>Magtape primary bootstrap for 16-track tape</td>
</tr>
<tr>
<td>MSBOOT.BOT</td>
<td>Magtape secondary bootstrap</td>
</tr>
<tr>
<td>Demonstration Programs</td>
<td></td>
</tr>
<tr>
<td>DEMOBOX.MAC</td>
<td>Demonstration source — SJ, FB macro program</td>
</tr>
<tr>
<td>DEMOED.TXT</td>
<td>Demonstration source</td>
</tr>
<tr>
<td>DEMOFG.MAC</td>
<td>Demonstration source — FB macro program</td>
</tr>
<tr>
<td>DEMOF1.FOR</td>
<td>Demonstration source — FB FORTRAN program</td>
</tr>
<tr>
<td>DEMOX1.MAC</td>
<td>Demonstration source</td>
</tr>
<tr>
<td>Source Files</td>
<td></td>
</tr>
<tr>
<td>BA.MAC</td>
<td>BATCH handler source file for system generation</td>
</tr>
<tr>
<td>BSTRAP.MAC</td>
<td>Bootstrap source file for system generation</td>
</tr>
<tr>
<td>CR.MAC</td>
<td>Card reader handler source file for system generation</td>
</tr>
<tr>
<td>CT.MAC</td>
<td>Cassette handler source file for system generation</td>
</tr>
<tr>
<td>DD.MAC</td>
<td>DECTape II handler source file for system generation</td>
</tr>
<tr>
<td>DL.MAC</td>
<td>RL01/02 handler source file for system generation</td>
</tr>
<tr>
<td>DM.MAC</td>
<td>RK06/07 handler source file for system generation</td>
</tr>
</tbody>
</table>

(continued on next page)
<table>
<thead>
<tr>
<th>Type of Software</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP.MAC</td>
<td>RP11 handler source file for system generation</td>
</tr>
<tr>
<td>DS.MAC</td>
<td>RJS03/04 handler source file for system generation</td>
</tr>
<tr>
<td>DT.MAC</td>
<td>DECTape handler source file for system generation</td>
</tr>
<tr>
<td>DU.MAC</td>
<td>MSCP device handler source file for system generation</td>
</tr>
<tr>
<td>DW.MAC</td>
<td>Winchester disk handler source file for system generation</td>
</tr>
<tr>
<td>DX.MAC</td>
<td>RX01 handler source file for system generation</td>
</tr>
<tr>
<td>DY.MAC</td>
<td>RX02 handler source file for system generation</td>
</tr>
<tr>
<td>DZ.MAC</td>
<td>Diskette handler source file for system generation</td>
</tr>
<tr>
<td>EDTGBL.MAC</td>
<td>Monitor edit log and global definition file for system generation</td>
</tr>
<tr>
<td>EL.MAC</td>
<td>Error logger handler source file for system generation</td>
</tr>
<tr>
<td>ELCOPY.MAC</td>
<td>Error log job source file for system generation</td>
</tr>
<tr>
<td>ELINIT.MAC</td>
<td>Error log job source file for system generation</td>
</tr>
<tr>
<td>ELTASK.MAC</td>
<td>Error log job source file for system generation</td>
</tr>
<tr>
<td>ERROUT.OBJ</td>
<td>Error log job source file for system generation</td>
</tr>
<tr>
<td>ERRTXT.MAC</td>
<td>Error log job source file for system generation</td>
</tr>
<tr>
<td>FB.MAC</td>
<td>FB conditional source file for system generation</td>
</tr>
<tr>
<td>FSM.MAC</td>
<td>Magtape file support source file for system generation</td>
</tr>
<tr>
<td>KMON.MAC</td>
<td>Keyboard monitor source file for system generation</td>
</tr>
<tr>
<td>KMOVLY.MAC</td>
<td>Keyboard monitor overlay source file for system generation</td>
</tr>
<tr>
<td>LD.MAC</td>
<td>Logical disk subsetting handler source file for system generation</td>
</tr>
<tr>
<td>LP.MAC</td>
<td>Line printer handler source file for system generation</td>
</tr>
<tr>
<td>LS.MAC</td>
<td>Serial printer handler source file for system generation</td>
</tr>
<tr>
<td>MTTEMT.MAC</td>
<td>Multiterminal programmed request source file for system generation</td>
</tr>
<tr>
<td>MTTINT.MAC</td>
<td>Multiterminal interrupt service source file for system generation</td>
</tr>
<tr>
<td>NL.MAC</td>
<td>Null handler source file for system generation</td>
</tr>
<tr>
<td>PC.MAC</td>
<td>High-speed paper tape handler source file for system generation</td>
</tr>
<tr>
<td>PD.MAC</td>
<td>PDT-11 handler source file for system generation</td>
</tr>
<tr>
<td>RF.MAC</td>
<td>RF11 handler source file for system generation</td>
</tr>
<tr>
<td>RK.MAC</td>
<td>RK05 handler source file for system generation</td>
</tr>
</tbody>
</table>

(continued on next page)
### Table 2–1: RT–11 Software Components (Cont.)

<table>
<thead>
<tr>
<th>Type of Software</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMONFB.MAC</td>
<td>FB/XM resident monitor source file for system generation</td>
</tr>
<tr>
<td>RMONSJ.MAC</td>
<td>SJ resident monitor source file for system generation</td>
</tr>
<tr>
<td>SJ.MAC</td>
<td>SJ conditional source file for system generation</td>
</tr>
<tr>
<td>SP.MAC</td>
<td>Transparent print spooler handler source file for system generation</td>
</tr>
<tr>
<td>TJ.MAC</td>
<td>TJU16 handler source file for system generation</td>
</tr>
<tr>
<td>TM.MAC</td>
<td>TM11 handler source file for system generation</td>
</tr>
<tr>
<td>TRMTBL.MAC</td>
<td>Multiterminal table source file for system generation</td>
</tr>
<tr>
<td>TS.MAC</td>
<td>TS11 handler source file for system generation</td>
</tr>
<tr>
<td>TT.MAC</td>
<td>TT.SYS source file for system generation</td>
</tr>
<tr>
<td>USR.MAC</td>
<td>USR source file for system generation</td>
</tr>
<tr>
<td>VM.MAC</td>
<td>VM.SYS source file for system generation</td>
</tr>
<tr>
<td>XC.MAC</td>
<td>Professional communication port modem handler source file for system generation</td>
</tr>
<tr>
<td>XL.MAC</td>
<td>DL–11 communication port modem handler source file for system generation</td>
</tr>
<tr>
<td>XM.MAC</td>
<td>XM conditional source file for system generation</td>
</tr>
<tr>
<td>XMSUBS.MAC</td>
<td>XM monitor subroutines for system generation</td>
</tr>
</tbody>
</table>

**Unsupported Software**

| Automatic system software configuration command file |
| Automatic system software configuration services program |
| Change boot-time console terminal |
| Customization symbol location file |
| Date and time source file for system generation |
| Date and time utility |
| System activity monitor |
| File split utility |
| RT–11 version 5 release notes |
| Virtual RUN utility |
| VT11 macro library |

#### 2.3 Selecting Components for Your Working System

Because space on volumes is limited, you should include in your system only those components that are essential to your application. To decide which components these should be, study the space limitations of your par-
ticular device, then consider the suggestions given for each software component (Sections 2.3.1 through 2.3.14).

Before selecting components, you should become familiar with your medium's space limitations. (See Table 2-2.) Establish how many blocks are occupied by the components residing on each volume and how much free space is available on the volume.

Table 2-2: Device Size

<table>
<thead>
<tr>
<th>Device</th>
<th>Device Name</th>
<th>Size in Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>RX01 Diskette</td>
<td>DX</td>
<td>486</td>
</tr>
<tr>
<td>RX02 Diskette</td>
<td>DY</td>
<td>974</td>
</tr>
<tr>
<td>RK05 Disk</td>
<td>RK</td>
<td>4.8K</td>
</tr>
<tr>
<td>RL01/02 Disk</td>
<td>DL</td>
<td>10.2/20.4K</td>
</tr>
<tr>
<td>RK06/07 Disk</td>
<td>DM</td>
<td>27.1/53.7K</td>
</tr>
<tr>
<td>RX50 Diskette</td>
<td>DU,DZ</td>
<td>786</td>
</tr>
<tr>
<td>RD50 Disk</td>
<td>DU,DW</td>
<td>9.7K</td>
</tr>
<tr>
<td>RD51 Disk</td>
<td>DU,DW</td>
<td>19.6K</td>
</tr>
<tr>
<td>RC25 Disk</td>
<td>DU</td>
<td>43K (approximate)</td>
</tr>
<tr>
<td>RA80 Disk</td>
<td>DU</td>
<td>220K (approximate)</td>
</tr>
<tr>
<td>Virtual Memory</td>
<td>VM</td>
<td>Variable, depending on physical memory available</td>
</tr>
</tbody>
</table>

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

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

```
17-SEP-82
SWAP .SYS 26 26-AUG-82 RT11FB.SYS 86 26-AUG-82
DIR .SAV 18 26-AUG-82 RK .SYS 3 26-AUG-82
DX .SYS 3 26-AUG-82 DL .SYS 4 26-AUG-82
PIP .SAV 29 26-AUG-82 DUP .SAV 43 26-AUG-82
FORMAT .SAV 20 26-AUG-82 RESORC.SAV 20 26-AUG-82
RT11FB.SYS 86 26-AUG-82
10 FILES, 239 BLOCKS
735 FREE BLOCKS
```
You can also find the total free space on an initialized blank volume by obtaining a directory of the volume.

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

1. Monitor
2. SWAP.SYS File
3. System Device Handler
4. Other Device Handlers
5. Default System Library
6. Exercises
7. Help Package
8. Line Printer Handler
9. MACRO Assembler
10. Queue Package
11. Source Files
12. Start-up Command File
13. Text Editors
14. Utility Programs

As you make selections, record your choices on the worksheet at the end of the chapter (Figure 2–2).

2.3.1 Monitor

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

2.3.2 SWAP.SYS File

You need the file SWAP.SYS on a system volume to serve as temporary storage for part of a program in memory when KMON, the USR, or both must swap over that program. When KMON or the USR are no longer needed, the system reads this external swap file back into main memory. The keyboard command, SET EXIT NOSWAP, precludes the necessity of using SWAP.SYS once you have bootstrapped the system. However, you must have the SWAP.SYS file on your distribution disk in order to boot-strap your system successfully.
2.3.3 System Device Handler

You need a system device handler on each system volume. For example, if you build an FB system with RL02 as the system device, the file DL.SYS must be on the system disk.

2.3.4 Other Device Handlers

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

2.3.5 Default System Library

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

Generally, SYSLIB for your application should contain the system subroutines (the file SYSLIB.OBJ found in the software kit), installation-specific libraries of application subroutines, and the FORTRAN OTS routines. If SYSLIB must contain application subroutines and language routines, you must customize it to include these routines. If you intend to link overlaid files, you need SYSLIB, because it contains the overlay handlers. If you are a MACRO-only user, requiring only the overlay handlers in SYSLIB (and not the other routines), you can create a separate library containing only the overlay handlers by using the EXTRACT option with the LIBRARY command. You can name the new, smaller library SYSLIB. Section 2.7.10 describes the procedure for creating such a library. Section 2.7.10 also describes adding the overlay handlers to another library (for example, FORLIB or the DIBOL library) when the other library already exists.

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

```
.LIBRARY/INSERT,REMOVE SYSLIB.OBJ xxxxxxx.OBJ
Global? $OV RH
Global? $ERRS
Global? $ERRTE
Global? $ERR
```

Preparing for Installation  2–11
Note that you must also remove the globals $ERRS and $ERRTB from the library directory if you have included the FORTRAN library in SYSLIB.


2.3.6 Exercises

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

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

To do FORTRAN IV exercises in Introduction to RT-11, you also need:

FORTRA.SAV
SYSLIB.OBJ (with FORLIB.OBJ included in it if possible)
FORLIB.OBJ (if not included in SYSLIB.OBJ)
DEMOF1.FOR

To do BASIC-11 exercises in Introduction to RT-11, you also need:

BASIC.SAV

If the system volume is an RX01, RX02, or RX50, you will not have room for all components on the system volume. See Section 2.4.5 for the file arrangement you should use.

2.3.7 Help Package

HELP.SAV consists of HELP.EXE and HELP.MLB (resulting file after the librarian, LIBR, processes HELP.TXT) merged into a ready-to-use utility.
Unless you want to change the help text, HELP.SAV is the only file you need.

Section 2.7.14 describes how to customize the help text to your specific needs.

### 2.3.8 Line Printer Handlers

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

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

```
.RENAME LS.SYSLP.SYS
,BOOT SET
```

You must reboot the system after renaming the handler so the proper handler will be installed in the device table. Then, when you use the PRINT command, the system sends output to the serial printer. Since you will store both the distribution medium and the backup volume(s) you make during the installation process, you will always have copies of both handlers.

If your serial printer is installed at nonstandard vector and control status register addresses, you can use the SET command to change the addresses. (See Section 2.7.11.1.) Note, however, that once you have renamed LS.SYS to LP.SYS, you must use the device name LP: in a SET command. The available SET options would still be those for the LS handler.

### 2.3.9 MACRO–11 Assembler

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

### 2.3.10 Line Printer Utilities

RT–11 contains two utilities, QUEUE and SPOOL, that are used to send output to the printer. Both utilities run with the FB and XM monitors only, and can be used as foreground or system jobs. You must perform the system generation process to generate support for system jobs under the FB monitor.

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

To use SPOOL, you need the files SPOOL.REL (which gathers the output,
stores it in a work file, and sends it to the printer), and SP.SYS (FB monitor
SPOOL handler) or SPX.SYS (XM monitor SPOOL handler).

There are significant functional differences in QUEUE as opposed to
SPOOL. Consult the RT-11 System Utilities Manual for a complete descrip-
tion of each utility. For information on system job support, consult the
RT-11 System Generation Guide. Section 2.7 in this manual contains soft-
ware customizations for these utilities.

2.3.11 Source Files

Normally, you do not need any source files in your working system except
demonstration sources. Otherwise, the source files (file type .MAC) are re-
quired only for the system generation process.

You need the demonstration sources in the working system to perform the
exercises in Chapters 3 through 11 of this manual and in Introduction to
RT-11. Once you have finished these exercises, the demonstration source
files are no longer useful and can be deleted from your working disk.

2.3.12 Start-Up Command File

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

2.3.13 Text Editors

RT-11 gives you a choice of three text editors. EDIT is a character-oriented
text editor that you can use with either a hard-copy terminal or a video
terminal. The Keypad Editor (KED or K52) is an easy-to-use, cursor-ori-
ented editor that you can use only on a VT100-compatible or VT52 video
terminal.

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

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

If you choose the keypad editor, the file you need depends on the type of
terminal you have. You need the file KED.SAV if your terminal is a
VT1xx; you need K52.SAV if your terminal is a VT52.

2–14 Preparing for Installation
2.3.14 Utility Programs

You do not need any system utility programs you do not intend to use, but remember that most of the keyboard monitor commands need certain system programs in order to work. For example, PIP, DUP, and DIR are necessary for most of the keyboard commands to function. Appendix A of the RT–11 System User’s Guide summarizes the keyboard monitor commands and lists the system utility programs that each command requires.

Refer to the RT–11 System Utilities Manual for descriptions of the functions of the utility programs. If your system device is a large disk, you will probably find it useful to have most of the utility programs on your working system device. On the other hand, if you have a small system device, you will not have room for all the utilities.

In addition to PIP, DUP, and DIR, you probably need to use an editor, LINK, FORMAT (for certain devices), and HELP frequently, and you may want them on the system volume. However, the utilities you use less frequently can occupy a second volume, which you can insert in and run from Unit 1.

2.4 Planning the Arrangement of Components

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

1. Assign the default device to the data device.
2. Create a separate utilities volume.
3. Create bad blocks on TU58 DECtape II cartridges to avoid excessive rewinds.
4. Create several system volumes.
5. Create volumes for use with Introduction to RT–11.

More details on each of these suggestions follow. (You can use the worksheet at the end of this chapter, Figure 2–2, to record the arrangement as you plan it.)

2.4.1 Assigning the Default Device to the Data Device

Once you have installed the system and it is running, you can assign the default device DK: to the data device (Unit 1). Most temporary files, data files, and others default to the second unit, minimizing demand for system device capacity.
The command to assign the default device to Unit 1 is as follows:

\texttt{* ASSIGN \texttt{xx1: \texttt{DK:} \texttt{RE}}}

You can include this command in your start-up command file to assign the default device to \texttt{xx1}: whenever you boot the system. The procedures in this manual assume \texttt{DK:} is the system device unless indicated otherwise. (See the \textit{RT–II System User's Guide}.)

Be sure to make any adjustments in your procedures if you assign the default to the data device.

### 2.4.2 Creating a Separate Utilities Volume

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

**NOTE**

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

To run a non-overlaid utility from the utilities volume, use the following commands, where \texttt{xx} is the physical device name, and \texttt{aaaaaa} is the utility program’s name.

\texttt{* RUN \texttt{xx1:\texttt{aaaaaa}}\texttt{RE}}

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

\texttt{*CTRL-C}

However, if you run an overlaid utility from Unit 1, the volume containing that utility must remain in Unit 1 at all times. Therefore, you should generally include the overlaid utilities on your system volume. The overlaid components are PIP, DUP, MACRO, LINK, LIBR, KED, K52, FORMAT, HELP, IND, BUP, SPOOL, and VTCOM.

ODT.OBJ is also useful on the system volume to debug MACRO–11 programs.
An example of this kind of arrangement of volumes into a system volume and a utilities volume follows:

<table>
<thead>
<tr>
<th>System Volume</th>
<th>Utilities Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWAP.SYS</td>
<td>FILEX.SAV</td>
</tr>
<tr>
<td>RT11FB.SYS</td>
<td>SIPP.SAV</td>
</tr>
<tr>
<td>DX.SYS</td>
<td>BINCOM.SAV</td>
</tr>
<tr>
<td>LP.SYS</td>
<td>SRCCOM.SAV</td>
</tr>
<tr>
<td>IND.SAV</td>
<td>DUMP.SAV</td>
</tr>
<tr>
<td>PIP.SAV</td>
<td>RESORC.SAV</td>
</tr>
<tr>
<td>DUP.SAV</td>
<td>PAT.SAV</td>
</tr>
<tr>
<td>DIR.SAV</td>
<td>FORMAT.SAV</td>
</tr>
<tr>
<td>KED.SAV</td>
<td>BUP.SAV</td>
</tr>
<tr>
<td>MACRO.SAV</td>
<td></td>
</tr>
<tr>
<td>LINK.SAV</td>
<td></td>
</tr>
<tr>
<td>LIBR.SAV</td>
<td></td>
</tr>
<tr>
<td>ODT.OBJ</td>
<td></td>
</tr>
<tr>
<td>HELP.SAV</td>
<td>— required for the HELP command</td>
</tr>
</tbody>
</table>

### 2.4.3 Creating Bad Blocks on TU58 DECTape II Cartridges

If your volumes are TU58 DECTape II cartridges, you may encounter slow response time due to excessive rewinds of the tape. Because of the way DECTape II stores data in records, you can actually improve system performance by creating dummy bad blocks in strategic locations. DECTape II writes data records in a specific sequence and pattern; to write an entire cartridge, for example, it performs the following actions.

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

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

You can create dummy bad blocks in three locations:

- Block 128 (decimal), at the beginning of the second pass on track 1
- Block 256 (decimal), at the beginning of the first pass on track 2
- Block 384 (decimal), at the beginning of the second pass on track 2
This arrangement prevents the system from writing across rewinds, since RT-11 requires contiguous free space in which to write files. However, this technique prevents you from creating any file over 127 (decimal) blocks long and also increases fragmentation. Figure 2-1 illustrates the location of blocks on tape.

Figure 2-1: Block Locations on DECtape II

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

```
,CREATE/START:128, xx1:FIL1.BAD
,CREATE/START:256, xx1:FIL2.BAD
,CREATE/START:384, xx1:FIL3.BAD
```

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

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

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

2.4.5 Creating Volumes for Use with Introduction to RT-11
If you intend to perform the exercises in Introduction to RT-11, you need certain components on your working system. If your system device is one of the small devices (RX01, RX02 or RX50), you need to build four volumes, which are listed below with the files they contain. Be sure to copy the bootstrap to each volume.
System Volume

SWAP.SYS
RT11FB.SYS
PI.SYS (if running on Professional 325)
xx.SYS (system device handler)
LP.SYS or LS.SYS (if appropriate)
PIP.SAV
DUP.SAV
DIR.SAV
LINK.SAV
EDIT.SAV
SRCCOM.SAV
RESORC.SAV
LIBR.SAV
IND.SAV
ODT.OBJ
MACRO.SAV
SYSMAC.SML
CREF.SAV
DEMOED.TXT
DEMOF1.FOR
DEMOX1.MAC
DEMOBG.MAC
DEMOFG.MAC

FORTRAN IV Language Volume

SWAP.SYS
RT11FB.SYS
PI.SYS (if running on Professional 325)
xx.SYS (system device handler)
LP.SYS or LS.SYS (if appropriate)
PIP.SAV
DUP.SAV
DIR.SAV
EDIT.SAV
FORTRA.SAV
DEMOF1.FOR

LINK Volume

SWAP.SYS
RT11FB.SYS
PI.SYS (if running on Professional 325)
xx.SYS (system device handler)
LP.SYS or LS.SYS (if appropriate)
PIP.SAV
DUP.SAV
DIR.SAV
LINK.SAV
SYSLIB.OBJ (including FORLIB.OBJ)
FORLIB.OBJ (if not included in SYSLIB.OBJ)
BASIC-11 Language Volume

SWAP.SYS
RT11FB.SYS
PI.SYS (if running on Professional 325)
xx.SYS (system device handler)
LP.SYS or LS.SYS (if appropriate)
PIP.SAV
DUP.SAV
DIR.SAV
LINK.SAV
EDIT.SAV
BASIC.SAV
SYSLIB.OBJ

2.4.6 Limiting Components on the System Volume

Limit the system volume to necessary and frequently used system components. Place on the system volume only one monitor file and only the handler files for devices in your configuration. For example, a typical RX01 system volume need have only:

SWAP.SYS
RT11FB.SYS
DX.SYS (RX01 device handler)
LP.SYS (if your system includes a line printer)

The device handler is not included in the monitor file, so you must include the system device handler on any system volume. If you use the BL monitor or the SJ monitor without multiterminal support, you do need TT.SYS, because the terminal service is not resident in those monitors.

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

2.5 Installing Software Updates

To make sure that RT-11 operates correctly, you must install software updates. Software updates are critical to system or component operation.

Software updates are performed to provide normal maintenance; for example, to correct documentation or software errors detected since the last software release. These updates also introduce new software functions that are not described in the documentation; for example, an update would be required to support new hardware, to release a bundled layered product, or to support a released layered product.

Software updates are distributed, not as source modules or binary patches, but as complete software module replacements. For example, if a software error exists in the PIP utility program, the update kit will contain the complete PIP software module. The correction will already have been installed and fully tested. Your only requirement is to copy the new module
from the distribution medium to your system, replacing the old software module with the new. After completing the update, you may be required to perform part of the installation procedure, for example, to rebuild your working system.

When you submit an SPR, the *RT–11 Software Dispatch Review* will report serious problems and may contain suggested solutions to the problems until the module in error is replaced.

Each software update kit includes:

- A letter from Software Product Services
- Release notes — identify the contents of the update kit, list the known problems that the kit fixes, and provide any special information about the update.
- User documentation — provides instructions on how to update the system.
- Distribution medium — contains new software modules and IND procedures to automate the updating process.

The distribution medium is the same as the medium on which you received the base system. It contains modules for both base system and layered products.

For a complete description of how to update your software, refer to the *RT–11 Update User’s Guide*.

### 2.6 Acquiring Sufficient Media

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

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>to back up distribution</td>
</tr>
<tr>
<td>1 (or more)</td>
<td>for working system</td>
</tr>
<tr>
<td>1 (an equal number)</td>
<td>to back up working system</td>
</tr>
<tr>
<td>?</td>
<td>for your application’s data storage requirements</td>
</tr>
<tr>
<td>11+</td>
<td>total</td>
</tr>
</tbody>
</table>

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

### 2.7 Choosing Software Customizations

Although the RT–11 components as distributed need no alteration for most applications, you can alter them in some ways. Many alterations require
your going through the system generation process. Others require less
time-consuming procedures, such as modifying the distributed monitors.
Compare Tables 1–3 and 1–4, and read the following sections to establish
what you need to do. Identify any non-system-generation customizations
you need to make, and record them on the worksheet at the end of this
chapter (Figure 2–2). You can perform the procedure for each selected cus-
tomization during the installation process (Chapters 3 through 11). System
generation is necessary only if you need a customization that cannot be
achieved with one of the procedures described in the following sections.

NOTE

Refer to Appendix D in the RT–11 System Generation Guide
for additional modifications that further customize specially
generated monitors.

DIGITAL strongly recommends that you use the SIPP utility to install
software customizations. Also, use the feature of SIPP that creates an indi-
rect command file when it installs the customization. In this way, you can
store copies of the indirect command files, so that you can easily install the
customizations again if necessary. When you invoke SIPP and it responds
with an asterisk, enter the following:

```
RUN SIPP
*filnam.COM=filnam.type/L
```

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

In order to customize your software, you must:

1. Determine which customization(s) you wish to make.

2. Locate the symbol of the software component you wish to modify. The
values of symbols for monitor customizations may be obtained from the
link maps distributed with RT–11 as the files RTBL.MAP, RTSJ.MAP,
RTFB.MAP, and RTXM.MAP. The values of symbols for utility custom-
izations may be found in the file CUSTOM.TXT.

3. Follow the instructions to modify the software component, substituting
the actual address value of the symbol in place of the symbol in the
update.

NOTE

In the software customizations reproduced in this manual,
lowercase alphabetic x represents unknown characters.
These characters vary according to the specific software com-
ponent. All numeric input values are octal.

2.7.1 Changing Characters That Indicate Insertion/Deletion

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

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

```
.BRUN SIPP
.SRCCOM.SAV
.Base? 0
.Offset? 1000

Base   Offset   Old   New?
000000  001000  076157  iA
000000  001000  <o>   iAm
000000  001001  <i>   iAn
000000  001002  <c>   $M, V
```

2.7.2 Changing Default Output Device from Line Printer to Terminal

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

2.7.3 Changing Location of VT11/VS60 Floating Vectors

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

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

In the customization, monitr.SYS is the name of the monitor file that you want to modify, $RMON is the value of that symbol from the appropriate monitor link map, and nnn is the location of the first VT11/VS60 vector on your system. Find the monitor link map for the monitor you want to alter, and use the value of $RMON in the update. Note that nnn must be an even octal value between 70 and 464.
2.7.4 Changing the Number of Directory Columns

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

In the customization, \texttt{ffffff} is an octal number (in the range 1 to 11) for the number of directory columns to be displayed when you use the DIRECTORY/FAST command. The value \texttt{nnnnnn} is an octal number (in the range 1 to 11) for the number of columns to be displayed when you use the DIRECTORY command.

2.7.5 Changing the Default Order of Directory Listings

You can change the default order in which directory entries are listed in a directory listing. The current default order is by position in the directory.

In the following customization, SWS is the value of that symbol from the CUSTOM.TXT file. The variable \texttt{nnnnnn} represents the address of SWS that SIPP returns in response to the search command. The variable \texttt{yyyyy} represents the contents of \texttt{nnnnnn + 2}. The variable \texttt{zzzzzz} requests the contents of SWS+2. The variable \texttt{xxx} represents one of the following default ordering options:

- \texttt{DAT} Order by date (earliest to latest)
- \texttt{NAM} Alphabetical order by file name
- \texttt{POS} Order by position in directory (current default)
- \texttt{SIZ} Order by size (smallest to largest)
- \texttt{TYP} Alphabetical order by file type

2-24 Preparing for Installation
2.7.6 Changing the Number of /Q Program Sections LINK Allows

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

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

Load section: address?

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

To change the number of /Q p-sects LINK allows, use SIPP to change LINK.SAV as follows. QSWNUM is the value of that symbol from the CUSTOM.TXT file. The number nnn must be in the range 1 to 177 (octal); it represents the number of p-sects you want.

2.7.7 Changing the Size of LINK's Library Module List

You can modify LINK to change the default size of LINK's list of library modules. LINK creates a list of 252 (octal) modules to be included from
libraries during the link operation. Because the size of each entry in this
list is larger in RT-11 Version 5 than in previous versions, the list may not
be large enough for your application. You can use the LINK /P option at
link time to increase the size of this list. If you want to change the default
size of the list (and avoid using the /P option), you can modify the linker.
You can still override the new default at link time by using /P.

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

NOTE

You must install this change if you use DIBOL. Make the
default number of modules LINK holds 400 (octal) or greater.

In the following customization, LMLSIZ is the value of that symbol from
the CUSTOM.TXT file, and nnnnnn is the number of modules the list
should hold.

```
, RUN SIPP
*LINK.SAV
Segment? 0
Base? 0
Offset? LMLSIZ

Segment Base Offset Old New
000000 000000 LMLSIZ ? nnnnnn
000000 000000 LMLSIZ+2 ?

*TRUC
```

2.7.8 Changing the Size of the QUEUE Work File

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

In the following customization, QSIZE and QCBLK are the value of those
symbols from the CUSTOM.TXT file, and nnn is the default size of the
work file in octal blocks. To compute the approximate size of the work file
that would be required for the number of files you need queued at once, use
the formula:

\[ \text{nnn} = (\text{max no. of file specs in queue at one time} + 1)/32 + 1 \]

```
, RUN SIPP
*QUEUE.REL
Base? 0
Offset? QSIZE

Base Offset Old New
000000 QSIZE 000005 nnn
000000 QSIZE+2 000000

Offset? QCBLK

Base Offset Old New
000000 QCBLK 000004 (nnn-1)
000000 QCBLK+2 000200 (nnn-1)*32
000000 QCBLK+4 000000

*TRUC
```

2–26 Preparing for Installation
2.7.9 Modifying EDIT

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

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

However, if the problem is troublesome for your particular application, you
can remove it by reducing the size of the display window. Use the following
customization to make this change, where DSARG is the value of that
symbol from CUSTOM.TXT file, and nnn is the number of lines to be dis-
played above and below the cursor. To eliminate the problem, make nnn
smaller than 12 (octal).

```
, RUN SIPP
&EDIT.SAVE
Base? 000000
Offset? DSARG+2

Base  Offset  Old  New?
000000  DSARG  000012  nnn
000000  DSARG+2 010200

&END
```

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

If you have an older terminal, you can correct the problem by making the
following change, so that you can use the ESCAPE code as documented in
the EDIT chapter of the RT-11 System Utilities Manual. In the change,
ALTMODE is the value of that symbol from the CUSTOM.TXT file, and nnn
represents the octal code that your terminal generates when you type the
ESCAPE key on it.

```
, RUN SIPP
&EDIT.SAVE
Base? 000000
Offset? ALTMODE+2
```

Preparing for Installation  2–27
The character $ echoes on the terminal, regardless of the octal value used for the ESCAPE code. However, EDIT recognizes only the ESCAPE code you specify, not both.

2.7.10 Extracting the Overlay Handlers from SYSLIB

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

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

1. Extract the unmapped overlay handler (for LINK/O overlays) from SYSLIB:
   
   .LIBRARY/EXTRACT SYSLIB OHANDL
   Global? $OVRH
   Global? $V

2. Extract the virtual overlay handler (for LINK/V overlays) from SYSLIB:
   
   .LIBRARY/EXTRACT SYSLIB VHANDL
   Global? $OVRHV
   Global? $V

3. Combine the files you extracted in a new library (which can have any name, including SYSLIB). You must combine the files in the indicated order:
   
   .LIBRARY/REMOVE/CREATE newlib VHANDL,OHANDL
   Global? $OVRH
   Global? $V

In the command, you need both the /REMOVE and /CREATE options. /CREATE creates the new library, and /REMOVE removes a duplicate global, $OVRH, from the library directory. This global appears in both OHANDL and VHANDL, because VHANDL includes the program code found in OHANDL. Remember, VHANDL processes both unmapped and virtual overlays.
You can also put either handler in a library by itself. OHANDL handles only unmapped overlays. VHANDL handles both unmapped and extended memory overlays, but it is larger than OHANDL.

In the following command, which creates a library from one of the overlay handlers, x is 0 for OHANDL or V for VHANDL.

```
.LIBRARY/CREATE newlib xHANDL
```

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

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

### 2.7.11 Installing Other Devices

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

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

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

```
.REMOVE LS
.INSTALL LP
```

The standard (distributed) monitors provide a total of 16 (decimal) device slots. If your application requires more than 16 device slots simultaneously, you must perform the system generation process to create your own monitor and device handlers.

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

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

The bootstrap cannot install support for some devices. Thus, different procedures are required. The following sections describe how you can change the control status register (CSR) and vector addresses for the line printer, serial printer, DECTape II, RX01 and RX02 diskettes, and MSCP disks, and how you can install hardware magtape support, set magtape parity and density, and change RP02 support to RP03.

### 2.7.11.1 CSR and Vector Addresses for Line Printers, DECTape II, and MSCP Disks

With the SET command, you can change the CSR and vector addresses of six devices: line printer, serial printer, DECTape II, RX01 and RX02 diskettes, and MSCP disks. You need to change the addresses if the
controller is installed at nonstandard addresses. When using the SET command, enter the following commands, where dd is the device mnemonic (LP, LS, DD, DX, DY, or DU), aaaaaa is the CSR address, and bbb is the vector address.

\[\begin{align*}
\text{\texttt{SET dd: CSR2=aaaaaaa}} & \text{ \texttt{[83]}} \\
\text{\texttt{SET dd: VEC2=bbb}} & \text{ \texttt{[83]}} \\
\text{\texttt{SET dd: CSR=aaaaaaa}} & \text{ \texttt{[83]}} \\
\text{\texttt{SET dd: VECTOR=bbb}} & \text{ \texttt{[83]}}
\end{align*}\]

In addition, the following commands are valid for MSCP disks:

\[\begin{align*}
\text{\texttt{SET DU: CSR3=aaaaaaa}} & \text{ \texttt{[83]}} \\
\text{\texttt{SET DU: VEC3=bbb}} & \text{ \texttt{[83]}} \\
\text{\texttt{SET DU: CSR4=aaaaaaa}} & \text{ \texttt{[83]}} \\
\text{\texttt{SET DU: VEC4=bbb}} & \text{ \texttt{[83]}}
\end{align*}\]

You can also partition MSCP disks and assign unit numbers to the ports. Use the following commands:

\[\begin{align*}
\text{\texttt{SET DU[n]: PART=x}} & \text{ \texttt{[83]}} \\
\text{\texttt{where n = unit number}} & \\
\text{\texttt{x = partition number}}
\end{align*}\]

\[\begin{align*}
\text{\texttt{SET DU[n]: PORT=x}} & \text{ \texttt{[83]}} \\
\text{\texttt{where n = unit number}} & \\
\text{\texttt{x = port number}}
\end{align*}\]

See the description of the SET command in the *RT–11 System User’s Guide* and the *RT–11 Software Support Manual* for more information on partitioning MSCP disks and assigning ports.

These commands permanently alter the handler .SYS file. That is, the CSR and vector addresses you specify remain in effect even if the system is rebooted. If you want to change the addresses again, you can use the SET command again.

### 2.7.11.2 Hardware Magtape Support

You can choose to install hardware magtape support, rather than use file-structured magtape support. File-structured handlers include hardware magtape handler features; however, hardware magtape handlers are smaller.

File-structured magtape handlers are distributed as part of the RT–11 operating system; hardware magtape handlers are not. If you want to use hardware magtape handlers, you must perform a system generation, using a new disk for output. System generation will produce either file-structured or hardware magtape handlers, but not both at the same time.
The files MT.SYS, MTX.SYS, MM.SYS, MMX.SYS, MS.SYS, and MSX.SYS are distributed file-structured handlers for TM11, T1U16, and TS11 magtape devices. The files MTHD.SYS, MTHDX.SYS, MMHD.SYS, MMHDX.SYS, MSHD.SYS, and MSHDX.SYS are the system-generated hardware magtape handlers for the same devices. There are two sets of handlers: one for SJ and FB monitors; the other for XM monitors. The filename suffix X indicates device support for XM monitors.

There are two ways to install a hardware magtape handler in place of the file-structured handler:

1. Rename the distributed file-structured handler to save it, rename the corresponding hardware magtape handler to the original name of the file-structured handler, and reboot the system to let the bootstrap install the handler.

2. Use the INSTALL and REMOVE commands:
   - Make sure the hardware magtape handler for your monitor and device is on the system disk.
   - Rename the distributed file-structured handler to save it.
   - Remove the file-structured handler from the system volume.
   - Rename the corresponding hardware magtape handler to the name of the distributed file-structured handler.
   - Install the renamed hardware magtape handler.

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

1. Rename and remove the file-structured handler. (You must remove the old handler before you can install the new one.)

   \texttt{\textasciitilde RENAME MT\textunderscore SYS MT\textunderscore FIL\\textasciitilde}
   \texttt{\textasciitilde REMOVE MT\textasciitilde}

2. Rename and install the hardware magtape handler.

   \texttt{\textasciitilde RENAME MTHD\textunderscore SYS MT\textunderscore SYS\\textasciitilde}
   \texttt{\textasciitilde INSTALL MT\textasciitilde}

The handler you have named MT.SYS will be installed in the device table when the system is bootstrapped.

Table 2–3 identifies hardware handlers that you must rename, if you are using an SJ or FB monitor. Make sure you save the distributed (file-structured) handler.
Table 2–3:  File-Structured and Hardware Magtape Handlers for SJ and FB Monitors

<table>
<thead>
<tr>
<th>Device</th>
<th>File-Structured Handler</th>
<th>Hardware Handler</th>
<th>Renamed Hardware Handler Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TM11</td>
<td>MT.SYS</td>
<td>MTHD.SYS</td>
<td>MT.SYS</td>
</tr>
<tr>
<td>TJU16</td>
<td>MM.SYS</td>
<td>MMHD.SYS</td>
<td>MM.SYS</td>
</tr>
<tr>
<td>TS11</td>
<td>MS.SYS</td>
<td>MSHD.SYS</td>
<td>MS.SYS</td>
</tr>
</tbody>
</table>

Table 2–4 identifies hardware handlers that you must rename, if you are using an XM monitor. Make sure you save the distributed (file-structured) handler.

Table 2–4:  File-Structured and Hardware Magtape Handlers for XM Monitors

<table>
<thead>
<tr>
<th>Device</th>
<th>File-Structured Handler</th>
<th>Hardware Handler</th>
<th>Renamed Hardware Handler Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TM11</td>
<td>MTX.SYS</td>
<td>MTHDX.SYS</td>
<td>MTX.SYS</td>
</tr>
<tr>
<td>TJU16</td>
<td>MMX.SYS</td>
<td>MMHDX.SYS</td>
<td>MMX.SYS</td>
</tr>
<tr>
<td>TS11</td>
<td>MSX.SYS</td>
<td>MSHDX.SYS</td>
<td>MSX.SYS</td>
</tr>
</tbody>
</table>

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

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

2.7.12 Modifying BATCH to Save Space

To minimize space demands on your system device, you can modify BATCH to access certain system programs on DK: rather than on SY:. These customizations allow you to store certain system programs on DK: rather than on SY: and let BATCH access them there.
You can make any or all of the customizations. Modify BATCH for the
system programs you need to remove from the system device. Copy the
programs for which you install changes to the device on which you want
them to reside; then, delete them from the system device (SY:). Finally, use
the ASSIGN command to assign the logical name DK: to the device to
which you copied the system programs. Then, run BATCH as usual.

The following change to BATCH makes DK: the default storage volume for
one of the specified programs. Use the octal value of nnnnnnn (from the
table) that corresponds to the program you want to affect.

<table>
<thead>
<tr>
<th>Program</th>
<th>Octal Value of nnnnnnn</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIR</td>
<td>1367</td>
</tr>
<tr>
<td>MACRO</td>
<td>2007</td>
</tr>
<tr>
<td>FORTRAN</td>
<td>2020</td>
</tr>
<tr>
<td>LINK</td>
<td>2037</td>
</tr>
<tr>
<td>PIP</td>
<td>2052</td>
</tr>
<tr>
<td>BASIC</td>
<td>2071</td>
</tr>
</tbody>
</table>

.*RUN SIPPER
*BATCH.SAV RED
Base?      BASE RED
Offset?    nnnnnnn RED

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Old</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASE</td>
<td>nnnnnnn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BASE</td>
<td>nnnnnnn+1</td>
<td>040</td>
<td></td>
</tr>
</tbody>
</table>

.*RUN
*SY:myjob=SY:myjob RED

2.7.13 **Modifying LINK to Change the Default SYSLIB Device**

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

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

.*RUN
*SY:dev=SY:dev RED
2.7.14 Modifying the Help Text

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

The files HELP.TXT and HELP.EXE, which together make up the program HELP.SAV, are not provided on the distribution kit. Therefore, if you want to change your HELP text, you must first recreate HELP.TXT and HELP.EXE from HELP.SAV using the unsupported utility SPLIT.

To recreate these files, type this command:

```
.SPLIT ddn:HELP.EXE, ddn:HELP.TXT=ddn:HELP.SAV/B::HLP1::HLP2
```

In the command, ddn: represents the device on which to create the files HELP.TXT and HELP.EXE, or the device on which HELP.SAV exists. The variables ..HLP1 and ..HLP2 represent the boundaries along which to split HELP.SAV. Refer to the file CUSTOM.TXT on your distribution kit for the values to substitute in the command line for ..HLP1 and ..HLP2.

HELP.SAV is the only file you need if you do not want to change the help text. However, if you do want to change the text that prints when you use the HELP command, you must perform the following procedure.

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

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

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

MACRO

• MACRO TOOLTIP NAME

TOPICNAME ONE LINE DESCRIPTION OF THE TOPIC

• SUBTOPICNAME

• TEXT ABOUT THE SUBTOPIC

• MORE TEXT (ANY NUMBER OF LINES)

• DESCRIPTION OF SUBTOPIC ITEM (USUALLY ONLY FOR THE SUBSTITUTE ‘OPTION’), ANY NUMBER OF LINES

• ANOTHER SUBTOPIC, STARTING FROM THE BLANK LINE ABOVE

• END

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

LIBR/MACRO HELP.MLB HELP.TXT

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

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

COPY/CONCATENATE HELP.EXE,HELP.MLB HELP.SAV

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

HELP

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

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

COPY HELP.(EXE+MLB) HELP.SAV

In this case too, you can back up HELP.EXE and HELP.TXT on another volume and delete them from the system volume. You can also delete the file HELP.MLB, since you can recreate it from the file HELP.TXT.
2.7.15 Preventing Fatal System Errors from Causing a Reset

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

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

In the customization, monitr.SYS is the name of the FB or XM monitor file you want to modify, and FATAL is the value of that symbol from the monitor link map.

```
.RUN SI          
*monitr.SYS      
Base? 0       
Offset? FATAL  
             Base Offset  Old New?
 000000  FATAL  000005  240827 
 000000  FATAL+2 010127  240827 
*STLC       
```

2.7.16 Running RT–11 in Less Memory Than Is Available

If your application requires that RT–11 run in less memory than is available, you can make a customization that allows you to bootstrap the system to run in the lower 12K words or 8K words of a 16K word machine. The BL, SJ, and FB monitors have bootstraps that allow the system to run in less memory than is available. (This cannot be done for XM monitors, because the XM monitor requires 32K memory.) The distributed monitors automatically make use of all available memory, since most applications require that RT–11 do so. However, if your configuration includes a hardware switch register and your application requires less memory, you can make the following customization. In the customization, monitr.SYS is the name of the monitor file that you want to modify, and BHALT is the value of that symbol from the monitor link map.

```
.RUN SI          
* monitr.SYS     
Base? 0        
Offset? BHALT  
             Base Offset  Old New?
 000000  BHALT  000405 086001 
 000000  BHALT+2 013704 086001 
*STLC        
```

If this is the hardware bootable monitor, you must write a new system bootstrap with the COPY/BOOT command after you install this change.
Once you make the change to your monitor, a halt occurs whenever you boot that monitor file. While the system is halted, set the switch register to one of the values from the following table and press the CONTINUE switch; the bootstrap operation then completes for the memory size you specify.

<table>
<thead>
<tr>
<th>Value</th>
<th>Size in Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>40000</td>
<td>8K</td>
</tr>
<tr>
<td>44000</td>
<td>9K</td>
</tr>
<tr>
<td>50000</td>
<td>10K</td>
</tr>
<tr>
<td>54000</td>
<td>11K</td>
</tr>
<tr>
<td>60000</td>
<td>12K</td>
</tr>
<tr>
<td>64000</td>
<td>13K</td>
</tr>
<tr>
<td>70000</td>
<td>14K</td>
</tr>
<tr>
<td>74000</td>
<td>15K</td>
</tr>
<tr>
<td>100000</td>
<td>16K</td>
</tr>
<tr>
<td>104000</td>
<td>17K</td>
</tr>
<tr>
<td>110000</td>
<td>18K</td>
</tr>
<tr>
<td>114000</td>
<td>19K</td>
</tr>
<tr>
<td>120000</td>
<td>20K</td>
</tr>
<tr>
<td>124000</td>
<td>21K</td>
</tr>
<tr>
<td>130000</td>
<td>22K</td>
</tr>
<tr>
<td>134000</td>
<td>23K</td>
</tr>
<tr>
<td>140000</td>
<td>24K</td>
</tr>
<tr>
<td>144000</td>
<td>25K</td>
</tr>
<tr>
<td>150000</td>
<td>26K</td>
</tr>
<tr>
<td>154000</td>
<td>27K</td>
</tr>
</tbody>
</table>

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

```
.RUN SIPP
*monitr.SYS
Base? 0
Offset? BHALT

Base Offset Old New?
000000 BHALT 000405 240
000000 BHALT 013704 12704
000000 BHALT 177570 nnnnnn
000000 BHALT 042704 

*HALT
```

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

To run RT–11 in exactly 28K words, use the customization in Section 2.7.21.
2.7.17 Setting VTCOM Default Dial String

Apply the following customization to VTCOM.REL or VTCOM.SAV to set a default dial string for the DIAL command.

In the customization, the symbol ..DIAL represents the address of the first character in the dial string, which is dictated by your modem. For example, the first character required in a dial string sent to a DF03 modem is `B. The next 39(decimal) bytes are reserved for the remainder of the dial string, including other characters your modem may require. You can find the correct value for the symbol ..DIAL in the file CUSTOM.TXT on your distribution kit. Replace the symbols aaa, bbb, and so on, with the characters that make up your dial string. Use a null character in the last byte to terminate the dial string.

To modify VTCOM.REL:

```
.R SIPP
*VTCOM.REL
Base? 0
Offset? ..DIAL

Base  Offset  Old  New?
000000  ..DIAL  000002 \RD

Base  Offset  Old  New?
000000  ..DIAL  002 \RD
000000  ..DIAL+1  ***  aaar\RD
000000  ..DIAL+2  ***  bbr\RD
  .  .  .  .
  .  .  .  .
000000  ..DIAL+51  ***  \RD\RD

*EOL\RD
```

2.7.18 Setting Upper Limit on a File Size

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

In the customization, monitr.SYS is the name of the monitor file that you want to modify, $RMON is the value of that symbol from the monitor link map, and nnnnnnn is the octal number of blocks that is to be the maximum file size for a general .ENTER.

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

Base  Offset  Old  New?
$RMON  000314  177777  nnnnnn \RD
$RMON  000316  ****  \RD\RD
```

Preparing for Installation 2–39
2.7.19 Specifying 50-Cycle Instead of 60-Cycle Clock Rate

You can modify the monitor so that it causes the TIME command to base calculations on a 50-cycle clock rate rather than a 60-cycle rate, which is the standard rate in RT–11. The 50-cycle clock has specialized uses and is the common frequency in Europe. To alter the rate, you must modify the monitor so that bit 5 is set in the monitor configuration word. This customization has no effect on Professional 300 series processors.

In the customization, monitr.SYS is the name of the monitor file that you want to modify, $RMON is the value of that symbol from the monitor link map, and the new value that you enter is the sum of the old value displayed by SIPP plus 40 (octal).

```
,RUN SIPP$RE
*monitr.SYS$RE
Base? $RMON$RE
Offset? 300$RE

Base  Offset  Old  New?
$RMON  000300  nnnnn  nnnnn+n=40$RE (You must add 40.)
$RMON  000302  000000  TRLY$RE

*CTRLC
```

2.7.20 Using CAPS–11 to Load Files

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

```
,RUN SIPP$RE
*CT.SYS$RE
Base? 1000$RE
Offset? 3463$RE

Base  Offset  Old  New?
001000  003463  001  0$RE
001000  003464  040  TRLY$RE

*CTRLC
```

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

```
\-LEVEL: /,,
LEVEL: OFF, BYTE NANO
/
2.7.21 Setting Upper Limit on Memory Size

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

```
,RUN SIPP
*monitr.SYS
Base? 10000
Offset? 0

Base Offset Old New?
001000 000060 170000 160000
001000 000062 001402
```

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

2.7.22 Suppressing the Bootstrap Message

If you want to prevent the monitor identification message from printing when you bootstrap a monitor, you can modify that monitor. In the customization, monitr.SYS is the name of the monitor file that you want to modify, and ..SLNT is the value of that symbol from the monitor link map.

```
,RUN SIPP
*monitr.SYS
Base? 0
Offset? ..SLNT

Base Offset Old New?
000000 ..SLNT 000000 1
000000 ..SLNT+2 001000
```

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

2.7.23 Suppressing the Start-Up Indirect Command File

If you want to prevent the start-up command file from executing when you bootstrap a monitor, you can modify that monitor. The standard monitors include start-up indirect command file support although you need not select it if you perform the system generation process to create special monitors. In the customization, monitr.SYS is the name of the monitor file that you
want to modify, and ..NIND is the value of that symbol from the monitor link map.

```
.RUN SIIPP
*monitr.SYS
Base?  0
Offset? ..NIND
       Base  Offset  Old  New?
       000000 ..NIND      004000  0
       000000 ..NIND+2    000044  1
*CTRL/C
```

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

### 2.7.24 Suppressing the Start-Up Indirect Command File Echo

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

```
.RUN SIIPP
*monitr.SYS
Base?  0
Offset? ..TTQU
       Base  Offset  Old  New?
       000000 ..TTQU      000000  1
       000000 ..TTQU+2    001403  1
*CTRL/C
```

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

### 2.7.25 Changing the Bootstrap Message

If you want to change the monitor identification message that appears when you bootstrap a monitor, you can modify that monitor. Run SIIPP to modify the monitor file. Place the string (the message) in the monitor image starting at location BSTRNG. End the string with a null byte. The string must not be more than 20(decimal) bytes long, including the null byte. If the monitor you modify is the hardware bootable monitor, write a new system bootstrap with the COPY/BOOT command.

### 2.7.26 Changing the Default Device for Indirect Command Files

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

In the customization, monitr.SYS is the name of the monitor file that you
want to modify, ..ATDK is the value of that symbol from the monitor link
map, and nnn is the new default device name.

```
.RUN SIPI
*monitr.SYS
Base? ..ATDK
Offset? ..ATDK+1

<table>
<thead>
<tr>
<th>Base</th>
<th>Offset</th>
<th>Old</th>
<th>New?</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000 ..ATDK 015270</td>
<td>iRRED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000000 ..ATDK &lt;DK&gt;</td>
<td>iRnnn RED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000000 ..ATDK+2 &lt;AW1&gt;</td>
<td>iRELY RED</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

2.7.27 Changing the Default File Type for Indirect Command Files

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

In the customization, monitr.SYS is the name of the monitor file that you
want to modify, ..ATFX is the value of that symbol from the monitor link
map, and nnn is the new default file type.

```
.RUN SIPI
*monitr.SYS
Base? ..ATFX
Offset? ..ATFX+1

<table>
<thead>
<tr>
<th>Base</th>
<th>Offset</th>
<th>Old</th>
<th>New?</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000 ..ATFX 012445</td>
<td>iRRED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000000 ..ATFX &lt;COM&gt;</td>
<td>iRnnn RED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000000 ..ATFX+2 &lt;***&gt;</td>
<td>iRELY RED</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

2.7.28 Changing the Default Device for the FRUN Command

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

In the customization, monitr.SYS is the name of the monitor file that you
want to modify, ..FRDK is the value of that symbol from the monitor link
map, and nnn is the new default device name.
2.7.29 Changing the Default File Type for the FRUN Command

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

In the customization, monitr.SYS is the name of the monitor file that you want to modify, ..FRUX is the value of that symbol from the monitor link map, and nnn is the new default file type.

```
Base  Offset   Old  New?
000000  ..FRUX  070524  REL
000000  ..FRUX  <REL>  iRnnn REL
000000  ..FRUX+2 <ANG>  REL
```

2.7.30 Changing the Default Device for the EDIT Command

If you want to change the default device for the EDIT command, you can modify the monitor. Normally, when you invoke an editor by typing the EDIT command, the default device where the monitor looks for EDIT.SAV is SY:. If you have a special application, you can change this default device to any three-character device name. The customization changes the default device for EDIT only; it does not change the default device for KED or K52.

In the customization, monitr.SYS is the name of the monitor file that you want to modify, ..EDDV is the value of that symbol from the monitor link map, and nnn is the new default device name.

```
Base  Offset   Old  New?
000000  ..EDDV  075250  iRnnn
000000  ..EDDV  <SY>  iRnnn
000000  ..EDDV+2 <CHU>  REL
```

2–44 Preparing for Installation
2.7.31 Changing the Default File Name for the EDIT Command

If you want the monitor to run a program other than the default editor when you type the EDIT command, you can modify the monitor to change the default file name.

In the customization, monitr.SYS is the name of the monitor file that you want to modify, PROGDF is the value of that symbol from the monitor link map, and EEE is 200 (octal) plus the new byte value of one of the following symbols from the monitor link map:

\$\$EDIT Command value for default editor EDIT
\$\$KED Command value for default editor KED
\$\$K52 Command value for default editor K52

\.RUN SIPPER
\#monitr.SYS
Base? 0
Offset? PROGDF

<table>
<thead>
<tr>
<th>Base</th>
<th>Offset</th>
<th>Old</th>
<th>New?</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000</td>
<td>PROGDF</td>
<td>103225</td>
<td>\YES</td>
</tr>
</tbody>
</table>

Base Offset Old New?
000000 PROGDF 225 EEE\YES
000000 PROGDF+1 206 \YES

\.END

2.7.32 Using Examine and Deposit Above the Background Job

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

In the customization, monitr.SYS is the name of the monitor file that you want to modify, and ..EMON is the value of that symbol from the monitor link map.

\.RUN SIPPER
\#monitr.SYS
Base? 0
Offset? ..EMON

<table>
<thead>
<tr>
<th>Base</th>
<th>Offset</th>
<th>Old</th>
<th>New?</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000</td>
<td>..EMON</td>
<td>103041</td>
<td>240\YES</td>
</tr>
<tr>
<td>000000</td>
<td>..EMON+2</td>
<td>103007</td>
<td>\YES</td>
</tr>
</tbody>
</table>

\.END

2.7.33 Changing the Default Device for QUEMAN

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

In the customization, monitr.SYS is the name of the monitor file that you want to modify. ..QULP is the value of that symbol from the monitor link map, and dv is the two-character device name that you want as the default.

```
, RUN SIPP
* monitr.SYS
Base? 0
Offset? ..QULP
```

<table>
<thead>
<tr>
<th>Base</th>
<th>Offset</th>
<th>Old</th>
<th>New?</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000</td>
<td>..QULP</td>
<td>050114</td>
<td>IA</td>
</tr>
<tr>
<td>000000</td>
<td>..QULP</td>
<td>&lt;L&gt;</td>
<td>IA</td>
</tr>
<tr>
<td>000000</td>
<td>..QULP+1</td>
<td>&lt;P&gt;</td>
<td>IA</td>
</tr>
<tr>
<td>000000</td>
<td>..QULP+2</td>
<td>&lt;:&gt;</td>
<td>IA</td>
</tr>
</tbody>
</table>

```
*CTRL/C
```

### 2.7.34 Changing the Indirect Command File Nesting Depth

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

In the customization, monitr.SYS is the name of the monitor file that you want to modify, $RMON is the value of that symbol from the monitor link map, and nnn is the maximum indirect file nesting depth that you want; nnn should be a small integer, and must not be zero.

```
, RUN SIPP
* monitr.SYS
Base? $RMON
Offset? 377
```

<table>
<thead>
<tr>
<th>Base</th>
<th>Offset</th>
<th>Old</th>
<th>New?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$RMON</td>
<td>000377</td>
<td>003</td>
<td>nnn</td>
</tr>
<tr>
<td>$RMON</td>
<td>000400</td>
<td>xxx</td>
<td>nnn</td>
</tr>
</tbody>
</table>

```
*CTRL/C
```

### 2.7.35 Changing the Threshold for Resuming Output-Stalled Jobs

You can improve FB or XM system throughput by changing the threshold for resuming output-stalled jobs. In an RT–11 FB or XM system, a job is placed in a stalled state whenever it has terminal output to print but there is no room in its terminal output ring buffer. The system restarts the job when room becomes available in that ring buffer. The system’s default mode of operation is to restart the job as soon as a single character of space

---

2–46 Preparing for Installation
is available. If more than one job is running, this mode of operation can cause the system to spend much time swapping in the context of a job, simply to have it output a single character and then stall again.

You can patch the monitor so that a job that is stalled waiting for room in the terminal output buffer does not resume execution until several characters are available in the ring buffer. If you have a foreground or system job that produces a large amount of terminal output, installing this change can greatly improve system throughput.

In the following customization, monitr.SYS is the name of the monitor file that you want to modify, TTON is the value of that symbol from the monitor link map, and nnn is the threshold value for resuming a terminal output stalled job.

The monitor will resume such a job when there are nnn-1 characters left to print in the output ring. The default value, 50 (octal), is the size of the ring; consequently, the monitor resums a job when 47 (octal) characters are left to print (that is, when only one character position is available in the output ring). You can specify any value from 1 to the size of the output ring buffer (which is normally octal 50, but can be changed at system generation time). Note that a value of 1 will cause the job to stay stalled until its ring buffer is empty, and may cause terminal output to appear unaligned.

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

<table>
<thead>
<tr>
<th>Base</th>
<th>Offset</th>
<th>Old</th>
<th>New?</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000</td>
<td>..TTON</td>
<td>000050</td>
<td>nnn</td>
</tr>
<tr>
<td>000000</td>
<td>..TTON+2</td>
<td>xxxxxxx</td>
<td>only</td>
</tr>
</tbody>
</table>

2.7.36 Changing the Default Number of Directory Segments

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

If the default number of directory segments for a volume is unacceptable for your application, you can use the /SEGMENTS:n option with INITIALIZE to initialize a volume with n directory segments. (Refer to the RT-11 System User's Guide.) However, if you want DUP to use a specific number, you can modify DUP to change DUP's directory segment table (duplicated in the following).
In the customization, nnnnnn is an octal offset value from the table below. The value mmmmmmm is an octal number (in the range 1 to 37) for the number of directory segments you want as the default. Find the size of the volume for which you are changing the default number of directory segments in the table. Enter the corresponding offset value in nnnnnn. Then, enter the number of default directory segments you want in mmmmmmm.

<table>
<thead>
<tr>
<th>Size of Volume (Blocks) (decimal)</th>
<th>Offset Value (octal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;= 512</td>
<td>000002</td>
</tr>
<tr>
<td>&lt;= 2048</td>
<td>000006</td>
</tr>
<tr>
<td>&lt;= 12288</td>
<td>000012</td>
</tr>
<tr>
<td>&lt;= 65535</td>
<td>000016</td>
</tr>
</tbody>
</table>

2.7.37 Changing the Banner Page Width

You can modify QUEUE to change the width of the banner page on line printer output from 132 positions to 80 positions.

In the customization, ..NB1, ..NB2, and ..NB3 are the values of those symbols from the link map.

Base Offset Old New
00000 ..NB1 122 15          
00000 ..NB1+1 124 12         
00000 ..NB1+2 055 0          
00000 ..NB1+3 0B1 0          

2–48 Preparing for Installation
2.7.38 Modifying Listing Page Length in LINK

If you do not use line printer paper of a standard size (10.5 inches long) or if your configuration does not include a line printer, you may need to modify the listing page length in LINK. RT-11 LINK sets the number of lines printed on each listing page at 60. This line count is generally satisfactory only for applications with line printers that use paper 10.5 inches long. However, you may require a listing of a different length. In the customization, LINPPG is the value of that symbol in the link map, and nnn is the desired listing page length (in lines).

```
* RUN SIPPED
* LINK.SAVRED
Segment? 0 RED
Base? 0 RED
Offset? LINPPG RED
Segment Base Offset Old New?
000000 000000 LINPPG ? nnn RED
000000 000000 LINPPG+2 ? ONLY RED
* FRM C
```

2.7.39 Assigning the HELP File

If you want to assign your HELP file to a file and/or device other than SY.HELP.SAV, you can modify the monitor.

In the customization, monitr.SYS is the name of the monitor file that you wish to modify, and ..HELP is the value of that symbol from the monitor link map. The RAD50 file specification starts at ..HELP.

```
* RUN SIPPED
* monitr.SYSRED
Base? 0 RED
Offset? ..HELP RED
Base Offset Old New?
000000 ..HELP 075250 iRED
000000 ..HELP <SY> iRedu RED
000000 ..HELP+2 <HEL> iRefi RED
000000 ..HELP+4 <P> iRnam RED
000000 ..HELP+6 <SAV> iRext RED
000000 ..HELP+10 <xxx> ONLY RED
* FRM C
```
2.7.40 Changing the Device from Which IND.SAV Is Run

By default, the system runs IND.SAV from SY: when KMON is set to IND and you type @IND. You can modify the monitor to change the device from which the IND.SAV file is run. To change the device, change the contents of ..INDN to the RAD50 device name of the device from which IND will be run. For example, to run IND from the virtual device (VM), open the ..INDN location in the link map and change its contents to VM.

In the customization, monitr.SYS is the name of the monitor file that you wish to modify, and ..INDN is the value of that symbol from the monitor link map.

```
.RUN SIPP
#monitr.SYS
Base?  0
Offset? ..INDN

    Base   Offset    Old   New?
  000000  ..INDN   075250  IR  OK
  000000  ..INDN    <SY> IR dev fail
  000000  ..INDN+2  <IND>  Y not OK
```

2.7.41 Supporting Bad Block Replacement in User-Written Handlers

If a user-written handler supports bad block replacement, one of the following customizations must be applied. If all bad blocks are replaceable (such as the RL01/02), apply customization A. If only bad sector errors are replaceable (such as the RK06/07), apply customization B.

**Customization A**

```
.R SIPP
#DUP.SAV
Segment?  1
Base?  0
Offset? ARDSP 4

Segment  Base  Offset  Old  New?
  000001  000030  ARDSP  000  377  OK
  000001  000030  ARDSP+1  000  419  Y not OK
```

**Customization B**

```
.R SIPP
#DUP.SAV
Segment?  1
Base?  0
Offset? SRDSP 4

Segment  Base  Offset  Old  New?
  000001  000000  SRDSP  000000  \OK
```
### Segment Base Offset Old New?

<table>
<thead>
<tr>
<th>Base</th>
<th>Offset</th>
<th>Old</th>
<th>New?</th>
</tr>
</thead>
<tbody>
<tr>
<td>000001</td>
<td>000000</td>
<td>SRDPS</td>
<td>000</td>
</tr>
<tr>
<td>000001</td>
<td>000000</td>
<td>SRDPS+1</td>
<td>000</td>
</tr>
</tbody>
</table>

#### 2.7.42 Supporting User-Written Magtape Handlers

If a user-written magtape handler is used, the following three customizations must be made. All three customizations assume that the device code for the device is 377 (octal).

### Customization 1

```plaintext
.R SIPP\(\text{OE}\)
#DUP.SAV\(\text{OE}\)
Segment? 1 \(\text{OE}\)
Base? 0 \(\text{OE}\)
Offset? MTDPS \(\text{OE}\)
```

<table>
<thead>
<tr>
<th>Segment</th>
<th>Base</th>
<th>Offset</th>
<th>Old</th>
<th>New?</th>
</tr>
</thead>
<tbody>
<tr>
<td>000001</td>
<td>000000</td>
<td>MTDPS</td>
<td>000</td>
<td>(\text{OE})</td>
</tr>
</tbody>
</table>

### Customization 2

```plaintext
.R SIPP\(\text{OE}\)
#PIP.SAV\(\text{OE}\)
Segment? 1 \(\text{OE}\)
Base? 0 \(\text{OE}\)
Offset? PIPMT \(\text{OE}\)
```

<table>
<thead>
<tr>
<th>Segment</th>
<th>Base</th>
<th>Offset</th>
<th>Old</th>
<th>New?</th>
</tr>
</thead>
<tbody>
<tr>
<td>000001</td>
<td>000000</td>
<td>PIPMT</td>
<td>000</td>
<td>(\text{OE})</td>
</tr>
</tbody>
</table>

### Customization 3

```plaintext
.R SIPP\(\text{OE}\)
#MDUP.SAV\(\text{OE}\)
Base? 0 \(\text{OE}\)
Offset? MDUPMT \(\text{OE}\)
```

<table>
<thead>
<tr>
<th>Base</th>
<th>Offset</th>
<th>Old</th>
<th>New?</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000</td>
<td>MDUPMT</td>
<td>000</td>
<td>(\text{OE})</td>
</tr>
</tbody>
</table>

Preventing for Installation 2–51
2.7.43 Replacing Magtape Bootstrap in DISMT1.COM

Module MBOT16.BOT is a primary bootstrap for producing a 1600 bits/in, phase-encoded, bootable magtape. It is usable on either MS or MM drives. To use this bootstrap, edit DISMT1.COM by replacing MBOOT.BOT with MBOT16.BOT in the INITIALIZE command line. This line should then read INITIALIZE/NOQUERY/VOLUMEID/FILE:DIS:MBOT16.BOT TAP:.

2.7.44 Changing Default File Type of Logical Disk Files

By default, the file type of logical disk files is .DSK. You can alter the default file type in one of two ways. (Note that the same procedures also apply to LDX.SYS for the XM monitor.)

Customization 1

Edit the conditional file SYSGEN.CND and add the following line:

DEF$LD = 'Rtype'  ; define default for LD filetype

where typ is the desired default file type. Then, reassemble and relink LD.SYS using the conditional file SYSGEN.CND.

Customization 2

:UNPROTECT SY:LD.SYS
:R SIPP
:SY:LD.SYS/A
Base? ..LDEX
Offset? 0

| Base | Offset | Old | New?
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>000000</td>
<td>..LDEX</td>
<td>?????</td>
<td>IR A1</td>
</tr>
<tr>
<td>000000</td>
<td>..LDEX</td>
<td>&lt;DSK&gt;</td>
<td>IRtype A2</td>
</tr>
<tr>
<td>000000</td>
<td>..LDEX+2</td>
<td>&lt;DSK&gt;</td>
<td>IRtype A3</td>
</tr>
<tr>
<td>000000</td>
<td>..LDEX+4</td>
<td>&lt;DSK&gt;</td>
<td>IRtype A4</td>
</tr>
<tr>
<td>000000</td>
<td>..LDEX+6</td>
<td>&lt;DSK&gt;</td>
<td>IRtype A5</td>
</tr>
<tr>
<td>000000</td>
<td>..LDEX+10</td>
<td>?????</td>
<td>E10 A6</td>
</tr>
</tbody>
</table>

*RUN SIPP

2.7.45 Changing QUEUE to Allow First Form Feed

QUEUE suppresses the first form feed in a file, because QUEUE assumes that the LP or LS handler is set to FORM0, which generates a form feed. If the line printer handler is set to NOFORM0, no form feed is generated.

You can apply the following customization so that QUEUE will never suppress the initial form feed in a file. If you apply this customization and set your printer handler to FORM0, an extra blank page will be produced.

In the customization, ..DOFF is the value of that symbol from the file CUSTOM.TXT on your distribution kit.

:RUN SIPP

*QUEUE.REL
Base? 0
Offset? ..DOFF

2–52 Preparing for Installation
Base   Offset   Old   New?
000000   .D0FF   xxxxxx   240 \RED
000000   .D0FF+2   xxxxxx   240 \RED
000000   .D0FF+4   xxxxxx   \TRLY \RED

2.7.46 Changing Listing Page Length in MACRO and CREF

The default listing page size for MACRO and CREF, 60 lines long, is suitable for line printers that use standard size line printer paper (10.5 inches long). If you use line printer paper that is not standard size, or if your configuration does not include a line printer, you may need to modify the listing page length in MACRO and CREF. In the customizations, nnn represents the desired listing page length (octal). Substitute for the symbol PGSIZE the value of that symbol given in the file CUSTOM.TXT on your distribution kit.

To modify MACRO:

.RUN SIPPST
*MACRO.SAVED
Segment?   0RED
Base?      0RED
Offset?    PGSIZE \RD

Segment Base Offset Old New?
000000    000000    PGSIZE    000074    nnn \RD
000000    000000    PGSIZE+2  003467    \TRLY \RD

To modify CREF:

.RUN SIPPST
*CREF.SAVED
Base?    0RED
Offset?  PGSIZE \RD

Segment Base Offset Old New?
000000    000000    PGSIZE    000074    nnn \RD
000000    000000    PGSIZE+2  003467    \TRLY \RD

2.7.47 Changing Default Command File Processor to IND

In the distributed monitors, the default command file processor is KMON. Apply the following customization to change the default command file processor to IND. After you have applied this customization, you may use an IND control file as your start-up command file, providing the file IND.SAV resides on your system volume.

In the customization, monitr.SYS is the monitor file you want to modify, and ..INDR is the value of that symbol from the appropriate monitor link map. If the monitor you customize is hardware bootable, write a new system bootstrap with the COPY/BOOT command.
If you do not wish to apply this customization, you can set the default command file processor to IND by setting the conditional IND$ON to 1 in the conditional file.

### 2.7.48 Limiting Amount of Memory KEX Requests

KEX, by default, requests all available 16-bit memory up to 32KW. This customization allows you to limit the amount of memory KEX requests with a .SETTOP programmed request.

In the customization, the symbol ..MAXM represents the highest address KEX will request. (In the distributed monitors, this value is set to 177776, the highest address possible.) If you set this value too low, KEX will fail and print the error message ?KEX–F–Insufficient memory. You can find the correct value for the symbol ..MAXM in the file CUSTOM.TXT on your distribution kit. Replace the symbol nnnnnnn with the new maximum address value you want KEX to request.

### 2.7.49 Changing MACRO’s Default .LIST/.NLIST Options

MACRO–11/RT is distributed with the following .NLIST defaults:

```
.NLIST LD,ME,MEB,TTM
```

You may change the .NLIST defaults to any of the following, where the following bit significance applies (bit asserted implies ".NLIST"):

- **BEX** = 2
- **BIN** = 4
- **CND** = 10
- **COM** = 20
- **LD**  = 40
- **LOC** = 100
- **MC**  = 200
MD = 400
ME = 1000
MEB = 2000
SEQ = 4000
SRC = 10000
SYM = 20000
TOC = 40000
TTM = 100000

.RUN SIPP
*MACRO.SAV
Segment? "RED"
Base? "RED"
Offset? "LDBITS"

Base Offset Old New?
000000 LDBITS 103040 nnnnnn "RED"
000000 LDBITS+2 x"RED" "RED"

2.7.50 Changing MACRO's Default .ENABLE/.DISABLE Options

MACRO–11/RT is distributed with the following .DSABL defaults:

.DSABL ABS,AMA,CDR,DBG,FPT,LCM,LSB,MCL

You may change the .DSABL defaults to any of the following, where the following bit significance applies (bit asserted implies " .DSABL "):

ABS = 1
AMA = 2
CDR = 4
CRF = 10
DBG = 20
FPT = 40
GBL = 100
LC = 200
LCM = 400
LSB = 1000
MCL = 2000
PNC = 4000
REG = 10000

Note that the flag must always be set.

.RUN SIPP
*MACRO.SAV
Segment? "RED"
Base? "RED"
Offset? "EDBITS"
2.7.51 Modifying KED Default File Type

KED now supports default file types. When editing a file, the default input file type is .MAC; the default output file type is the same as the input file type. When inspecting a file, the default input file type is .LST. There is no default input file type when creating a file.

To specify a file with no file type, type only the file name and the period separating the file name and type (FILNAM.).

You can modify the default file types with the following software customization:

```
.R SIPP.R
*aaa.SAV/A.R
Base? 0.R
Offset? bbbbbb.R
```

In this customization, substitute KED, K52, or KEX for aaa. Substitute the value for the symbol ..EEXT (for the editing-file default) or ..IEXT (for the inspecting-file default) for bbbbbb. These values can be found in CUSTOM.TXT on your distribution kit (be sure to use the proper KED variant). Substitute the three-character file type default you want for ccc.

2.7.52 Changing SPOOL's Work File Size

SPOOL allocates by default 1024(decimal) blocks on SFD: or SY: for its work file SPOOL.SYS. You can change the default size of SPOOL.SYS by applying the following software customization. In the customization, ..SPSZ is the offset for the current number of blocks SPOOL allocates for its work file. Substitute for ..SPSZ the value provided in the file CUSTOM.TXT. nnnnnn is the number (octal) of blocks you want SPOOL to allocate for its work file. xxxxxxx is a number that varies; it is not important for you to know this number.
If there is not enough room on the volume for a work file of the default size, SPOOL.SYS occupies the largest empty area on the volume.

### 2.7.53 Changing SPOOL's Output Device

You can change SPOOL's default output device to any RT–11 non-file-structured device by installing the following software customization. In the customization, substitute for ..SPSO the value provided in the file CUSTOM.TXT. nnn is the new output device's mnemonic. xxxxxx represents a number that varies but is not important for you to know.

```
, RUN SIPP
* SPOOL.REL/Ada
Base?  0
Offset? ..SPSZ

  Base   Offset  Old   New?
  000000  ..SPSZ    xxxxxx
  000000  ..SPSZ+2  xxxxxx

*CTRL/C
```

### 2.7.54 Changing LINK for Default 132-Column Link Map and Global Cross-Reference Table

The following customization patch causes the link map and global cross-reference table to default to 132 columns, rather than 80 columns. The value of ..WDSZ can be found in CUSTOM.TXT on your distribution kit.

```
, RUN SIPP
* LINK.SAV
Segment?  1
Base?  0
Offset? ..WDSZ

  Segment  Base   Offset  Old   New?
  000001   000000  ..WDSZ    000003
  000001   000000  ..WDSZ+2  xxxxxx

*CTRL/C
```

Preparing for Installation 2–57
**Figure 2–2: Installation Worksheet**

<table>
<thead>
<tr>
<th>SELECTED COMPONENTS</th>
<th>SIZE IN BLOCKS</th>
<th>SOURCE FILES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ RT11BL.SYS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ RT11SJ.SYS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ RT11FB.SYS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ RT11XM.SYS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Device Handler</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other Device Handlers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ SWAP.SYS</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Utility Programs</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ BATCH.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ BINCOM.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ BUP.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ CREF.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ DIR.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ DUMP.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ DUP.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ EDIT.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ ERROUT.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ FILEX.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ FORMAT.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ HELP.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ IND.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ KED.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ KEX.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ KSZ.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ LBR.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ LINK.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ MDUP.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ PAT.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ POP.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ QUEMAN.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ QUEUE.REL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ RESORC.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ SETUP.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ SIPP.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ SLP.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ SPOOL.REL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ SRC.COM.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ UCL.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ VTCOM.REL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ VTCOM.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Help Files</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ HELP.SAV</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Libraries</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ SYS.LIB.OBJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ SYSMAC.SML</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Assemblers</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>□ MACRO</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ARRANGEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System Volume</strong></td>
</tr>
<tr>
<td>Volume 1</td>
</tr>
<tr>
<td>Volume 2</td>
</tr>
<tr>
<td>Volume 3</td>
</tr>
<tr>
<td>Volume 4</td>
</tr>
<tr>
<td>Volume 5</td>
</tr>
</tbody>
</table>

**Customizations**

<table>
<thead>
<tr>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAGE</td>
</tr>
<tr>
<td>PAGE</td>
</tr>
<tr>
<td>PAGE</td>
</tr>
</tbody>
</table>
Chapter 3
Installing a System Distributed on RX01 to Run on a Small Device

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

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

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

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

1. Bootstrap the distribution volume.
2. Preserve the distribution volumes.
3. Install software updates.
4. Create the working system from chosen components.
5. Install the bootstrap on volumes that need to be bootable.
6. Customize the system.
7. Compress each volume.
8. Preserve the working system.
9. Test the working system.
10. If appropriate, perform the system generation process.
The following sections correspond to each of these steps and describe in detail the procedures you must perform to complete each step.

Figure 3–1: Sample Backup Volumes

3.1 Bootstrapping the Distribution Volume

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

Begin by making sure that the processor is powered up but not running. Insert distribution volume 1 in RX01 Unit 0. For RX01 diskette, the device name is DX..

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

RT–11 should respond with the following message if you have successfully bootstrapped the volume:
After you boot the system, you can create a start-up file containing any commands or sequence of commands that you would normally want to execute at the start of each terminal session.

You can update your start-up file at any time to change, add, or delete commands. When you first create the file, or after you update it, you can execute it with the @ command, so that the commands it contains become effective.

### 3.2 Preserving the Distribution Volumes

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

**NOTE**

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

Insert a blank volume (write-enabled) in Unit 1. Then use the INITIALIZE command to initialize the blank volume (procedure follows). Use the /BAD-BLOCKS option with INITIALIZE to cover any bad blocks that may be on your volume. If the volume contains bad blocks, the ?DUP–W–Bad blocks detected nnnnnn message appears on the terminal.

**NOTE**

DIGITAL recommends that you use only volumes that do not have bad blocks when you back up distribution volumes and build a working system. To ascertain whether an already initialized volume has bad blocks, use the command DIRECTORY/BAD xnn:. You can use volumes with bad blocks later, for working or data volumes. However, there is an exception to this rule. If your volumes are TU58 tape cartridges, you
can use the procedure in Section 2.4.3 when you build your working system but not when you back up the distribution volumes. The procedure in Section 2.4.3 creates dummy bad blocks on cartridges to improve response time.

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

There may be a significant delay (as much as 8 minutes) while the system scans the volume for bad blocks and creates a new directory. The monitor dot appears when this process is complete.

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

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

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

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

**NOTE**

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

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

3-4 Installing a System Distributed on RX01 to Run on a Small Device
Remove the newly created backup volume from Unit 1, write-protect it, and label it "Backup RT-11 V05 1". (Use a soft-tipped pen when you label diskettes.)

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

Then, type:

```
.COPY/BOOT x\$1:RT11FB.SYS x\$1:RE
```

Mount output volume in x\$1: Continue?

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

```
Y
g
Mount input volume in x\$0: Continue?
```

Replace the volume in Unit 0 with distribution volume 2 (write-protected).

```
Y
g
Mount system volume in x\$0: Continue?
```

Replace the volume in Unit 0 with distribution volume 1 (write-protected).

```
Y
g
```

Remove the volume from Unit 1, write-protect it, and label it "Backup RT-11 V05 2".

Again type the command:

```
.COPY/WAIT/OUTPUT:x\$1: x\$0:
```

Mount output volume in x\$1: Continue?

Insert another initialized blank volume in Unit 1.

```
Y
g
Mount input volume in x\$0: Continue?
```

Replace the volume in Unit 0 with distribution volume 3 (write-protected).

```
Y
g
Mount system volume in x\$0: Continue?
```

Replace the volume in Unit 0 with distribution volume 1 (write-protected).

```
Y
g
```

Installing a System Distributed on RX01 to Run on a Small Device 3–5
Remove the volume from Unit 1, label it "Backup RT-11 V05 3", write-protect it, and insert another initialized blank volume in Unit 1. Repeat these procedures to copy the rest of the distribution volumes.

Now halt the processor. Replace distribution volume 1 with backup volume 1 (write-protected) in Unit 0. Write-protect the distribution volumes, and store them. You will use the backup copies to build a working system.

Use the hardware bootstrap to boot backup volume 1.

```
RT-11FB V05.xx
```

(Followed by any start-up file commands.)

**NOTE**

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

Next, remove the protection from all the files on the backup volumes. The files on the distribution volumes have been protected to prevent you from accidentally deleting them. (Refer to the *RT-11 System User's Guide* for a description of file protection.) When you copied the files to the backup volumes, RT-11 also copied the protection (note the P that prints next to the file size in the directory). For the rest of these procedures, you need to remove the protection from the backup volumes. Type the following command to remove it from the files on backup volume 1. The /SYSTEM option is required to remove protection from system files (.SYS) if wildcards are used in the file specification.

```
.UNPROTECT/SYSTEM *.
Files unprotected:
DK:aaaaaaaa.ttt
DK:bbbbbb.ttt
DK:ccccccc.ttt
DK:ddddddd.ttt
...
DK:zzzzzzz.ttt
```

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

```
.UNPROTECT/SYSTEM xx1:*
Files unprotected:
xx1:aaaaaaaa.ttt
xx1:bbbbbb.ttt
xx1:ccccccc.ttt
xx1:ddddddd.ttt
...
xx1:zzzzzzz.ttt
```

3-6 Installing a System Distributed on RX01 to Run on a Small Device
Replace backup volume 2 with backup volume 3 in Unit 1 and repeat this command. In the same way, remove protection from the files on the rest of the backup volumes.

3.3 Installing Software Updates

To make sure that RT–11 operates correctly, you must install updated software modules at this point. Updated modules are critical to system or component operation. They correct software errors discovered since the last release, and may add new functions to the operating system.

Updated software modules are distributed in update kits. Each kit contains complete software module replacements. Software corrections will already have been installed in the new modules and fully tested. Your only requirement is to copy the new modules from the distribution medium to your system, using the procedures described in the RT–11 Update User’s Guide. Updates are distributed on the same medium on which you received the base system and contain modules for both base system and layered products.

When you submit an SPR, the RT–11 Software Dispatch Review will report serious problems and may contain suggested solutions to problems, until the module containing an error is corrected by an update kit.

For a complete description of how to update your software, refer to the RT–11 Update User’s Guide.

3.4 Creating the Working System from Chosen Components

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

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

NOTE

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

Then, use the COPY command to copy selected files from backup volume 1 to the volume that becomes your working system volume. The /SYSTEM option is required for copying .SYS files only if wildcards are used in the input file specification.

COPY xx0:filnam.type xx1:filnam.type

Installing a System Distributed on RX01 to Run on a Small Device 3–7
You can use the following command to avoid typing numerous file specifications.

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

To copy files from nonbootable volumes, alternate volumes.

Use the SET command to set the USR to NOSWAP.

```
.SET USR NOSWAP
```

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

```
.COPY/WAIT xx1:filnam.typ xx0:filnam.typ
Mount input volume in xx1: Continue?
```

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

```
Y@?
Mount output volume in xx0: Continue?
```

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

```
Y@?
Mount system volume in xx0: Continue?
```

Replace the volume in Unit 0 with backup volume 1 (write-protected).

```
Y@?
```

Repeat this procedure to copy all the files for the working system volume. When you have copied all the files you have planned for the working system volume, label it “RT–11 Working System V05 1”. Repeat these procedures to create the other volumes in the working system.
When you have created and labeled all the working system volumes, you can permit the USR to swap again.

.SET USR SWAP

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

Once you have created your system, you need to install the bootstrap on any volumes that must be bootable (that is, that you can use as the system volume). Generally, any volume that includes a monitor file and system device handler should be bootable (but remember that the volume would need SWAP.SYS and, for the SJ monitor, TT.SYS).

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

.COPY/BOOT x1:RT11aa.SYS x1:.

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

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

RT-11aa VM 05.x
(Followed by any start-up file commands.)

Store the updated backup volumes for future updating purposes.

3.6 Customizing the System

You may want to make certain customizations (described in Section 2.7) to the distributed RT-11 components. At this point, perform the procedures to implement any of these software customizations. Table 1–3 summarizes the available customizations and directs you to the section in Chapter 2 that describes a particular customization and the procedure for implementing it.

NOTE

Later, you can perform the system generation process to implement additional customizations. (See Section 3.10.)
3.7 Compressing Each Volume

DIGITAL recommends that you compress each working system volume to make its free space contiguous. Consolidating free space allows you to use space on the volume that would otherwise be too fragmented to be usable. However, if your volumes are TU58 tape cartridges and you create bad blocks to avoid excessive rewinds, the amount of contiguous free space possible is limited. (Refer to Section 2.4.3.)

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

`SQUEEZE xx0;RE
xx0:/Squeeze; Are you sure? Y(RE

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

RT-11aa V05.00
(Followed by any start-up file commands.)
.

The system automatically reboots when you compress a system volume.

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

`SQUEEZE xx1;RE
xx1:/Squeeze; Are you sure? Y(RE
.

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

NOTE

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

3.8 Preserving the Working System

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

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

`PROTECT/SYSTEM **RE
Files protected:
DK:aaaaaa.ttt
DK:bbbbbb.ttt
DK:cccccc.ttt
DK:ddddd.ttt
.
.
DK:zzzzzz.ttt

3–10 Installing a System Distributed on RX01 to Run on a Small Device
To protect files on other volumes in the working system, insert each volume in Unit 1 and use the following command:

```
.PROTECT/SYSTEM xx1:.*
Files protected:
xx1:aaaaaa.ttt
xx1:bbbbbb.ttt
xx1:cccccc.ttt
xx1:dddddd.ttt

xx1:zzzzzz.ttt
```

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

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

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

### 3.9 Testing the Working System

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

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

- SWAP.SYS
- RT11FB.SYS
- xx.SYS (system device handler)
- LP.SYS
- EDIT.SAV
- MACRO.SAV
- SYSMAC.SML
- LINK.SAV
- PIP.SAV
- DUP.SAV
- DIR.SAV
- DEMOBS.MAC
- DEMOFG.MAC
Display the directory of the system volume on the terminal. The directory varies according to your particular working system. As long as a directory prints, you need not worry if it does not match the one in the following example.

```
DIRECTORY/BRIF/COLUMNS:1 SY:SY
     dd-mmm-yyyy
SWAP .SYS
RT11FB.SYS
LP .SYS
DX .SYS
EDIT .SAV
MACRO .SAV
SYSMAC.SML
LINK .SAV
PIP .SAV
.

XXX Files, bbb Blocks
fff Free blocks
```

**NOTE**

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

Before you can execute the background and foreground demonstration programs, you must first edit, assemble, and link the background program, DEMO_GB.MAC, and you must assemble and link the foreground program, DEMOFG.MAC.

### 3.9.1 Preparing the Background Demonstration Program

#### 3.9.1.1 Edit the Background Demonstration Program — Use a text editor, for example, EDIT.SAV, to modify the background demonstration program, DEMO_GB.MAC. One of the output lines in the program is preceded by a semicolon, which makes the line a comment field. The semicolon prevents the line from being printed; thus, it must be deleted from that line. If DEMO_GB.MAC is a protected file, remove the protection before making the edits (UNPROTECT SY:DEMO_GB.MAC).

```
EDIT SY:DEMO_GB.MAC
*F1:ASCII
*OAD
*EX
```

#### 3.9.1.2 Assemble the Background Demonstration Program — The background program, DEMO_GB.MAC, is an assembly language source file; it must be assembled and linked before you can execute it. To assemble DEMO_GB.MAC and obtain a listing, make sure that your configuration has a line printer that is on-line and ready.
NOTE

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

ASSIGN TT: LST

Assemble DEMOBG.MAC as follows:

MACRO/LIST: LST: DEMOBG
(See Figure 3–2.)

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

RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC

Figure 3–2: DEMOBG Assembly Listing

Installing a System Distributed on RX01 to Run on a Small Device 3–13
3.9.1.3 Link the Background Demonstration Program — Link the program DEMOBG.OBJ to produce an executable background program, DEMOBG.SAV.

```plaintext
.LINK DEMOBG
```

3.9.1.4 Run the Background Demonstration Program — Run the program DEMOBG to check the results of the first exercise.

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

If you did not delete the semicolon character, the last line will not print. Return to the monitor by typing two CTRL/Cs.

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

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

```plaintext
.RENAME SY:DEMOB.G.BAK SY:DEMOB.G.MAC
```

Then, repeat the editing procedure.

3.9.2 Preparing the Foreground Demonstration Program

To execute the foreground program, you must assemble the program DEMOFG.MAC, link it for the foreground, and execute it in conjunction with DEMOBG.SAV. DEMOFG.MAC is a small foreground program that sends a message every 2 seconds to DEMOBG (running in the background), telling it to ring the terminal bell. DEMOBG recognizes these messages and rings the bell once for each message sent.

Although DEMOFG is always active, sending messages to the background every 2 seconds, this exercise can execute other programs in the background besides DEMOBG. The circuit is complete and messages are successfully received and honored only when DEMOBG is active. During those periods when DEMOBG is not running, DEMOFG enters the messages in the monitor message queue. Once you restart DEMOBG in the background, the system immediately releases all the messages queued since the last forced exit, resulting in many successive bell rings. When the queue is empty, the normal send/receive cycle resumes and the bell rings every 2 seconds as each current message is sent and honored.

3.9.2.1 Assemble the Foreground Demonstration Program — The foreground demonstration program, DEMOFG.MAC, is an assembly language source
file; it must be assembled and linked before you can use it. Assemble DEMOFG.MAC as follows:

```
.MACRO LIST:LST: DEMOFG
.
```

The output resulting from this MACRO command is an object file called DEMOFG.OBJ. This file resides on your system volume.

### 3.9.2.2 Link the Foreground Demonstration Program —
You must link the DEMOFG.OBJ file to produce an executable program. Use the /FOREGROUND option to produce the load module DEMOFG.REL. The .REL file type signifies to the system that the file is a foreground program and is to be run as a priority job.

```
.LINK FOREGROUND DEMOFG
.
```

### 3.9.2.3 Run the Foreground and Background Demonstration Programs —
Type the following command to load and start DEMOFG.REL as the foreground job.

```
,FRUN DEMOFG F>
```

- FOREGROUND DEMONSTRATION PROGRAM
- SENDS A MESSAGE TO THE BACKGROUND PROGRAM DEMOBG EVERY 2 SECONDS, TELLING IT TO RING THE BELL.

```
B>
```

DEMOFG.REL is now running and queuing the message for DEMOBG every 2 seconds. Now execute DEMOBG.SAV in the background and receive the messages.

```
,RUN DEMOBG
```

(The bell rings quickly several times, then once every 2 seconds.)

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

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

```
,DIRECTORY
```

(The bell stops ringing.)

```
   d - m m m - y y
```

(The directory of the device DK prints on the terminal.)
Rerun DEMOBG to collect all the foreground messages queued while the directory was printing.

```plaintext
.RUN DEMOBG
```

(The bell rings several times in rapid succession, then rings once every 2 seconds.)

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

```
,.
```

(The bell stops ringing.)

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

```
EXIT
F>
```

```
B>
UNLOAD FRED
```

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

### 3.10 Performing the System Generation Process

If you have decided that you need RT–11 features that are available only if you generate your own monitor(s) and handlers, perform the system generation process at this point. You should have thoroughly studied Chapter 1 to make this decision and to establish that you can perform the system generation process on your particular hardware configuration. Read the *RT–11 System Generation Guide* for guidance in planning and performing system generation.
Chapter 4
Installing a System Distributed on RX01 to Run on a Hard Disk

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

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

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

1. Bootstrap the distribution volume.
2. Copy the distribution volumes to the disk.
3. Preserve the distribution volumes.
4. Install software updates.
5. Create the working system from chosen components.
6. Install the bootstrap on the disk.
7. Customize the system.
8. Compress the disk.
9. Preserve the working system.
10. Test the working system.
11. If appropriate, perform the system generation process.

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

**Figure 4–1: Sample Backup Disks**

4.1 **Bootstrapping the Distribution Volume**

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

Begin by making sure that the processor is powered up but not running. Insert the distribution volume labeled 1 (write-protected) in RX01 diskette Unit 0. For RX01 diskette, the device name is DX.

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

RT–11 should respond with the following message if you have successfully bootstrapped the volume:
After you boot the system, you can create a start-up file containing any commands or sequence of commands that you would normally want to execute at the start of each terminal session.

You can update your start-up file at any time to change, add, or delete commands. When you first create the file, or after you update it, you can execute it with the @ command, so that the commands it contains become effective.

### 4.2 Copying the Distribution Volumes to the Disk

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

If the disk is an RK05 disk, mount the disk in Unit 0, format the disk, and then initialize it and cover any bad blocks on it.

If the disk is another type, initialize the disk and replace or cover any bad blocks. You have a choice of replacing or covering bad blocks if your disk is an RK06, RK07, RL01, or RL02. If your disk is another type, you should cover any bad blocks.

To replace bad blocks, use the INITIALIZE/REPLACE command, and to cover bad blocks, use the INITIALIZE/BADBLOCKS command (procedures follow). In the commands, xx is the permanent device name for your disk. If the disk contains bad blocks, the ?DUP-W-Bad blocks detected nnnnnn message appears on the terminal, but you can use the disk.

**RK05 disk**

```
,FORMAT RK0:RD
RK0:/FORMAT-Are you sure? YRD
?FORMAT-I-Formatting complete

,INITIALIZE/BADBLOCKS RK0:RD
RK0:/Initialize; Are you sure? YRD
?DUP-I-No bad blocks detected RK0:
```
Other type disk

,INITIALIZE/BADBLOCKS xX0:ED
xX0:/Initialize; Are you sure? YED
?DUP-I-No bad blocks detected xX0:

or

,INITIALIZE/REPLACE xX0:ED
xX0:/Initialize; Are you sure? YED
?DUP-I-No bad blocks detected xX0:

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


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

,SQUEEZE/OUTPUT:xx0: yy0:ED

NOTE

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

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

You must also copy the rest of the distribution volumes to the disk. Insert distribution volume 2 in RX01 Unit 1. In the following command, xx is the permanent device name for your disk, and yy is the name for your distribution device. The /SYSTEM option is required for copying .SYS files only if wildcards are used in the input file specification.

,COPY/SYSTEM yy1: xx0:ED
Files copied:
    yy1:aaaaaa.ttt to xx0:aaaaaa.ttt
    yy1:bbbbbb.ttt to xx0:bbbbbb.ttt

    ...
    ...
    yy1:zzzzzz.ttt to xx0:zzzzzz.ttt
Insert distribution volume 3 in RX01 Unit 1, and repeat this command. Copy all the distribution volumes to your disk in this way.

NOTE

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

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

```
COPY/BOOT xx0:RT11FB.SYS xx0:
SQUEEZE xx0:
xx0:/Squeeze? Are you sure? Y
```

4.3 Preserving the Distribution Volumes

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

```
RT-11FB V05.00
(Followed by any start-up file commands.)
```

NOTE

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

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

```
UNPROTECT/SYSTEM *.ttt
Files unprotected:
DK:aaaaaa.ttt
DK:bbbbbb.ttt
DK:cccccc.ttt
DK:dddddd.ttt
DK:zzzzzz.ttt
```

Installing a System Distributed on RX01 to Run on a Hard Disk 4–5
4.4 Installing Software Updates

To make sure that RT–11 operates correctly, you must install updated software modules at this point. Updated modules are critical to system or component operation. They correct software errors discovered since the last release, and may add new functions to the operating system.

Updated software modules are distributed in update kits. Each kit contains complete software module replacements. Software corrections will already have been installed in the new modules and fully tested. Your only requirement is to copy the new modules from the distribution medium to your system, using the procedures described in the RT–11 Update User's Guide. Updates are distributed on the same medium on which you received the base system and contain modules for both base system and layered products.

When you submit an SPR, the RT–11 Software Dispatch Review will report serious problems and may contain suggested solutions to problems, until the module containing the error is corrected by an update kit.

For a complete description of how to update your software, refer to the RT–11 Update User’s Guide.

4.5 Creating the Working System from Chosen Components

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

Mount a blank disk in disk Unit 1 and initialize it. If the disk is an RK05 disk, format it before you initialize it.

**RK05 disk**

```
$FORMAT RK1:
RK1:/FORMAT-Are you sure? Y
?FORMAT-I-Formatting complete

$INITIALIZE/BADBLOCKS RK1:
RK1:/Initialize: Are you sure? Y
?DUP-I-No bad blocks detected RK1:
```

**Other type disk**

```
$INITIALIZE/BADBLOCKS xx1:
xx1:/Initialize: Are you sure? Y
?DUP-I-No bad blocks detected xx1:
```

or

```
$INITIALIZE/REPLACE xx1:
xx1:/Initialize: Are you sure? Y
?DUP-I-No bad blocks detected xx1:
```
Copy the files you have selected from the disk in Unit 0 to the disk in Unit 1, which will become your working system disk. If you use the following command, RT–11 queries you about all the files on the disk in Unit 0, and you can choose the files it copies.

```
COPY/SYSTEM/QUERY xx0: xx1:
Files copied:
xx0:aaaaa.ttt to xx1:aaaaa.ttt? Y(N)
xx0:bbbbbb.ttt to xx1:bbbbbb.ttt? N(Y)
(and so on)
```

4.6 Installing the Bootstrap on the Disk

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

```
COPY/BOOT xx1:RT11aa.SYS xx1:
```

In this command, you identify the device on which the monitor that contains the bootstrap information resides (your disk), the name of that monitor file (RT11BL, RT11SJ, RT11FB, or RT11XM), and the device on which you need to install the bootstrap (your disk). This command copies bootstrap information from the monitor file to blocks 0 and 2 through 5 of the same disk.

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

```
RT-11aa V05.x
(Followed by any start-up file commands.)
```

4.7 Customizing the System

You may want to make certain customizations (described in Section 2.7) to the distributed RT–11 components. At this point, perform the procedures to implement any of these software customizations. Table 1–3 summarizes the available customizations and directs you to the section in Chapter 2 that describes a particular customization and the procedure for implementing it.

**NOTE**

Later, you can perform the system generation process to implement additional customizations. (See Section 4.11.)

4.8 Compressing the Disk

DIGITAL recommends that you compress the working system disk to make its free space contiguous. Consolidating free space allows you to use space on the disk that would otherwise be too fragmented to be usable.
Use the SQUEEZE command to compress free space. (The volume must be write-enabled.) The squeeze operation does not move files with the .BAD file type.

```
.SQUEEZE xx0:
xx0:/Squeeze ARE YOU SURE? Y
RT-11a V05.xx
```

(Followed by any start-up file commands.)

The system automatically reboots when you compress a system disk.

### 4.9 Preserving the Working System

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

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

```
.PROTECT/SYSTEM *.*
```

Files protected:

```
DK:aaaaaa.ttt
DK:bbbbbb.ttt
DK:cccccc.ttt
DK:ddddd.ttt
```

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

#### 4.9.1 Backing Up the System on RX01 Diskettes

Insert a blank RX01 diskette in Unit 1 (with RT-11 still booted from disk Unit 0). Use the INITIALIZE/BADBLOCKS command to initialize the blank diskette. Then, repeat the process to initialize the appropriate number of diskettes.

Copy all the files in your working system, using the COPY/MULTIVOLUME command. In the command, yy is the name for your disk. The system copies all the files from the input volume that will fit on the output volume. When no more files will fit on the output volume, the system prompts you to mount the next output volume and prints the Continue? message.

Mount another blank, initialized diskette, and type Y to continue. Continue to mount diskettes in this fashion until all the files are copied.

```
.COPY/MULTIVOLUME/SYSTEM yy0: DX1:*
```

Files copied:

```
yy0:aaaaaa.ttt to DX1:aaaaaa.ttt
yy0:bbbbbb.ttt to DX1:bbbbbb.ttt
```

(and so on)
Mount next output volume in DX1:; Continue? Y
(Log of files copied)
Mount next output volume in DX1:; Continue? Y
(Log of files copied)
Mount next output volume in DX1:; Continue? Y
(Log of files copied)
Mount next output volume in DX1:; Continue? Y
(Log of files copied)
Mount next output volume in DX1:; Continue? Y
(and so on)

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

.COPY/BOOT xx1:RT11aa.SYS xx1:  

.

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

4.9.2 Backing Up the System on Another Disk

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

**RK05 disk**

..FORMAT RK1:  
RK1:/FORMAT-Are you sure? Y  
?FORMAT-I-Formatting complete  

..INITIALIZE/BADBLOCKS RK1:  
RK1:/Initialize: Are you sure? Y  
?DUP-I-No bad blocks detected RK1:

**Other type disk**

..INITIALIZE/BADBLOCKS xx1:  
xx1:/Initialize: Are you sure? Y  
?DUP-I-No bad blocks detected xx1:  

or

..INITIALIZE/REPLACE xx1:  
xx1:/Initialize: Are you sure? Y  
?DUP-I-No bad blocks detected xx1:  

Copy all the files in your working system.

..SQUEEZE/OUTPUT:

Installing a System Distributed on RX01 to Run on a Hard Disk 4-9
Copy the bootstrap to the backup working system disk. In the following command, aa is BL, SJ, FB, or XM.

```
COPY/BOOT xx1:RT11aa.SYS xx1:RT
```

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

### 4.10 Testing the Working System

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

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

- SWAP.SYS
- RT11FB.SYS
- xx.SYS (system device handler)
- LP.SYS
- EDIT.SAV
- MACRO.SAV
- SYSMAC.SML
- LINK.SAV
- PIP.SAV
- DUP.SAV
- DIR.SAV
- DEMOBUG.MAC
- DEMOFG.MAC

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

```
DIRECTORY/BRIEF/COLUMNS:1 SY:RT
    dd-mmm-yyyy
SWAP   .SYS
RT11FB  .SYS
LP      .SYS
DX      .SYS
EDIT    .SAV
```

4–10 Installing a System Distributed on RX01 to Run on a Hard Disk
NOTE

If you have shifted output to a scope and the directory scrolls by too quickly to read, type \[\text{\texttt{\textasciitilde CRL}S}\] to stop the display and \[\text{\texttt{\textasciitilde CRL}O}\] to restart it.

Before you can execute the background and foreground demonstration programs, you must first edit, assemble, and link the background program, DEMOBG.MAC, and you must assemble and link the foreground program, DEMOFG.MAC.

4.10.1 Preparing the Background Demonstration Program

4.10.1.1 Edit the Background Demonstration Program — Use a text editor, for example, EDIT.SAV, to modify the background demonstration program, DEMOBG.MAC. One of the output lines in the program is preceded by a semicolon, which makes the line a comment field. The semicolon prevents the line from being printed; thus, it must be deleted from that line. If DEMOBG.MAC is a protected file, remove the protection before making the edits (UNPROTECT SY:DEMOBG.MAC).

\[\text{\texttt{\textasciitilde EDIT SY:DEMOBG.MAC\textasciitilde}}\]
\[\text{\texttt{\textasciitilde F:\tildas\textit{ASCII}ES01\textit{ES}}}\]
\[\text{\texttt{\textasciitilde 0A\textit{ES01\textit{ES}}}\texttt{\textit{ES}}}\]
\[\text{\texttt{\textasciitilde 0E\textit{ES01\textit{ES}}}\texttt{\textit{ES}}}\]

4.10.1.2 Assemble the Background Demonstration Program — The background program, DEMOBG.MAC, is an assembly language source file; it must be assembled and linked before you can execute it. To assemble DEMOBG.MAC and obtain a listing, make sure that your configuration has a line printer that is on-line and ready.

\[\text{\texttt{\textasciitilde ASSIGN LP: LST: N\textasciitilde}}\]
NOTE

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

.Assign
.TT: LIST: .RED
.

Assemble DEMOBG.MAC as follows:

.Macro LIST: LST: DEMOBG: .RED
(See Figure 4–2.)

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

.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC: .RED
.

Figure 4–2: DEMOBG Assembly Listing

4.10.1.3 Link the Background Demonstration Program — Link the program DEMOBG.OBJ to produce an executable background program, DEMOBG.SAV.

.Link DEMOBG: .RED
.

4–12 Installing a System Distributed on RX01 to Run on a Hard Disk
4.10.1.4 Run the Background Demonstration Program — Run the program DEMOBG to check the results of the first exercise.

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

If you did not delete the semicolon character, the last line will not print. Return to the monitor by typing two CTRL/Cs.

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

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

```
.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC
```

Then, repeat the editing procedure.

4.10.2 Preparing the Foreground Demonstration Program

To execute the foreground program, you must assemble the program DEMOFG.MAC, link it for the foreground, and execute it in conjunction with DEMOBG.SAV. DEMOFG.MAC is a small foreground program that sends a message every 2 seconds to DEMOBG (running in the background), telling it to ring the terminal bell. DEMOBG recognizes these messages and rings the bell once for each message sent.

Although DEMOFG is always active, sending messages to the background every 2 seconds, this exercise can execute other programs in the background besides DEMOBG. The circuit is complete and messages are successfully received and honored only when DEMOBG is active. During those periods when DEMOBG is not running, DEMOFG enters the messages in the monitor message queue. Once you restart DEMOBG in the background, the system immediately releases all the messages queued since the last forced exit, resulting in many successive bell rings. When the queue is empty, the normal send/receive cycle resumes and the bell rings every 2 seconds as each current message is sent and honored.

4.10.2.1 Assemble the Foreground Demonstration Program — The foreground demonstration program, DEMOFG.MAC, is an assembly language source file; it must be assembled and linked before you can use it. Assemble DEMOFG.MAC as follows:

```
.MACRO/LIST:LST: DEMOFG
```

The output resulting from this MACRO command is an object file called DEMOFG.OBJ. This file resides on your system volume.
4.10.2.2 Link the Foreground Demonstration Program — You must link the DEMOFG.OBJ file to produce an executable program. Use the /FOREGROUND option to produce the load module DEMOFG.REL. The .REL file type signifies to the system that the file is a foreground program and is to be run as a priority job.

```
.LINK/FOREGROUND DEMOFG
```

4.10.2.3 Run the Foreground and Background Demonstration Programs — Type the following command to load and start DEMOFG.REL as the foreground job.

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

DEMOFG.REL is now running and queuing the message for DEMOBG every 2 seconds. Now execute DEMOBG.SAV in the background and receive the messages.

```
RUN DEMOBG
```

(The bell rings quickly several times, then once every 2 seconds.)

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

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

```
DIR C:
```

(The bell stops ringing.)

```
DIRECTORY
 dd-mmm-yyyy
```

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

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

```
RUN DEMOBG
```

(The bell rings several times in rapid succession, then rings once every 2 seconds.)

4–14 Installing a System Distributed on RX01 to Run on a Hard Disk
RT-11 DEMONSTRATION PROGRAM
IF INCORRECTLY EDITED, THIS IS THE LAST LINE.
WELL DONE.

(The bell stops ringing.)

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

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

4.11 Performing the System Generation Process

If you have decided that you need RT-11 features that are available only if you generate your own monitor(s) and handlers, perform the system generation process at this point. You should have thoroughly studied Chapter 1 to make this decision and to establish that you can perform the system generation process on your particular hardware configuration. Read the RT-11 System Generation Guide for guidance in planning and performing system generation.
Chapter 5
Installing a System Distributed on Hard Disk to Run on Hard Disk

If RT-11 was distributed to you on hard disk (RL01/RL02, RK05, or RC25), and you intend to build a system to run on hard disk, perform the procedures described in this chapter. These procedures assume that your configuration includes two disk drives (not necessarily of the same type).

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

To install your system, perform the steps summarized in the following list. Sections 5.1 through 5.10 describe the procedures involved in each step. Figure 5-1 shows the various backup disks you create when you install RT-11 on a hard disk other than RC25.

Figure 5-2 illustrates the backup procedure for installing RT-11 on an RC25 disk. Notice that files are deleted from the backup disk (a fixed device) to create the working system disk.

1. Bootstrap the distribution disk.
2. Preserve the distribution disk.
3. Install software updates.
4. Create the working system from chosen components.
5. Install the bootstrap on the disk.
6. Customize the system.
7. Compress the disk.
8. Preserve the working system.
9. Test the working system.

10. If appropriate, perform the system generation process.

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

Figure 5–1: Sample Backup Disks (RL01, RL02, RK05)

Figure 5–2: Sample Backup Disks (RC25)
5.1 Bootstrapping the Distribution Disk

The first procedure you perform when installing RT-11 is bootstrapping the distribution disk.

Begin by making sure that the processor is powered up but not running. Mount the distribution disk (write-protected) in disk Unit 0. The device name is DL, RK, or DU, depending on the type of disk device. Use the hardware bootstrap to boot the disk. (If your configuration does not include a hardware bootstrap, refer to Appendix B for the toggle-in software bootstrap.)

RL02 and RC25 disks are normally installed manually. If you successfully bootstrap an RL02 or an RC25 disk, RT-11 responds with the dialog and installation verification message below. The dialog gives you the choice of installing RT-11 manually or by using the automatic installation procedure.

If you successfully bootstrap an RL01 or an RK05 disk, RT-11 responds with the installation verification message below.

Welcome to RT-11 Version 5.
You have bootstrapped the RT-11 Distribution Disk. Use this disk to install your RT-11 system, then store it in a safe place.

RT-11 V5 provides an automatic installation procedure which will back up your distribution disk and build a working system disk which should be used for your work with RT-11. This working system disk will only contain the RT-11 operating system. After the RT-11 installation is complete, follow the installation instructions packaged with any optional languages or utility software which you will be using.

Press the "RETURN" key when ready to continue.

You can choose to install RT-11 manually. This procedure is described in the RT-11 Installation Guide.

If you are a new user of RT-11, DIGITAL highly recommends that you use the automatic installation procedure.

Do you want to use the automatic installation procedure? (Type YES or NO and press the "RETURN" key): NO

The standard RT-11 monitor bootstrap (RT11FB) will now be copied to this disk and the system disk will be rebooted. Please refer to the RT-11 Installation Guide for assistance while manually installing RT-11 Version 5.

Press the "RETURN" key when ready to continue.

RT-11FB V05.xx

(TYPE V5USER.TXT

RT-11 V5.xx

Installation of RT-11 ....(followed by V5USER.TXT message)....

Installing a System Distributed on Hard Disk to Run on Hard Disk 5–3
Now you can create a start-up file containing any commands or sequence of commands that you would normally want to execute at the start of each terminal session.

You can update your start-up file at any time to change, add, or delete commands. When you first create the file, or after you update it, you can execute it with the @ command, so that the commands it contains become effective.

5.2 Preserving the Distribution Disk

The first procedure you perform with the running RT-11 system is to copy the distribution disk for backup, as a safety measure in case of machine failure or human error. You can then use the backup copy of the distribution disk to build your working system.

Mount a blank disk in the other disk drive, if necessary. (The disk in Unit 1 of RC25 devices is permanently mounted.) If the disk is an RK05 disk, format it, and then initialize it and cover any bad blocks on it.

If the disk is another type, initialize the disk and replace bad blocks (using the command INITIALIZE/REPLACE) or cover bad blocks (using the command INITIALIZE/BADBLOCKS). You have a choice of replacing or covering bad blocks if your disk is an RK06, RK07, RL01, or RL02. (Refer to the RT-11 System User’s Guide.) If your disk is another type, you should use INITIALIZE/BADBLOCKS to cover any bad blocks. If the disk contains bad blocks, the ?DUP-W-Bad blocks detected nnnn message appears on the terminal, but you can use the disk. In the commands, xx is the permanent device name and n is the unit number for your disk.

RK05 disk

. FORMAT RKn: ox
  RKn:/FORMAT-Are you sure? Y ox
  ?FORMAT-I-Formatting complete

  . INITIALIZE/BADBLOCKS RKn: ox
  RKn:/Initialize-Are you sure? Y ox
  ?DUP-I-No bad blocks detected RKn:

Other type disk

  . INITIALIZE/BADBLOCKS xxn: ox
  xxn:/Initialize-Are you sure? Y ox
  ?DUP-I-No bad blocks detected xxn:

or

  . INITIALIZE/REPLACE xxn: ox
  xxn:/Initialize-Are you sure? Y ox
  ?DUP-I-No bad blocks detected xxn:

5-4 Installing a System Distributed on Hard Disk to Run on Hard Disk
The system scans the disk for bad blocks and creates a new directory. The monitor dot appears when this process is complete.

The next step in the preservation process is to copy all the files from the distribution disk to the initialized blank disk.

The following command transfers files from the distribution to the backup disk and consolidates all the empty space at the end of the disk. In the command, yyn is the device name and unit number of your distribution disk, and xxn is the device name and unit number of your backup disk.

.SQUEEZE/OUTPUT:xxn: yyn:

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

Next, copy the bootstrap to the backup disk.

.COPY/BOOT xxn:RT11FB.SYS xxn:

Now halt the processor, remove the distribution disk from Unit 0, and store it. For RC25 devices, use the hardware bootstrap to boot the backup disk on disk Unit 1; for all other devices, mount the backup disk on Unit 0, and use the hardware bootstrap to boot the backup disk.

RT–11FB V05.xx
(Followed by any start-up file commands.)

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

Next, remove the protection from all the files on the backup disk. The files on the distribution disk have been protected to prevent you from accidentally deleting them. (Refer to the RT–11 System User's Guide for a description of file protection.) When you copied the files to the backup disk, RT–11
also copied the protection. (Note the P that prints next to the file size in the
directory.) For the rest of these procedures, you need to remove the protec-
tion from the backup disk. Use the following command:

`\UNPROTECT/SYSTEM *.\*\$`  
Files unprotected:  
DK:aaaaaaaa.ttt  
DK:bbbbbb.ttt  
DK:cccccccc.ttt  
DK:dddddddd.ttt  
  .  
  .  
  .  
DK:zzzzzzz.ttt  

5.3 Installing Software Updates

To make sure that RT-11 operates correctly, you must install updated soft-
ware modules at this point. Updated modules are critical to system or com-
ponent operation. They correct software errors discovered since the last
release, and may add new functions to the operating system.

Updated software modules are distributed in update kits. Each kit contains
complete software module replacements. Software corrections will already
have been installed in the new modules and fully tested. Your only require-
ment is to copy the new modules from the distribution medium to your
system, using the procedures described in the RT-11 Update User's Guide.
Updates are distributed on the same medium on which you received the
base system and contain modules for both base system and layered prod-
ucts.

When you submit an SPR, the RT-11 Software Dispatch Review will report
serious problems and may contain suggested solutions to problems, until
the module containing the error is corrected by an update kit.

For a complete description of how to update your software, refer to the

5.4 Creating the Working System from Chosen Components

Once you have chosen your system components (Section 2.3) and have
planned the best arrangement of them on disks (Section 2.4), you can create
the working system.

If you are using an RC25 disk, you create your working system by deleting
selected components from the backup disk. Use the following command to
delete unnecessary components:

`\DELETE/SYSTEM/QUERY DUI:\$`  
Files deleted?  
DUI:aaaaaaaa.ttt? Y\$  
DUI:bbbbbbbb.ttt? N\$  
(to delete a specific file)  
(to keep a specific file)  
(and so on)
For all other disks, you create your working system by copying selected components to one or more disks.

Mount a blank disk in another disk unit and initialize it. If your disk is an RK05, format the disk before you initialize it.

**RK05 disk**

```
> FORMAT RKn:     E7
RKn:/FORMAT-Are you sure? Y E7
?FORMAT-I-Formatting complete

> INITIALIZE/BADBLOCKS RKn: E7
RKn:/Initialize: Are you sure? Y E7
?DUP-I-No bad blocks detected RKn:
```

**Other type disk**

```
> INITIALIZE/BADBLOCKS xxn: E7
xxn:/Initialize: Are you sure? Y E7
?DUP-I-No bad blocks detected xxn:
```

or

```
> INITIALIZE/REPLACE xxn: E7
xxn:/Initialize: Are you sure? Y E7
?DUP-I-No bad blocks detected xxn:
```

Copy the files you have selected from the backup disk to the disk that will become your working system disk. If you use the following command, RT–11 queries you about all the files on the backup disk, and you can choose the files it copies. In the command, yyn is the device name and unit number of your backup disk, and xxn is the device name and unit number of the disk to be your working system disk.

```
> COPY/SYSTEM/QUERY yyn: xxn: E7
Files copied:
yyn:aaaaaa.ttt to xxn:aaaaaa.ttt? Y E7 (to include a specific file)
yyn:bbbbbb.ttt to xxn:bbbbbb.ttt? N E7 (to exclude a specific file)
(and so on)
```

If you wish to create multiple working system disks, mount one blank disk for each working system you wish to create, format blank RK05 disks, and initialize the disks. Then, use the preceding COPY command to copy selected files to the appropriate disks.

### 5.5 Installing the Bootstrap on the Disk

Once you have created your working system disk(s), you need to install the bootstrap on all disks, except RC25. In the following command, aa is BL, SJ, FB, or XM.

```
> COPY/BOOT xxn:RT11aa.SYS xxn: E7
```
In this command, you identify the device on which the monitor that contains the bootstrap information resides, the name of that monitor file (RT11BL, RT11SJ, RT11FB, or RT11XM), and the device on which you need to install the bootstrap. This command copies bootstrap information from the monitor file to blocks 0 and 2 through 5 of the same disk.

Remove the backup disk from Unit 0 and store it for future updates. Remove the working system disk from Unit n and mount it on Unit 0.

Use the hardware bootstrap to boot the new working system disk.

```
RT-11aa V05.xx
(Followed by any start-up file commands.)
```

### 5.6 Customizing the System

You may want to make certain customizations (described in Section 2.7) to the distributed RT-11 components. At this point, perform the procedures to implement any of these software customizations. Table 1–3 summarizes the available customizations and directs you to the section in Chapter 2 that describes a particular customization and the procedure for implementing it.

**NOTE**

Later, you can perform the system generation process to implement additional customizations. (See Section 5.10.)

### 5.7 Compressing the Disk

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

Continue to run RT-11 from Unit 0 (Unit 1 for RC25 disk), and use the SQUEEZE command to compress free space. (The disk must be write-enabled.) The squeeze operation does not move files with the .BAD file type.

```
SQUEEZE xnx:
xxn:/Squeeze? Are you sure? Y
RT-11aa V05.xx
(Followed by any start-up file commands.)
```

The system automatically reboots when you compress a system disk.
5.8 Preserving the Working System

Once you build a satisfactory working system, DIGITAL recommends that you preserve it on the backup medium of your choice. The following sections describe backing up on disk. If you want to back up the system on another device, the procedure may appear elsewhere in this manual; refer to the table of contents.

Insert a blank disk (write-enabled) in another disk unit and initialize the blank disk. If your disk is an RK05, format the disk before initializing it. In the following commands, xxn is the device name and unit number for the backup working system disk.

**RK05 disk**

```
,FORMAT RKn:RD
RKn:/FORMAT-Are you sure? YRD
?FORMAT-I-Formatting complete

,INITIALIZE/BADBLOCKS RKn:RD
RKn:/Initialize; Are you sure? YRD
?DUP-I-No bad blocks detected RKn:
```

**Other type disk**

```
,INITIALIZE/BADBLOCKS xxn:RD
xxn:/Initialize; Are you sure? YRD
?DUP-I-No bad blocks detected xxn:
```

or

```
,INITIALIZE/REPLACE xxn:RD
xxn:/Initialize; Are you sure? YRD
?DUP-I-No bad blocks detected xxn:
```

Copy all the files in your working system. In the following command, yyn is the device name and unit number for your working system disk, and xxn is the device name and unit number for the backup working system disk.

```
,SQUEEZE/OUTPUT:xxn yyn:RD
```

Copy the bootstrap to the backup working system disk. In the following command, aa is BL, SJ, FB, or XM.

```
,COPY/BOOT xxn:RT11aa;SYS xxn:RD
```
Store the backup working system disk. If you ever need to restore the working system, you can make a copy of the backup working system disk.

5.9 Testing the Working System

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

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

SWAP.SYS
RT11FB.SYS
xx.SYS (system device handler)
LP.SYS
EDIT.SAV
MACRO.SAV
SYSMAC.SML
LINK.SAV
PIP.SAV
DUP.SAV
DIR.SAV
DEMOBG.MAC
DEMOFG.MAC

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

DIRECTORY/BRIEF/COLUMNS:1 SY:R
                   dd-mm-yy
SWAP .SYS
RT11FB .SYS
LP .SYS
RK .SYS
EDIT .SAV
MACRO .SAV
SYSMAC .SML
LINK .SAV
PIP .SAV

xxxx Files, bbb Blocks
fff Free blocks

NOTE

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

5–10 Installing a System Distributed on Hard Disk to Run on Hard Disk
Before you can execute the background and foreground demonstration programs, you must first edit, assemble, and link the background program, DEMO BG.MAC, and you must assemble and link the foreground program, DEMO FG.MAC.

5.9.1 Preparing the Background Demonstration Program

5.9.1.1 Edit the Background Demonstration Program — Use a text editor, for example, EDIT.SAV, to modify the background demonstration program, DEMO BG.MAC. One of the output lines in the program is preceded by a semicolon, which makes the line a comment field. The semicolon prevents the line from being printed; thus, it must be deleted from that line. If DEMO BG.MAC is a protected file, remove the protection before making the edits (UNPROTECT SY:DEMO BG.MAC).

.EDIT SY:DEMO BG.MAC
*FILE ASCII
*OAD(15:15)
*EX(15:15)
.

5.9.1.2 Assemble the Background Demonstration Program — The background program, DEMO BG.MAC, is an assembly language source file; it must be assembled and linked before you can execute it. To assemble DEMO BG.MAC and obtain a listing, make sure that your configuration has a line printer that is on-line and ready.

.ASSIGN LP: LST
.

NOTE

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

.ASSIGN TT: LST
.

Assemble DEMO BG.MAC as follows:

.MACRO/LIST:LST: DEMO BG
(See Figure 5–3.)

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

.RENAME SY:DEMO BG.BAK SY:DEMO BG.MAC
.

Installing a System Distributed on Hard Disk to Run on Hard Disk 5–11
5.9.1.3 Link the Background Demonstration Program — Link the program DEMOBG.OBJ to produce an executable background program, DEMOBG.SAV.

.LINK DEMOBG

5.9.1.4 Run the Background Demonstration Program — Run the program DEMOBG to check the results of the first exercise.

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

If you did not delete the semicolon character, the last line will not print. Return to the monitor by typing two CTRL/Cs.
If you incorrectly edited the file, you can repeat this exercise, although you can continue without correcting the file. However, if you want to repeat the exercise, begin by using the backup demonstration program.

```
.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC
```

Then, repeat the editing procedure.

### 5.9.2 Preparing the Foreground Demonstration Program

To execute the foreground program, you must assemble the program DEMOFG.MAC, link it for the foreground, and execute it in conjunction with DEMOBG.SAV. DEMOFG.MAC is a small foreground program that sends a message every 2 seconds to DEMOBG (running in the background), telling it to ring the terminal bell. DEMOBG recognizes these messages and rings the bell once for each message sent.

Although DEMOFG is always active, sending messages to the background every 2 seconds, this exercise can execute other programs in the background besides DEMOBG. The circuit is complete and messages are successfully received and honored only when DEMOBG is active. During those periods when DEMOBG is not running, DEMOFG enters the messages in the monitor message queue. Once you restart DEMOBG in the background, the system immediately releases all the messages queued since the last forced exit, resulting in many successive bell rings. When the queue is empty, the normal send/receive cycle resumes and the bell rings every 2 seconds as each current message is sent and honored.

#### 5.9.2.1 Assemble the Foreground Demonstration Program —

The foreground demonstration program, DEMOFG.MAC, is an assembly language source file; it must be assembled and linked before you can use it. Assemble DEMOFG.MAC as follows:

```
.MACRO/LIST: LST: DEMOFG
```

The output resulting from this MACRO command is an object file called DEMOFG.OBJ. This file resides on your system volume.

#### 5.9.2.2 Link the Foreground Demonstration Program —

You must link the DEMOFG.OBJ file to produce an executable program. Use the /FOREGROUND option to produce the load module DEMOFG.REL. The .REL file type signifies to the system that the file is a foreground program and is to be run as a priority job.

```
.LINK/FOREGROUND DEMOFG
```
5.9.2.3 Run the Foreground and Background Demonstration Programs — Type the following command to load and start DEMOFG.REL as the foreground job.

.FRUN DEMOFG.REL

FOREGROUND DEMONSTRATION PROGRAM
Sends a message to the background program DEMOBG every 2 seconds, telling it to ring the bell.

DEMOCR.REL is now running and queuing the message for DEMOBG every 2 seconds. Now execute DEMOBG.SAV in the background and receive the messages.

.RUN DEMOBG.REL

(The bell rings quickly several times, then once every 2 seconds.)

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

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

.DIRECTORY.REL

(The bell stops ringing.)

.RER

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

.RER

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

.RUN DEMOBG.REL

(The bell rings several times in rapid succession, then rings once every 2 seconds.)

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

.RER

(The bell stops ringing.)
Now, stop the foreground program and remove it from memory.

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

5.10 Performing the System Generation Process

If you have decided that you need RT-11 features that are available only if you generate your own monitor(s) and handlers, perform the system generation process at this point. You should have thoroughly studied Chapter 1 to make this decision and to establish that you can perform the system generation process on your particular hardware configuration. Read the RT-11 System Generation Guide for guidance in planning and performing system generation.
Chapter 6
Installing a System Distributed on Hard Disk to Run on a Small Device

If RT–11 was distributed to you on hard disk (RL01/RL02, RK05, or RC25), and you intend to build a system to run on a small device (RX01, RX02, or RX50), perform the procedures described in this chapter. These procedures assume that your configuration includes two hard disk drives (not necessarily of the same type).

NOTE

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

To install your system, perform the steps summarized in the following list. Sections 6.1 through 6.10 describe the procedures involved in each step. Figure 6–1 shows the backup disk and diskettes you create when you install RT–11.

1. Bootstrap the distribution disk.
2. Preserve the distribution disk.
3. Install software updates.
4. Create the working system from chosen components.
5. Install the bootstrap on diskettes that need to be bootable.
6. Customize the system.
7. Compress each diskette.
8. Preserve the working system.
9. Test the working system.

10. If appropriate, perform the system generation process.

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

Figure 6-1: Sample Backup Disk and Diskettes

![Diagram showing backup and distribution process]

6.1 Bootstrapping the Distribution Disk

The first procedure you perform when installing RT–11 is to bootstrap the distribution disk.

Begin by making sure that the processor is powered up but not running. Mount the distribution disk (write-protected) in disk Unit 0. The device name is DL, RK, or DU, depending on the type of disk device. Use the hardware bootstrap to boot the disk. (If your configuration does not include a hardware bootstrap, refer to Appendix B for the toggle-in software bootstrap.)

RL02 and RC25 disks are normally installed manually. If you successfully bootstrap an RL02 or an RC25 disk, RT–11 responds with the following dialog and installation verification message. The dialog gives you the choice of installing RT–11 manually or by using the automatic installation procedure.

If you successfully bootstrap an RL01 or an RK05 disk, RT–11 responds with the following installation verification message.
Welcome to RT-11 Version 5.

You have bootstrapped the RT-11 Distribution Disk. Use this disk to install your RT-11 system, then store it in a safe place.

RT-11 V5 provides an automatic installation procedure which will back up your distribution disk and build a working system disk which should be used for your work with RT-11. This working system disk will only contain the RT-11 operating system. After the RT-11 installation is complete, follow the installation instructions packaged with any optional languages or utility software which you will be using.

Press the "RETURN" key when ready to continue.

You can choose to install RT-11 manually. This procedure is described in the RT-11 Installation Guide.

If you are a new user of RT-11, DIGITAL highly recommends that you use the automatic installation procedure.

Do you want to use the automatic installation procedure?
(Type YES or NO and press the "RETURN" key): NO

The standard RT-11 monitor bootstrap (RT11FB) will now be copied to this disk and this system disk will be rebooted. Please refer to the RT-11 Installation Guide for assistance while manually installing RT-11 Version 5.

Press the "RETURN" key when ready to continue.

RT-11FB V05.xx

.Type V5USER.TXT

RT-11 V5.xx

Installation of RT-11 ...(followed by V5USER.TXT message)....

Now you can create a start-up file containing any commands or sequence of commands that you would normally want to execute at the start of each terminal session.

You can update your start-up file at any time to change, add, or delete commands. When you first create the file, or after you update it, you can execute it with the @ command, so that the commands it contains become effective.

6.2 Preserving the Distribution Disk

The first procedure you perform with the running RT-11 system is to copy the distribution disk for backup, as a safety measure in case of machine failure or human error. You can then use the backup copy of the distribution to build your working system.

Mount a blank disk in the other disk drive, if necessary. (The disk in Unit 1 of RC25 devices is permanently mounted.) If the disk is an RK05 disk, format the disk, and then initialize it and cover any bad blocks on it.
If the disk is another type, initialize the disk and replace bad blocks (using the command INITIALIZE/REPLACE) or cover bad blocks (using the command INITIALIZE/BADBLOCKS). You have a choice of replacing or covering bad blocks if your disk is an RL01, RL02, RK06, or RK07. (Refer to the RT-11 System User's Guide.) If your disk is another type, for example, RC25, use INITIALIZE/BADBLOCKS to cover any bad blocks. If the disk contains bad blocks, the ?DUP–W–Bad blocks detected nnnnn message appears on the terminal, but you can still use the disk. In the commands, xx is the permanent device name, and n is the unit number for your disk.

**RK05 disk**

```
FORM RKn:RE
RKn:/FORMAT-Are you sure? YRE
?FORMAT-I-Formatting complete

INITIALIZE/BADBLOCKS RKn:RE
RKn:/Initialize Are you sure? YRE
?DUP-I-No bad blocks detected RKn:
```

**Other type disk**

```
INITIALIZE/BADBLOCKS xxn:RE
xxn:/Initialize Are you sure? YRE
?DUP-I-No bad blocks detected xxn:
```

or

```
INITIALIZE/REPLACE xxn:RE
xxn:/Initialize Are you sure? YRE
?DUP-I-No bad blocks detected xxn:
```

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

The next step in the preservation process is to copy all the files from the distribution disk to the initialized blank backup disk.

The following command transfers files from the distribution disk to the backup disk and consolidates all the empty space at the end of the disk. In the command, yyn is the device name and unit number of your distribution disk, and xxn is the device name and unit number of your backup disk.

```
SQUEEZE/OUTPUT:xxn: yyn:RE
```

**NOTE**

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

Next, copy the bootstrap to the backup disk.

```
COPY/BOOT x:x:RT11FB,SYS x:x:
```

Now halt the processor, remove the distribution disk from Unit 0, and store it. For RC25 devices, use the hardware bootstrap to boot the backup disk on disk Unit 1. For all other devices, mount the backup disk on Unit 0, and use the hardware bootstrap to boot the backup disk.

```
RT-11FB V05.xx
(Followed by any start-up file commands.)
```

**NOTE**

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

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

```
UNPROTECT/SYSTEM **
Files unprotected:
DK:aaaaaaa.ttt
DK:bbbbbbb.ttt
DK:ccccccc.ttt
DK:ddddddd.ttt

DK:zzzzzzz.ttt
```

6.3 Installing Software Updates

To make sure that RT-11 operates correctly, you must install updated software modules at this point. Updated modules are critical to system or component operation. They correct software errors discovered since the last release, and may add new functions to the operating system.

Updated software modules are distributed in update kits. Each kit contains complete software module replacements. Software corrections will already have been installed in the new modules and fully tested. Your only require-
ment is to copy the new modules from the distribution medium to your system, using the procedures described in the RT–11 Update User's Guide. Updates are distributed on the same medium on which you received the base system and contain modules for both base system and layered products.

When you submit an SPR, the RT–11 Software Dispatch Review will report serious problems and may contain suggested solutions to problems, until the module containing the error is corrected by an update kit.

For a complete description of how to update your software, refer to the RT–11 Update User's Guide.

6.4 Creating the Working System from Chosen Components

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

6.4.1 Initializing RX50 Diskettes

If you wish to create your working system on RX50 diskettes, insert a blank RX50 diskette in Unit 1 or 2 so you can initialize it. You can use the INITIALIZE command with the /BADBLOCKS option to initialize the diskette and to detect any bad blocks that may be on it. If the diskette contains bad blocks, the ?DUP–W–Bad blocks detected nnnnnn message appears on the terminal. In the command, x is the device unit number.

NOTE

RX50 diskettes are not formattable. DIGITAL recommends that you use only diskettes that have no bad blocks. To ascertain whether an already initialized diskette has bad blocks, use the command DIRECTORY/BAD DUx:. If bad blocks exist on a diskette, copy the contents of the diskette to an error-free diskette, and dispose of the diskette with bad blocks.

...IN INITIALIZE/BADBLOCKS DUx:#
DUx:/Initialize; Are you sure? Y#
?DUP–1–No bad blocks detected DUx:

There may be a delay of up to 1 minute while the system scans the diskette for bad blocks and creates a new directory. The monitor dot appears when this process is complete.

...

Now, remove the newly initialized diskette and initialize an adequate number of blank diskettes, leaving one initialized, blank, write-enabled (write-protect notch uncovered) diskette inserted in Unit 1 or 2.
6.4.2 Formatting and Initializing RX01/RX02 Diskettes

If you plan to create your working system on RX01 diskettes, initialize a number of blank RX01 diskettes. For RX02 diskettes, format and initialize the diskettes. Insert a blank diskette in Unit 1. Diskettes are available in single-density but not double-density format. Therefore, to create RX02 diskettes, you must format all your blank diskettes as double-density diskettes. You can use the FORMAT utility program to format the blank diskette and the INITIALIZE command to initialize it. Use the /BADBLOCKS option with INITIALIZE to cover any bad blocks that may be on your diskette. If the diskette contains bad blocks, the ?DUP-W-Bad blocks detected nnnnnn message appears on the terminal.

NOTE

DIGITAL recommends that you use only diskettes that do not have bad blocks when you build a working system. To ascertain whether an already initialized diskette has bad blocks, use the command DIRECTORY/BAD DYN:. You can use diskettes with bad blocks later, for working or data volumes.

For RX02 diskettes:

.COMMAND FORMAT D1:  
D1:/FORMAT-Are you sure? Y  
?FORMAT-1-Formatting complete  

.COMMAND INITIALIZE/BADBLOCKS D1:  
D1:/Initialize! Are you sure? Y  
?DUP-I-No bad blocks detected D1:

For RX01 diskettes, just initialize the diskette:

.COMMAND INITIALIZE/BADBLOCKS D1:  
D1:/Initialize! Are you sure? Y  
?DUP-I-No bad blocks detected D1:

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

Now, remove the newly formatted and initialized diskette. Repeat the process to create as many initialized blank diskettes as you need for the system that you have planned. Leave one diskette in Unit 1.

6.4.3 Copying the Selected Files onto Diskette

Use the COPY command with the /SYSTEM option to copy selected files from the backup disk to the diskettes that will become your working system diskettes. The /SYSTEM option is required for copying .SYS files only if
wildcards are used in the input file specification. In the command, yyn is the device name and unit number of your backup disk, and xxn is the device name and unit numbers of the diskettes to be your working system diskettes.

}.COPY yyn:filnam,typ xxn:filnam,typ

You can use the following command to avoid typing numerous file specifications. RT-11 queries you about all the files on the diskette, and you choose the files it copies.

}.COPY/SYSTEM/QUERY yyn: xxn: @
Files copied:
yyn:aaaaaa.ttt to xxn:aaaaaa.ttt? Y@T (to include a specific file)
yyn:bbbbbb.ttt to xxn:bbbbbb.ttt? N@T (to exclude a specific file)
(and so on)

When you have copied all the files you planned for the working system diskette, label the diskette "RT-11 Working System V05 x/y" (where x is the diskette number, and y is the number of diskettes in your working system).

6.5 Installing the Bootstrap on Any Diskettes That Need to Be Bootable

Once you have created your working system diskettes, you need to install the bootstrap on any diskettes that must be bootable (that is, that you can use as system diskettes). Generally, any diskette that includes a monitor file and system device handler should be bootable (but remember that the diskette would need SWAP.SYS and, for the SJ monitor, TT.SYS).

Insert the diskette on which you need to install the bootstrap. In the command, xxn represents the device name and unit number of your diskette, and aa is BL, SJ, FB, or XM.

}.COPY/BOOT xxn:RT11aa.SYS xxn: @

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

Then, insert working system diskette 1 in Unit 0, and boot your working system. Use the hardware bootstrap if possible. If not, type the following command; xxn represents the diskette to boot.
.BOOT x:x:8a

RT-11 aa VO5.xx

(Followed by any start-up file commands.)

Store your updated backup distribution disk for future updates.

6.6 Customizing the System

You may want to make certain customizations (described in Section 2.7) to the distributed RT-11 components. At this point, perform the procedures to implement any of these software customizations. Table 1–3 summarizes the available customizations and directs you to the section in Chapter 2 that describes a particular customization and the procedure for implementing it.

NOTE

Later, you can perform the system generation process to implement additional customizations. (See Section 6.10.)

6.7 Compressing Each Diskette

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

Continue to run RT-11 from Unit 0, and use the SQUEEZE command to compress free space. The squeeze operation does not move files with the .BAD file type.

,SQUEEZE x:x:C:RET
xx0:/Squeeze: Are you sure? Y:RET

There is a delay while the squeeze operation takes place.

RT-11 aa VO5.xx

(Followed by any start-up file commands.)

The system automatically reboots when you compress a system diskette.

Then, insert the next diskette that you need to compress in Unit 1.

,SQUEEZE x:x1:C:RET
xx1:/Squeeze: Are you sure? Y:RET

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

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

6.8 Preserving the Working System

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

Use the following command to protect all the files on the system diskette:

```
;PROTECT/SYSTEM *.*
Files Protected:
DK:aaaaaaaa.ttt
DK:bbbbbbbb.ttt
DK:cccccccc.ttt
DK:dddddddd.ttt
.
.
DK:zzzzzzz.ttt
```

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

```
;PROTECT/SYSTEM xx1:*.*
Files Protected:
xx1:aaaaaaaa.ttt
xx1:bbbbbbbb.ttt
xx1:cccccccc.ttt
xx1:dddddddd.ttt
.
.
xx1:zzzzzzz.ttt
```

Next, copy the working system to backup diskettes. Insert a blank diskette in Unit 1 with RT-11 still booted from Unit 0. Format (if you are using RX02 diskettes) and initialize the appropriate number of diskettes.

Copy all the files in your working system. You can use the SQUEEZE/OUTPUT:xx1: xx0: command to copy any bootable diskettes. Remember that you must use SQUEEZE/WAIT/OUTPUT:xx1: xx0: and change diskettes to copy the diskettes that are not bootable. (Refer to Section 3.4.) Also remember to copy the bootstrap to any diskettes that need to be bootable.

Store the backup working system diskettes. If you ever need to restore the working system, you can make copies of the backup working system diskettes.

6–10 Installing a System Distributed on Hard Disk to Run on a Small Device
6.9 Testing the Working System

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

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

SWAP.SYS
RT11FB.SYS
xx.SYS (system device handler)
LP.SYS
EDIT.SAV
MACRO.SAV
SYSMAC.SML
LINK.SAV
PIP.SAV
DUP.SAV
DIR.SAV
DEMOBG.MAC
DEMOFG.MAC

Insert a blank diskette in Unit 1. If you are using RX02 diskettes, format the disk first.

,.FORMAT DY1:RE
DY1:/FORMAT- Are you sure? Y
?FORMAT-I-Formatting complete

Initialize all diskettes, and check for bad blocks.

,.INITIALIZE/BADBLOCKS xx1:RE
xx1:/Initialize: Are you sure? Y
?DUP-I-No bad blocks detected xx1:
,.ASSIGN xx1: OK:RE

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

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

Before you can execute the background and foreground demonstration programs, you must first edit, assemble, and link the background program, DEMOBG.MAC, and you must assemble and link the foreground program, DEMOFG.MAC.

6.9.1 Preparing the Background Demonstration Program

6.9.1.1 Edit the Background Demonstration Program — Use a text editor, for example, EDIT.SAV, to modify the background demonstration program, DEMOBG.MAC. One of the output lines in the program is preceded by a semicolon, which makes the line a comment field. The semicolon prevents the line from being printed; thus, it must be deleted from that line. If DEMOBG.MAC is a protected file, remove the protection before making the edits (UNPROTECT SY:DEMOBG.MAC).

**.EDIT SY:DEMOBG.MAC**
*FILE.ASCII**
*OAD**
*EX**

6.9.1.2 Assemble the Background Demonstration Program — The background program, DEMOBG.MAC, is an assembly language source file; it must be assembled and linked before you can execute it. To assemble DEMOBG.MAC and obtain a listing, make sure that your configuration has a line printer that is on-line and ready.

**.ASSIGN LP: LST:**
NOTE
If your configuration does not include a line printer, use the console terminal.

ASSIGN TT: LST:

Assemble DEMOBG.MAC as follows:

MACRO LIST: LST: SY: DEMOBG

(See Figure 6-2.)

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

RENAME SY:DEMOBGBAK SY:DEMOBGMAC

Figure 6-2: DEMOBG Assembly Listing

DEMOB MACRO 085.01b Friday 09-Dec-83 10:16 Page 1

1 .TITLE DEMOBG
2 I DEMONSTRATION PROGRAM TO PRINT DEMONSTRATION MESSAGE, THEN
3 I RING BELL IF FD JOB SENDS A MESSAGE.
4
5 .NCALL .RCVDC:PRINT
6 0 000000
7 0 000003
8 0 000042 00077
9
10 .NCALL .RCVDC:PRINT
11 START: .RCVDC AREA:BUFFER:4000$:MSGIN:IPRT REQUEST FOR MESSAGE
12 .PRINT MSIN:IPRT DEMONSTRATION MESSAGE
13 BR
14 IAM LOOP
15
16 .NCALL .RCVDC:PRINT
17 1 COMPLETION ROUTINE ENTERED WHEN FD SENDS MESSAGE
18 MSGIN: .PRINT BELL
19 .RCVDC AREA:BUFFER:4000$:MSGIN:IPRT ANOTHER MESSAGE REQUESTED
20 RETURN IAM AND RETURN FROM COMPLETION ROUTINE
21
22 .NCALL .RCVDC:PRINT
23 ASCII MESSAGES
24 .MLIST BEX
25 AREA: .BLKW 4 I ERT ARGUMENT AREA
26 .BLKW 4 BUFFER: I RCVDC MESSAGE AREA
27 0 000000
28 0 000000

DEMOB MACRO 085.01b Friday 09-Dec-83 10:16 Page 1-1
Symbol table
AREA 000000: BUFFER 000000: ...VI = 000000 ...V2 = 000027
1000100: MSG 000000: .ASS 000000 000 (Rw-1:DBL:ABS:DV)
0000000: .ASS 000000 000 (Rw-1:DLR:ABS:DV)
Errors detected: 0

### Assembler statistics
Work file read: 0
Work file written: 0
Size of work file: 10KB Words ( 36 Pages)
Size of core pool: 16128 Words ( 63 Pages)
Operating system: NT-11
Elapsed time: 00:00:00:45
DRI:DEMOB:DR:DEMOB:DR:DEMOB

Installing a System Distributed on Hard Disk to Run on a Small Device 6-13
6.9.1.3 Link the Background Demonstration Program — Link the program DEMOOG.OBJ to produce an executable background program, DEMOOG.SAV.

\texttt{\textasciitilde LINK DEMOOG\$\textasciitilde}

6.9.1.4 Run the Background Demonstration Program — Run the program DEMOOG to check the results of the first exercise.

\texttt{\textasciitilde RUN DEMOOG\$\textasciitilde}
\texttt{RT-11 DEMONSTRATION PROGRAM}
\texttt{IF INCORRECTLY EDITED, THIS IS THE LAST LINE.}
\texttt{WELL DONE.}

If you did not delete the semicolon character, the last line will not print. Return to the monitor by typing two CTRL/Cs.

\texttt{CTRL\textasciitilde C}
\texttt{CTRL\textasciitilde C}

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

\texttt{\textasciitilde RENAME DEMOOG.BAK DEMOOG.MAC\$\textasciitilde}

Then, repeat the editing procedure.

6.9.2 Preparing the Foreground Demonstration Program

To execute the foreground program, you must assemble the program DEMOFG.MAC, link it for the foreground, and execute it in conjunction with DEMOOG.SAV. DEMOFG.MAC is a small foreground program that sends a message every 2 seconds to DEMOOG (running in the background), telling it to ring the terminal bell. DEMOOG recognizes these messages and rings the bell once for each message sent.

Although DEMOFG is always active, sending messages to the background every 2 seconds, this exercise can execute other programs in the background besides DEMOOG. The circuit is complete and messages are successfully received and honored only when DEMOOG is active. During those periods when DEMOOG is not running, DEMOFG enters the messages in the monitor message queue. Once you restart DEMOOG in the background, the system immediately releases all the messages queued since the last forced exit, resulting in many successive bell rings. When the queue is empty, the normal send/receive cycle resumes and the bell rings every 2 seconds as each current message is sent and honored.
6.9.2.1 **Assemble the Foreground Demonstration Program** — The foreground demonstration program, DEMOFG.MAC, is an assembly language source file; it must be assembled and linked before you can use it. Assemble DEMOFG.MAC as follows:

```
.MACRO/LIST:LST: SY:DEMOFG@E
```

The output resulting from this MACRO command is an object file called DEMOFG.OBJ. This file resides on your data volume (DK).

6.9.2.2 **Link the Foreground Demonstration Program** — You must link the DEMOFG.OBJ file to produce an executable program. Use the /FOREGROUND option to produce the load module DEMOFG.REL. The .REL file type signifies to the system that the file is a foreground program and is to be run as a priority job.

```
.LINK/FOREGROUND DEMOFG@E
```

6.9.2.3 **Run the Foreground and Background Demonstration Programs** — Type the following command to load and start DEMOFG.REL as the foreground job.

```
.RUN DEMOFG@E
```

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

DEMOFG.REL is now running and queuing the message for DEMOBG every 2 seconds. Now execute DEMOBG.SAV in the background and receive the messages.

```
.RUN DEMOBG@E
```

(The bell rings quickly several times, then once every 2 seconds.)

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

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

```
CTR C
CTR C
```

(The bell stops ringing.)

```
DIRECTORY@E
```

The directory of the device DK prints on the terminal.)
Rerun DEMOBG to collect all the foreground messages queued while the directory was printing.

. RUN DEMO BG

(The bell rings several times in rapid succession, then rings once every 2 seconds.)

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

. RUN <
. RUN <
.
(The bell stops ringing.)

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

. RUN <
F >
. RUN <
. RUN <
B >
UNLOAD F
.

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

6.10 Performing the System Generation Process

If you have decided that you need RT-11 features that are available only if you generate your own monitor(s) and handlers, perform the system generation process at this point. You should have thoroughly studied Chapter 1 to make this decision and to establish that you can perform the system generation process on your particular hardware configuration. Read the RT-11 System Generation Guide for guidance in planning and performing system generation.
Chapter 7
Installing a System Distributed on RX02 to Run on RX02

If RT–11 was distributed to you on RX02 double-density diskette, and you intend to build a system to run on double-density diskette, perform the procedures described in this chapter.

Your distribution kit contains two exact copies of the RT–11 operating system on RX02 diskettes. Store one copy of the RX02 diskettes in a safe place, and do not modify their contents; they provide a backup (master) copy of the distributed RT–11 operating system.

Store the second copy of diskettes in a safe place after installing software updates and creating your working system diskettes. Use these diskettes to create another working system. If these diskettes are ever damaged, use the master copy to reproduce them.

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

To install your system, perform the steps summarized in the following list. Sections 7.1 through 7.9 describe the procedures involved in each step. Figure 7–1 shows the diskettes you create when you install RT–11.

1. Bootstrap the distribution diskette.
2. Install software updates.
3. Create the working system from chosen components.
4. Install the bootstrap on diskettes that need to be bootable.
5. Customize the system.
6. Compress each diskette.
7. Preserve the working system.
8. Test the working system.
9. If appropriate, perform the system generation process.

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

Figure 7–1: Sample Backup Diskettes

7.1 Bootstrapping the Distribution Diskette

The first procedure you perform when installing RT–11 is to bootstrap distribution diskette 1.

Begin by making sure that the processor is powered up but not running. Insert the distribution diskette labeled 1 in Unit 0 (which has the physical device name DY0:). Use the hardware bootstrap to boot the diskette. (If your configuration does not include a hardware bootstrap, see Appendix B for toggle-in software bootstraps.)

RT–11 should respond with the following message if you have successfully bootstrapped the diskette:
Installation of RT-11 ... (followed by V5USER.TXT message) ....

After you boot the system, you can create a start-up file containing any commands or sequence of commands that you would normally want to execute at the start of each terminal session.

You can update your start-up file at any time to change, add, or delete commands. When you first create the file, or after you update it, you can execute it with the @ command, so that the commands it contains become effective.

Next, remove the protection from all the files on the distribution diskettes. The files on the distribution diskettes have been protected to prevent you from accidentally deleting them. (Refer to the RT-11 System User’s Guide for a description of file protection.) (Note the P that prints next to the file size in the directory.) For the rest of these procedures, you need to remove the protection from these diskettes. Type the following command to remove it from the files on distribution diskette 1:

```
.UNPROTECT/SYSTEM *. **R
Files unprotected:
DK: aaaaaa.ttt
DK: bbbbbbb.ttt
DK: ccccccc.ttt
DK: ddddddd.ttt

DK: zzzzzzz.ttt
```

Insert distribution diskette 2 in Unit 1 and type the following command:

```
.UNPROTECT/SYSTEM DY1:* **R
Files unprotected:
DY1: aaaaaa.ttt
DY1: bbbbbbb.ttt
DY1: ccccccc.ttt
DY1: ddddddd.ttt

DY1: zzzzzzz.ttt
```
Replace distribution diskette 2 with distribution diskette 3 in Unit 1 and repeat this command. In the same way, remove protection from the files on the remaining distribution diskettes.

7.2 Installing Software Updates

To make sure that RT-11 operates correctly, you must install updated software modules at this point. Updated modules are critical to system or component operation. They correct software errors discovered since the last release, and may add new functions to the operating system.

Updated software modules are distributed in update kits. Each kit contains complete software module replacements. Software corrections will already have been installed in the new modules and fully tested. Your only requirement is to copy the new modules from the distribution medium to your system, using the procedures described in the RT-11 Update User's Guide. Updates are distributed on the same medium on which you received the base system and contain modules for both base system and layered products.

When you submit an SPR, the RT-11 Software Dispatch Review will report serious problems and may contain suggested solutions to problems, until the module containing the error is corrected by an update kit.

For a complete description of how to update your software, refer to the RT-11 Update User's Guide.

7.3 Creating the Working System from Chosen Components

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

Start by formatting and initializing a number of blank diskettes. Insert a blank diskette in Unit 1 so you can format and initialize it. Diskettes are available in single-density but not double-density format. Therefore, you must reformat all your blank diskettes as double-density diskettes. You can use the FORMAT utility program, which is included on distribution diskette 1, to format the blank diskettes and the INITIALIZE command to initialize them. Use the /BADBLOCKS option with INITIALIZE to cover any bad blocks that may be on your diskettes. If a diskette contains bad blocks, the ?DUP=W-Bad blocks detected nnnnnn message appears on the terminal.

NOTE

DIGITAL recommends that you use only diskettes that do not have bad blocks when you build a working system. To ascertain whether an already initialized diskette has bad blocks, use the command DIRECTORY/BAD DYn:. You can use diskettes with bad blocks later for working or data volumes.
The system scans the diskette for bad blocks and creates a new directory. The monitor dot appears when this process is complete.

Repeat the process to create as many initialized blank diskettes as you need for the system that you have planned, leaving one initialized, blank diskette in Unit 1.

Then, use the COPY command with the /SYSTEM option to copy selected files from distribution diskette 1 to the diskette that becomes your working system diskette. The /SYSTEM option is required for copying .SYS files only if wildcards are used in input file specifications.

`COPY DYO:filnam.typ DYO:filnam.typ`

You can use the following command to avoid typing numerous file specifications. RT-11 queries you about all the files on the diskette, and you choose the files it copies.

`COPY/SYSTEM/QUERY DYO: DYO:`

Files copied:
DYO:aaaaaa.ttt to DYO:aaaaaa.ttt? Y (to include a specific file)
DYO:bbbbbb.ttt to DYO:bbbbbb.ttt? N (to exclude a specific file)
(and so on)

To copy files from nonbootable diskettes, you have to alternate diskettes.

Use the SET command to set the USR to NOSWAP.

`SET USR NOSWAP`

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

`COPY/WAIT DYO:filnam.typ DYO:filnam.typ`

Mount input volume in DYO: Continue?

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

Y

Mount output volume in DYO: Continue?
Replace the system diskette in Unit 0 with the diskette to which you want to copy filnam.typ.

Mount system volume in DY0: Continue?

Replace the diskette in Unit 0 with backup diskette 1.

Repeat this procedure to copy all the files you planned for the working system diskette. When you have copied all the files, label the diskette "RT-11 Working System V05 1". Repeat these procedures to create the other diskettes in the working system.

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

7.4 Installing the Bootstrap on Any Diskettes That Need to Be Bootable

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

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

COPY/BOOT DY1:RT11aa.SYS DY1:

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

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

RT-11aa V05.xx
(Followed by any start-up file commands.)

Store the updated distribution diskettes for future updates.
7.5 Customizing the System

You may want to make certain customizations (described in Section 2.7) to the distributed RT-11 components. At this point, perform the procedures to implement any of these software customizations. Table 1–3 summarizes the available customizations and directs you to the section in Chapter 2 that describes a particular customization and the procedure for implementing it.

NOTE
Later, you can perform the system generation process to implement additional customizations. (See Section 7.9.)

7.6 Compressing Each Diskette

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

Continue to run RT-11 from Unit 0 and use the SQUEEZE command to compress free space. The squeeze operation does not move files with the .BAD file type.

```
.SQUEEZE DYO:
DYO:/Squeeze: Are you sure? Y
```

There is a delay while the squeeze operation takes place.

```
RT-11xx VO5,xx
(Followed by any start-up file commands.)
```

The system automatically reboots when you compress a system diskette.

Then insert the next diskette that you need to compress in Unit 1.

```
.SQUEEZE DYO:
DYO:/Squeeze: Are you sure? Y
```

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

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

Installing a System Distributed on RX02 to Run on RX02 7-7
7.7 Preserving the Working System

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

Use the following command to protect all the files on the system diskette:

```
,PROTECT/SYSTEM *.*
Files protected:
DK:aaaaaa.ttt
DK:bbbbbb.ttt
DK:cccccc.ttt
DK:dddddd.ttt
...
DK:zzzzzz.ttt
```

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

```
,PROTECT/SYSTEM DY1:*.*
Files protected:
DY1:aaaaaa.ttt
DY1:bbbbbb.ttt
DY1:cccccc.ttt
DY1:dddddd.ttt
...
DY1:zzzzzz.ttt
```

Next, copy the working system to backup diskettes. Insert each blank diskette in Unit 1 with RT-11 still booted from Unit 0. Format and initialize the appropriate number of diskettes.

Copy all the files in your working system. You can use the SQUEEZE/OUTPUT:DY1: DY0: command to copy any bootable diskettes. Remember that you must use SQUEEZE/WAIT/OUTPUT:DY1: DY0: and change diskettes to copy the diskettes that are not bootable. Also remember to copy the bootstrap to any diskettes that need to be bootable.

Store the backup diskettes. If you ever need to restore the working system, you can make copies of the backup working system diskettes.

7.8 Testing the Working System

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

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

SWAP.SYS
RT11FB.SYS
DY.SYS (system device handler)
LP.SYS
EDIT.SAV
MACRO.SAV
SYSMAC.SML
LINK.SAV
PIP.SAV
DUP.SAV
DIR.SAV
DEMOBG.MAC
DEMOFG.MAC

Insert a blank diskette in Unit 1.

,FORMAT DY1:RED
DY1:/FORMAT-Are you sure? YRED
?FORMAT-I-Formatting complete

,INITIALIZE/BADBLOCKS DY1:RED
DY1:/Initialize- Are you sure? YRED
?DUP-I-No bad blocks detected DY1:

,ASSIGN DY1: OK:RED

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

,DIRECTORY/BRIEF/COLUMNS:1 SY:RED
dd-mmm-yyyy
SWAP .SYS
RT11FB.SYS
LP .SYS
DY .SYS
EDIT .SAV
MACRO .SAV
SYSMAC.SML
LINK .SAV
PIP .SAV

... xxx Files, bbb Blocks
fff Free blocks
NOTE

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

Before you can execute the background and foreground demonstration programs, you must first edit, assemble, and link the background program, DEMOBG.MAC, and you must assemble and link the foreground program, DEMOFG.MAC.

7.8.1 Preparing the Background Demonstration Program

7.8.1.1 Edit the Background Demonstration Program — Use a text editor, for example, EDIT.SAV, to modify the background demonstration program, DEMOBG.MAC. One of the output lines in the program is preceded by a semicolon, which makes the line a comment field. The semicolon prevents the line from being printed; thus, it must be deleted from that line. If DEMOBG.MAC is a protected file, remove the protection before making the edits (UNPROTECT SY:DEMOBG.MAC).

\texttt{.EDIT SY:DEMOBG.MAC^{®}}
\texttt{\$F:^{®} ASCII^{®}}
\texttt{\$O:ASCII\^{®}}
\texttt{\$X:ASCII\^{®}}

7.8.1.2 Assemble the Background Demonstration Program — The background program, DEMOBG.MAC, is an assembly language source file; it must be assembled and linked before you can execute it. To assemble DEMOBG.MAC and obtain a listing, make sure that your configuration has a line printer that is on-line and ready.

\texttt{.ASSIGN LP: LST:^{®}}

\texttt{.ASSIGN TT: LST:^{®}}

NOTE

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

\texttt{.ASSIGN TT: LST:^{®}}

Assemble DEMOBG.MAC as follows:

\texttt{.MACRO/LIST:LST: SY:DEMOBG^{®}}
(See Figure 7–2.)

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

\texttt{.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC^{®}}

7–10 Installing a System Distributed on RX02 to Run on RX02
7.8.1.3 Link the Background Demonstration Program — Link the program DEMOBG.OBJ to produce an executable background program, DEMOBG.SAV.

\texttt{.LINK \text{DEMOBG.OBJ}}

7.8.1.4 Run the Background Demonstration Program — Run the program DEMOBG to check the results of the first exercise.

\texttt{.RUN \text{DEMOBG}}

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

\texttt{.RENAME SY:DEMOBG.BAK SY:DEMOBG.MACED}

Then, repeat the editing procedure.
7.8.2 Preparing the Foreground Demonstration Program

To execute the foreground program, you must assemble the program DEMOFG.MAC, link it for the foreground, and execute it in conjunction with DEMOBG.SAV. DEMOFG.MAC is a small foreground program that sends a message every 2 seconds to DEMOBG (running in the background), telling it to ring the terminal bell. DEMOBG recognizes these messages and rings the bell once for each message sent.

Although DEMOFG is always active, sending messages to the background every 2 seconds, this exercise can execute other programs in the background besides DEMOBG. The circuit is complete and messages are successfully received and honored only when DEMOBG is active. During those periods when DEMOBG is not running, DEMOFG enters the messages in the monitor message queue. Once you restart DEMOBG in the background, the system immediately releases all the messages queued since the last forced exit, resulting in many successive bell rings. When the queue is empty, the normal send/receive cycle resumes and the bell rings every 2 seconds as each current message is sent and honored.

7.8.2.1 Assemble the Foreground Demonstration Program — The foreground demonstration program, DEMOFG.MAC, is an assembly language source file; it must be assembled and linked before you can use it. Assemble DEMOFG.MAC as follows:

```
.MACRO/LIST:LST: SY:DEMOFG

```

The output resulting from this MACRO command is an object file called DEMOFG.OBJ. This file resides on your data volume (DK).

7.8.2.2 Link the Foreground Demonstration Program — You must link the DEMOFG.OBJ file to produce an executable program. Use the /FOREGROUND option to produce the load module DEMOFG.REL. The .REL file type signifies to the system that the file is a foreground program and is to be run as a priority job.

```
.LINK/FOREGROUND DEMOFG

```

7.8.2.3 Run the Foreground and Background Demonstration Programs — Type the following command to load and start DEMOFG.REL as the foreground job.

```
.FRUN DEMOFG

```

FOREGROUND DEMONSTRATION PROGRAM
Sends a message to the background program DEMOBG
Every 2 seconds, telling it to ring the bell.

7-12 Installing a System Distributed on RX02 to Run on RX02
DEMOFG.REL is now running and queueing the message for DEMOBG every 2 seconds. Now execute DEMOBG.SAV in the background and receive the messages.

\texttt{.RUN DEMOBG}\texttt{\textcopyright} \\

(The bell rings quickly several times, then once every 2 seconds.)

\texttt{RT-11 DEMONSTRATION PROGRAM}  \\
\texttt{IF INCORRECTLY EDITED, THIS IS THE LAST LINE.}  \\
\texttt{WELL DONE.}  \\

Execute a \texttt{DIRECTORY} command in the background to obtain a directory listing.

\texttt{\textcopyright}  \\
\texttt{\textcopyright}  \\

(The bell stops ringing.)

\texttt{.DIRECTORY}\texttt{\textcopyright}  \\
\texttt{dd-mmm-yyyy}  \\

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

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

\texttt{.RUN DEMOBG}\texttt{\textcopyright} \\

(The bell rings several times in rapid succession, then rings once every 2 seconds.)

\texttt{RT-11 DEMONSTRATION PROGRAM}  \\
\texttt{IF INCORRECTLY EDITED, THIS IS THE LAST LINE.}  \\
\texttt{WELL DONE.}  \\
\texttt{\textcopyright}  \\
\texttt{\textcopyright}  \\

(The bell stops ringing.)

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

\texttt{\textcopyright}  \\
\texttt{ F>}  \\
\texttt{\textcopyright}  \\
\texttt{\textcopyright}  \\
\texttt{B>}  \\
\texttt{UNLOAD F}\texttt{\textcopyright}  \\

If you completed these exercises without error, your system has passed this minimal test and you can consider it successfully installed.
7.9 Performing the System Generation Process

If you have decided that you need RT–11 features that are available only if you generate your own monitor(s) and handlers, perform the system generation process at this point. You should have thoroughly studied Chapter 1 to make this decision and to establish that you can perform the system generation process on your particular hardware configuration. Read the *RT–11 System Generation Guide* for guidance in planning and performing system generation.
Chapter 8
Installing a System Distributed on Magtape to Run on Hard Disk

If RT-11 was distributed to you on magtape, and you intend to build a system to run on hard disk, perform the procedures described in this chapter. These procedures assume that your configuration includes only one disk drive. With two disk drives, you still have to perform the logical steps summarized in the following list; however, you can simplify some procedures by using the additional disk drive. Sections 8.1 through 8.11 describe the procedures involved in each step. Figure 8-1 shows the various backup magtapes you create when you install RT-11.

NOTE

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

1. Bootstrap the first distribution magtape.
2. Preserve both distribution magtapes.
3. Install software updates.
4. Create updated master magtapes.
5. Create the working system on the disk from chosen components.
6. Install the bootstrap on the disk.
7. Customize the system.
8. Compress the disk.
9. Preserve the working system.
10. Test the working system.

11. If appropriate, perform the system generation process.

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

**Figure 8-1: Sample Backup Magtapes**

![Diagram showing backup and distribution magtapes]

### 8.1 Bootstrapping the First Distribution Magtape

The first procedure you perform when installing RT-11 is to bootstrap the first distribution magtape.

Begin by making sure that the processor is powered up but not running. Mount the distribution magtape labeled 1/2 (remove the write ring from the back of the tape reel). Manually position the magtape at the load point if the magtape is not in this position.

If your system has a hardware bootstrap capable of bootstrapping the magtape, boot the tape and proceed. If not, use the appropriate toggle-in software bootstrap in Appendix B.

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

MSBOOT responds on the terminal.

```
MSBOOT \000 \000
```

The next step depends on whether you have a TM11 or TJU16 as a magtape device. If you have a TM11, use the magtape build program named
MDUP.MT. If you have a TJU16, use the magtape build program named MDUP.MM.

MDUP.MT

or

MDUP.MM

The magtape moves as the specified MDUP program is loaded.

MDUP V0X.rr
   *

The next procedure is to build a minimal system on the disk. Start by initializing the system disk. If your disk is an RK05, be sure to use a formatted disk.

You need to initialize the system disk and scan it for bad blocks before you can copy system files to it. You specify these operations to MDUP by entering the device name followed by combinations of the following options.

/2Z to initialize the disk

/2B to scan it for bad blocks

Mount a formatted disk (write-enabled).

Use the following command, where xx is the permanent device name (RK, DL, DM, DP, or DU) for your disk.

xx0:/2Z ?

The system scans the disk for bad blocks and creates a directory.

*

Now you can build a minimal system on the disk. Depending on the type of tape drive you have, use one of the two following commands (where xx is the permanent device name for your disk).

If the magtape is TM11:

xx0:A=M1C:BE

If the magtape is TJU16:

xx0:A=MMC:BE

The tape moves while the system copies the monitor, swap file, system device handler, terminal handler, line printer handler, magtape handler, PIP, DUP, and DIR to the disk. When the files are copied, MDUP boots the minimal system from the disk.
You are now running from the minimal system on the disk. This minimal system supports enough file maintenance commands to allow you to complete the building process.

**NOTE**

MDUP does not support automatic replacement of bad blocks for RK06, RK07, RL01, and RL02 disks. If your disk is an RK06, RK07, RL01, or RL02 and you want automatic bad block replacement, you must initialize a second disk and copy your files to it at a later time.

If you wish to set the date and time, use the DATE command to set the date (where dd–mmm–yy is the day, month, and year in the form 10–JAN–83); use the TIME command to set the time (where hh:mm:ss is the hour, minutes, and seconds).

```plaintext
.DATE dd–mmm–yy
.TIME hh:mm:ss
```

Now copy the rest of the files from distribution magtape 1 to the disk. Use the following command, where xx is MT or MM.

```plaintext
.COPY /SYSTEM/NOREPLACE xx0:*.* DK:
Files copied:
x0:aaaaa.ttt to DK:aaaaa.ttt
.
xx0:zzzzz.ttt to DK:zzzzz.ttt
?PIP–W–Reboot
```

**NOTE**

You must use the /NOREPLACE option in this command, so that the files you copied to the disk when you built the minimal system will not be copied again. The system prints a message to tell you which files it does not copy (for example, SWAP.SYS not copied).

Reboot as advised.

```plaintext
.Boot sy
```

### 8.2 Preserving Both Distribution Magtapes

Copy both distribution magtapes for backup, as a safety measure in case of machine failure or human error. To back up the magtapes, complete the following steps:

---

8–4 Installing a System Distributed on Magtape to Run on Hard Disk
1. Replace distribution magtape 1 in the tape drive with a blank magtape.
2. Copy all the distribution files from the disk to the blank backup magtape.
3. Replace the backup magtape with distribution magtape 2.
4. Copy everything from distribution magtape 2 to disk.
5. Replace distribution magtape 2 with a blank magtape.
6. Copy those files that were on distribution magtape 2 from the disk to the blank backup magtape 2.

**Step 1**

Remove distribution magtape 1 and mount a blank magtape on the tape drive. Leave the write ring in the back of the reel, and make sure that the tape is positioned at the load point.

**Step 2**

Copy all files from the disk to the magtape by invoking the indirect command file DISMT1.COM (procedure follows). DISMT1.COM initializes the blank magtape, writes the primary bootstrap on it, and copies a duplicate of distribution magtape 1. Before you invoke this indirect command file, use the ASSIGN command to assign the logical name DIS: to your disk device and the logical name TAP: to your tape drive. In the commands, xx is MT or MM, and yy is your disk.

```
ASSIGN xxxx: TAP:
ASSIGN yyyy: DIS:
@DISMT1
```

(The commands in the indirect file print on the terminal.)

**Step 3**

Rewind the newly created backup magtape, remove it, and label it “Backup RT-11 V05 1/2”.

Remove the write ring from the back of distribution magtape 2, if necessary, and mount the reel.

**Step 4**

Copy all files from distribution magtape 2 to the disk. Use the following command, where xx is MT or MM.

```
COPY/SYSTEM/NOREPLACE xxx0:*** DK:
Files copied:
xxx0:aaaaaa.ttt to DK:aaaaaa.ttt

xxx0:zzzzzz.ttt to DK:zzzzzz.ttt
?PIP-W-Reboot
```

Reboot the disk.

---

*Installing a System Distributed on Magtape to Run on Hard Disk* 8-5
Step 5
Rewind the distribution magtape and remove it. Mount another blank mag-
tape, leaving the write ring in the back of the reel. Make sure that the tape
is positioned at the load point.

Step 6
Use the indirect command file DISMT2.COM to initialize the magtape and
then to duplicate the copies of files from distribution magtape 2 on the
newly initialized blank magtape (procedure follows). Note that this tape is
not a bootable magtape. Use the ASSIGN command to assign the logical
name DIS: to your disk device and the logical name TAP: to your tape
drive. In the commands, xx is MT or MM, and yy is your disk.

```
;ASSIGN  xxn:  TAP:   
;ASSIGN  yyn:  DIS:   
 Toolbox
```
(The commands in the indirect file print on the terminal.)

Then, rewind the newly created backup magtape, remove it, and label it
"Backup RT–11 V05 2/2".

Store the distribution magtapes.

8.3 Installing Software Updates
To make sure that RT–11 operates correctly, you must install updated soft-
ware modules at this point. Updated modules are critical to system or com-
ponent operation. They correct software errors discovered since the last
release, and may add new functions to the operating system.

Updated software modules are distributed in update kits. Each kit contains
complete software module replacements. Software corrections will already
have been installed in the new modules and fully tested. Your only require-
ment is to copy the new modules from the distribution medium to your
system, using the procedures described in the RT–11 Update User’s Guide.
Updates are distributed on the same medium on which you received the
base system and contain modules for both base system and layered prod-
ucts.

When you submit an SPR, the RT–11 Software Dispatch Review will report
serious problems and may contain suggested solutions to problems, until
the module containing the error is corrected by an update kit.

For a complete description of how to update your software, refer to the

8.4 Creating Updated Master Magtapes
Once you have installed updates in the components you copied to the disk,
copy all the files to blank magtapes. These magtapes can serve as updated
masters. As other mandatory updates are published, you can copy the af-
fected component from the updated master magtape to the disk, install the
update, and copy the file back to the updated master. In this way, you can
make sure that you have up-to-date versions of all RT–11 components, even
if your working system is destroyed.

Mount another blank magtape, leaving the write ring in the back of the
reel. Make sure that the tape is positioned at the load point.

Use the following procedures to initialize the magtape and copy the same
files to updated master magtape 1 as you copied to backup magtape 1 (in-
cluding the bootstrap) and the same files to updated master magtape 2 as
you copied to backup magtape 2. In the commands, xx is MT or MM, and yy
is your disk.

```
. ASSIGN x xn: TAP: ;
. ASSIGN y yn: DIS: ;
. 0DISMT1

(The commands in the indirect file print on the terminal.)
```

Rewind the newly created magtape, remove it, and label it "Updated Mas-
ter RT–11 V05 1/2". Mount another blank magtape, leaving the write ring
in the back of the reel. Make sure that the tape is positioned at the load
point.

Use the following procedures to initialize the magtape and copy the same
files (now updated) to this magtape that you copied to backup magtape 2. In
the command, xx is MT or MM, and yy is your disk.

```
. ASSIGN x xn: TAP: ;
. ASSIGN y yn: DIS: ;
. 0DISMT2

(The commands in the indirect file print on the terminal.)
```

Then, rewind the newly created magtape, remove it, and label it "Updated
Master RT–11 V05 2/2".

Store the two updated master magtapes, and use them when you install
any future software updates.

### 8.5 Creating the Working System from Chosen Components

Once you have chosen your system components (Section 2.3), backed up the
distribution magtapes, and created updated masters, you can create the
working system by deleting unneeded components from the disk.

The following command queries you about all the files on the disk. Choose
the files you want to delete.

```
. DELETE/SYSTEM/QUERY *. *

Files deleted:

DK: aaaaa. ttt? Y
DK: bbbbb. ttt? N

(to delete a specific file)

(to retain a specific file)

(and so on)
```
8.6 Installing the Bootstrap on the Disk

Once you have created your working system, copy the bootstrap from the monitor file of your choice (RT11BL, RT11SJ, RT11FB, or RT11XM) to the disk. In the command, xx is the permanent device name for your disk, and aa is BL, SJ, FB, or XM.

```
.COPY/BOOT xx0:RT11aa.SYS xx0:RE
```

Then, halt the processor, and use the hardware bootstrap to boot the working system disk.

```
RT-11aa V05.xx
```

(Followed by any start-up file commands.)

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

8.7 Customizing the System

You may want to make certain customizations (described in Section 2.7) to the distributed RT-11 components. At this point, perform the procedures to implement any of these software customizations. Table 1–3 summarizes the available customizations and directs you to the section in Chapter 2 that describes a particular customization and the procedure for implementing it.

NOTE
Later, you can perform the system generation process to implement additional customizations. (See Section 8.11.)

8.8 Compressing the Disk

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

Use the SQUEEZE command for this procedure. (The disk must be write-enabled.) The squeeze operation does not move files with the .BAD file type. In the command, xx is the device name for your disk.

```
.SQUEEZE xxn:RE
xxn:/Squeeze? Are you sure? YRE
```

```
RT-11aa V05.xx
```

(Followed by any start-up file commands.)

The system automatically reboots when you compress a system disk.
8.9 Preserving the Working System

Once you build a satisfactory working system, DIGITAL recommends that you protect all the files in it and preserve the system on the backup medium of your choice.

Use the following procedures to back up the disk on magtape. First, protect all the files on the disk.

```
PROTECT/SYSTEM *.tik
Files protected:
DK:aaaaaa.ttt
DK:bbbbbb.ttt
DK:cccccc.ttt
DK:dddddd.ttt
DK:zzzzzz.ttt
```

Next, initialize another blank magtape and copy files to it. You may be able to fit the working system on one backup magtape. You can place approximately 3000 blocks on a 60 foot 800 bits/in magtape.

Mount a blank magtape, leaving the write ring in the back of the reel. Make sure that the tape is positioned at the load point. Use the following command, where xx is MT or MM:

```
INITIALIZE/FILE:MSBOOT.BOT xx0:00
xx0:/Initialize; Are you sure? Y00
```

Then, copy all the files from the disk to the magtape, in the order shown. Essentially, you should use the following procedure whenever you build a bootable magtape. If you do not copy the files in this order, bootstrapping the magtape will be a painfully long process. Keep track of all the files you copy, so that when you copy the rest of the files, you will know which files you have already copied. If you have deleted some device handlers from the system, the log that prints on the terminal will not include the deleted device handlers.

The following COPY procedure shows a log from a system that includes all the device handlers. In the commands, xx is MT, MM, or MS, and aa is BL, SJ, FB, or XM.

```
COPY MSBOOT.BOT xx0:MSBOOT.BOT/POSITION:-100
COPY/SYSTEM MDUP.* xx0:MDUP.*/POSITION:-100
Files copied:
DK:MDUP.MM to xx0:MDUP.MM
DK:MDUP.MT to xx0:MDUP.MT
DK:MDUP.MS to xx0:MDUP.MS
COPY SWAP.SYS xx0:SWAP.SYS/POSITION:-100
COPY RT11aa.SYS xx0:RT11aa.SYS/POSITION:-100
```
Copy the rest of the files. The following command queries you about all the files on the disk so that you can choose the files it copies.

Rewind the newly created backup working system magtape, remove it, label it "Backup Working System RT-11 V05", and then store it.

8.10 Testing the Working System

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

To execute this demonstration, you need a terminal with a bell, and your working system must include at least the following components.

SWAP.SYS
RT11FB.SYS
xx.SYS (system device handler)
LP.SYS
EDIT.SAV
MACRO.SAV
SYSMAC.SML
LINK.SAV
PIP.SAV
DUP.SAV
DIR.SAV
DEMOBG.MAC
DEMOFG.MAC

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

```
$ Directory/brief/columns:1 sy;off
   dd-mm-yy
   SWAP .SYS
   RT11FB.SYS
   LP .SYS
   DL .SYS
   MT .SYS
   EDIT .SAV
   MACRO .SAV
   SYSMAC.SML
   LINK .SAV
   PIP .SAV

   xxx Files, bbb Blocks
   fff Free blocks
```

**NOTE**

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

Before you can execute the background and foreground demonstration programs, you must first edit, assemble, and link the background program,
DEMOBG.MAC, and you must assemble and link the foreground program, DEMOFG.MAC.

8.10.1  Preparing the Background Demonstration Program

8.10.1.1 Edit the Background Demonstration Program — Use a text editor, for example, EDIT.SAV, to modify the background demonstration program, DEMOUG.MAC. One of the output lines in the program is preceded by a semicolon, which makes the line a comment field. The semicolon prevents the line from being printed; thus, it must be deleted from that line. If DEMOBG.MAC is a protected file, remove the protection before making the edits (UNPROTECT SY:DEMOBG.MAC).

.EDIT SY:DEMOBG.MAC
.*F; (63, ASCII (63, 63)
.*OAD (63, 63)
.*EX (63, 63)
.

8.10.1.2 Assemble the Background Demonstration Program — The background program, DEMOUG.MAC, is an assembly language source file; it must be assembled and linked before you can execute it. To assemble DEMOUG.MAC and obtain a listing, make sure that your configuration has a line printer that is on-line and ready.

.*ASSIGN LP; LST: (63)
.

NOTE

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

.*ASSIGN TT; LST: (63)
.

Assemble DEMOUG.MAC as follows:

.*MACRO/LIST; LST: DEMOUG (63)
(See Figure 8–2.)

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

.*RENAME SY:DEMOUG.BAK SY:DEMOUG.MAC (63)
.

8–12  Installing a System Distributed on Magtape to Run on Hard Disk
Figure 8–2: DEMOBUG Assembly Listing

DEMOBUG MACRO V05.01E Friday, 09-Dec-89 10:16 Page 1

1 2 3 4 5 6
7 8 9 10 11 12
13 14 15 16 17 18
19 20 21 22 23 24
25 26 27 28 29

.DEVICE DEMOBUG
.DEVICE V05.00
"DEMOGRAPHIC PROGRAM TO PRINT DEMONSTRATION MESSAGE, THEN"
"RING BELL IF FS SENDS A MESSAGE."
") CALL .RCVDC
"AREA+#BUFFER+#MSG=#PRINT REQUEST FOR MESSAGE"
"MSG=PRINT DEMONSTRATION MESSAGE"
") LOOP
") COMPLETION ROUTINE ENTERED WHEN FS SENDS MESSAGE"
"AREA+#BUFFER+#MSG=#PRINT MESSAGE IN RESPONSE TO MESSAGE"
") RETURN FROM COMPLETION ROUTINE"
") ASCII MESSAGES"
") LIST REX"
") 00011067 200"
") BELL = .BYTE 7;200"
") MESSAGE THAT RINGS BELL"
") ASCII "RT-11 DEMONSTRATION PROGRAM/15 12"
") ASCII "IF INCORRECTLY EDITED, THIS IS THE LAST LINE./15 12"
") ASCII "WELL DONE./"
") ASCII "BYTE 0"
") ASCII "AREA = .BLANK 6"
") ASCII "ARGUMENT AREA"

DEMOBUG MACRO V05.01E Friday, 09-Dec-89 10:16 Page 1-1
Symbol table

AREA BUFFER MSG START
000000 000042 000000 000000

ABS. 000000 000000 000000

Errors detected: 0

+++ Assembler statistics

Work file read: 0
Work file written: 0
Size of work file: 0 KB Words ( 56 Pages)
Size of core file: 0 16128 Words ( 63 Pages)
Operating system: RT-11

Elapsed time: 0:10:01:05.45
DK:DEMOBUG.DM:DEMOBUG.DA:DEMOBUG

8.10.1.3 Link the Background Demonstration Program — Link the program DEMOBUG.OBJ to produce an executable background program, DEMOBUG.SAV.

.LINK DEMOBUG

8.10.1.4 Run the Background Demonstration Program — Run the program DEMOBUG to check the results of the first exercise.

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

If you did not delete the semicolon character, the last line will not print. Return to the monitor by typing two CTRL/Cs.

CTRL C
CTRL C

Installing a System Distributed on Magtape to Run on Hard Disk 8–13
If you incorrectly edited the file, you can repeat this exercise, although you can continue without correcting the file. However, if you want to repeat the exercise, begin by using the backup demonstration program.

.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC

Then, repeat the editing procedure.

8.10.2 Preparing the Foreground Demonstration Program

To execute the foreground program, you must assemble the program DEMOGF.MAC, link it for the foreground, and execute it in conjunction with DEMOBG.SAV. DEMOFG.MAC is a small foreground program that sends a message every 2 seconds to DEMOBG (running in the background), telling it to ring the terminal bell. DEMOBG recognizes these messages and rings the bell once for each message sent.

Although DEMOFG is always active, sending messages to the background every 2 seconds, this exercise can execute other programs in the background besides DEMOBG. The circuit is complete and messages are successfully received and honored only when DEMOBG is active. During those periods when DEMOBG is not running, DEMOFG enters the messages in the monitor message queue. Once you restart DEMOBG in the background, the system immediately releases all the messages queued since the last forced exit, resulting in many successive bell rings. When the queue is empty, the normal send/receive cycle resumes and the bell rings every 2 seconds as each current message is sent and honored.

8.10.2.1 Assemble the Foreground Demonstration Program — The foreground demonstration program, DEMOFG.MAC, is an assembly language source file; it must be assembled and linked before you can use it. Assemble DEMOFG.MAC as follows:

.MACRO/LIST:LST: DEMOFG
.

The output resulting from this MACRO command is an object file called DEMOFG.OBJ. This file resides on your system volume.

8.10.2.2 Link the Foreground Demonstration Program — You must link the DEMOFG.OBJ file to produce an executable program. Use the /FOREGROUND option to produce the load module DEMOFG.REL. The .REL file type signifies to the system that the file is a foreground program and is to be run as a priority job.

.LINK/FOREGROUND DEMOFG
.

8.10.2.3 Run the Foreground and Background Demonstration Programs — Type the following command to load and start DEMOFG.REL as the foreground job.
.RUN DEMOFG.RE
F>
FOREGROUND DEMONSTRATION PROGRAM
SEND A MESSAGE TO THE BACKGROUND PROGRAM DEMOBG
EVERY 2 SECONDS, TELLING IT TO RING THE BELL.
RUN DEMOBG.RE
B>

DEMOFG.REL is now running and queuing the message for DEMOBG every 2 seconds. Now execute DEMOBG.SAV in the background and receive the messages.

, RUN DEMOFG.RE

(The bell rings quickly several times, then once every 2 seconds.)

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

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

, DIRECTORY_RE
   dd-mmmyy

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

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

, RUN DEMOFG.RE

(The bell rings several times in rapid succession, then rings once every 2 seconds.)

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

, (The bell stops ringing.)

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

F>
RUN DEMOFG.RE
B>
UNLOAD F.RE

Installing a System Distributed on Magtape to Run on Hard Disk 8–15
If you completed these exercises without error, your system has passed this minimal test and you can consider it successfully installed.

8.11 Performing the System Generation Process

If you have decided that you need RT–11 features that are available only if you generate your own monitor(s) and handlers, perform the system generation process at this point. You should have thoroughly studied Chapter 1 to make this decision and to establish that you can perform the system generation process on your particular hardware configuration. Read the *RT–11 System Generation Guide* for guidance in planning and performing system generation.
Chapter 9
Installing RT–11 on a MICRO/PDP–11 Processor

If you are installing RT–11 to run on a MICRO/PDP–11 processor, perform the procedures described in this chapter. You will have received the RT–11 operating system on RX50 diskettes, and will be installing it to run on a nonremovable RD51 disk.

Your distribution kit contains two exact copies of the RT–11 operating system on RX50 diskettes. Store one copy in a safe place, and do not modify the contents; this provides a backup (master) copy of the distributed RT–11 operating system.

Store the second copy in a safe place after copying the contents onto the RD51 disk. Use these diskettes if it is ever necessary to copy their contents onto the permanent disk, for example, to create another working system. If these diskettes are ever damaged, use the master copy to reproduce them.

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

To install your system, perform the steps summarized in the following list. Sections 9.1 through 9.11 describe the procedures involved in each step. Figure 9–1 shows the backup working system diskettes you create when you install RT–11 on a MICRO/PDP–11 processor.

1. Bootstrap the system diskette.
2. Install software updates.
3. Copy the system diskette onto the system disk.
4. Install the bootstrap on the system disk.
5. Copy the distribution diskettes onto the system disk.
6. Create the working system from chosen components.
7. Customize the system.
8. Compress the system disk.
9. Preserve the working system.
10. Test the working system.
11. If appropriate, perform the system generation process.

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

Figure 9–1: Sample Backup Diskettes

9.1 Bootstrapping the System Diskette

The first procedure you perform when installing RT–11 is to bootstrap the system diskette.

Begin by making sure that the processor is powered up but not running. Insert the system diskette, labeled 1, in RX50 diskette Unit 1 (which has the physical device name DU1:). Use the hardware bootstrap to boot the diskette. RT–11 should respond with the following message if you have successfully bootstrapped the system diskette.
After you boot the system, you can create a start-up file containing any sequence of commands that you would normally want to execute at the start of each terminal session.

You can update your start-up file at any time to change, add, or delete commands. When you first create the file, or after you update it, you can execute it with the @ command, so that the commands it contains become effective.

9.2 Installing Software Updates

To make sure that RT–11 operates correctly, you must install updated software modules at this point. Update modules are critical to system or component operation. They correct software errors discovered since the last release, and may add new functions to the operating system.

Updated software modules are distributed in update kits. Each kit contains complete software module replacements. Software corrections will already have been installed in the new modules and fully tested. Your only requirement is to copy the new modules from the update kit to your system, using the procedures described in the RT–11 Update User’s Guide. Updates are distributed on the same medium on which you received the base system and contain modules for both base system and layered products.

When you submit an SPR, the RT–11 Software Dispatch Review will report serious problems and may contain suggested solutions to the problems until the module containing the error is corrected by an update kit.

For a complete description of how to update your software, refer to the RT–11 Update User’s Guide.

9.3 Copying the System Diskette onto the System Disk

Once you have installed software updates, copy the system diskette onto the system disk (RD51).

You must first use the INITIALIZE command to initialize the RD51 system disk in Unit 0. Use the /BADD BLOCKS option with INITIALIZE to cover any bad blocks that may be on your disk. If the disk contains bad blocks, the ?DUP–W–Bad blocks detected nnnnnn message appears on the terminal.
The system scans the disk for bad blocks and creates a new directory. The monitor dot appears when this process is complete.

Then, copy the contents of the system diskette to the disk.

9.4 Installing the Bootstrap on the System Disk

Once you have copied the contents of the system diskette onto the system disk, you need to install the bootstrap on the system disk.

In this command, you identify the device on which the monitor that contains the bootstrap information resides (DU0:), the name of the monitor file (RT11FB), and the device on which you need to install the bootstrap (DU0:). This command copies bootstrap information from the monitor file to blocks 0 and 2 through 5 of the same volume. Remove the system diskette from Unit 1, and use the hardware bootstrap to boot the system disk.

RT-11 should respond with the following message if you have successfully bootstrapped the system disk:

RT-11FB V05.xx
(Followed by any start-up file commands.)

9.5 Copying the Distribution Diskettes onto the System Disk

The next operation you perform with the running RT-11 system is to copy the remaining distribution diskettes onto the system disk, which will serve as your working system disk.

Insert distribution diskette 2 in Unit 1, and copy all its files to the system disk by typing the command:

Remove the distribution diskette from Unit 1 and store it; mount distribution diskette 3 in Unit 1, and copy all its files to the system disk by typing the command:

9-4 Installing RT-11 on a MICRO/PDP-11 Processor
Repeat this procedure to copy the remaining distribution diskettes onto the system disk. When you have completed copying all distribution diskettes, store them in a safe place.

Next, remove the protection from all the files on the system disk. The files on the distribution diskettes have been protected to prevent you from accidentally deleting them. (Refer to the RT–11 System User’s Guide for a description of file protection.) When you copied the files to the system disk, RT–11 also copied the protection. (Note the P that prints next to the file size in the directory.) For the rest of these procedures, you must remove the protection from the files on the system disk.

Type the following command to remove the protection from all files:

```
,UNPROTECT/SYSTEM DUO:.*&
    Files unprotected:
    DUO:aaaaaa.ttt
    DUO:bbbbbb.ttt
    DUO:ccccccc.ttt
    DUO:ddddddd.ttt
    ...
    DUO:zzzzzz.ttt
```

### 9.6 Creating the Working System from Chosen Components

Once you have chosen your system components (Section 2.3), you can create the working system by deleting selected components from your system disk.

You can use the following command to avoid typing numerous file specifications. RT–11 queries you about all the files on the disk, and you choose the files it deletes.

```
,DELETE/SYSTEM/QUERY DUO: &
    Files deleted:
    DUO:aaaaaa.ttt  Y& (to delete a specific file)
    DUO:bbbbbb.ttt  N& (to retain a specific file)
    (and so on)
```

Repeat this procedure to retain only the files you planned for the working system disk.

### 9.7 Customizing the System

You may want to make certain customizations (described in Section 2.7) to the distributed RT–11 components. At this point, perform the procedures to implement any of these software customizations. Table 1–3 summarizes the available customizations and directs you to the section in Chapter 2 that describes a particular customization and the procedure for implementing it.

**NOTE**

Later, you can perform the system generation process to implement additional customizations. (See Section 9.11.)
9.8 Compressing the System Disk

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

Use the SQUEEZE command to compress free space. (The volume must be write-enabled.)

```
 , SQUEEZE DUO:\n DUO:/Squeeze i Are you sure? Y\n RT-11xx VO5.xx
```

The system automatically reboots when you compress a system disk.

9.9 Preserving the Working System

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

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

```
 , PROTECT/SYSTEM DUO:.*\n Files Protected:
 DUO:aaaaaaa.ttt
 DUO:bbbbbb.ttt
 DUO:cccccc.ttt
.
...
 DUO:zzzzzz.ttt
```

Insert a blank RX50 diskette in Unit 1 or 2 so you can initialize it. You can use the INITIALIZE command with the /BADBLOCKS option to initialize the diskette and to detect any bad blocks that may be on it. If the diskette contains bad blocks, the ?DUP-W-Bad blocks detected nnnnnn message appears on the terminal. In the command, x is the device unit number.

**NOTE**

RX50 diskettes are not formattable. DIGITAL recommends that you use only diskettes that have no bad blocks. To ascertain whether an already initialized diskette has bad blocks, use the command DIRECTORY/BAD DUn:, where n is the drive number. If bad blocks exist on a diskette, copy the contents of the diskette to an error-free diskette, and dispose of the diskette with bad blocks.

```
 , INITIALIZE/BADBLOCKS DUx:\n DUx:/Initialize i Are you sure? Y\n ?DUP-I-No bad blocks detected DUx:
```

9-6 Installing RT-11 on a MICRO/PDP-11 Processor
There will be a delay while the system scans the diskette for bad blocks and creates a new directory. The monitor dot appears when this process is complete.

Now, remove the newly initialized diskette and initialize an adequate number of blank diskettes, leaving one initialized, blank, write-enabled (write-protect notch uncovered) diskette inserted in Unit 1 or 2.

Copy all the files in your working system, using the COPY/MULTIVOLUME command. In the command, x is the device unit number. The system copies all the files from the input volume that will fit on the output volume. When no more files will fit on the output volume, the system prompts you to mount the next output volume and prints the Continue? message.

Mount another blank, initialized diskette, and type Y to continue. Continue to mount diskettes in this fashion until all the files are copied.

```
COPY/MULTIVOLUME/SYSTEM DUO: DX1; R
Files copied:
YY0:aaaaaa.ttt to DX1:aaaaaa.ttt
YY0:bbbbbb.ttt to DX1:bbbbbb.ttt
(and so on)
```

Mount next output volume in DX1; Continue? Y
(Log of files copied)
Mount next output volume in DX1; Continue? Y
(Log of files copied)
Mount next output volume in DX1; Continue? Y
(Log of files copied)
Mount next output volume in DX1; Continue? Y
(Log of files copied)
Mount next output volume in DX1; Continue? Y
(and so on)

Then, install the bootstrap on any backup working system diskettes that need to be bootable. In the following command, aa is BL, SJ, FB, or XM.

```
COPY/BOOT DX1:aa DUO: RX
```

Remove the newly created working system diskette from Unit 1 or 2, and label it "Backup RT-11 V05 x" (where x is the diskette number). (Use a soft-tipped pen when you label diskettes.) Then, store it in a safe place. If you ever need to restore the working system, you can make copies of the backup working system diskettes.

As long as the diskette you intend to copy is bootable and contains certain system utility programs, you can boot RT-11 from that diskette and copy
the diskette. Distribution diskette 0 is bootable, but the rest of the diskettes in your kit are not bootable, because they lack the necessary components. Remember that a bootable diskette needs an appropriate monitor file, a bootstrap, a system device handler, the SWAP.SYS file, and for the SJ monitor, the TT.SYS file.

9.10 Testing the Working System

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

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

SWAP.SYS
RT11FB.SYS
DU.SYS (system device handler)
LP.SYS
EDIT.SAV
MACRO.SAV
SYSMAC.SML
LINK.SAV
PIP.SAV
DUP.SAV
DIR.SAV
DEMOBG.MAC
DEMOFG.MAC

NOTE

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

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

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

dd-mm-yy
SWAP .SYS
RT11FB.SYS
LP .SYS
DU .SYS
EDIT .SAV
MACRO .SAV
```
NOTE

If you have shifted output to a scope and the directory scrolls by too quickly to read, type \texttt{CTRL-S} to stop the display and \texttt{CTRL-Q} to restart it.

Before you can execute the background and foreground demonstration programs, you must first edit, assemble, and link the background program, DEMOBG.MAC, and you must assemble and link the foreground program, DEMOFG.MAC.

9.10.1 Preparing the Background Demonstration Program

9.10.1.1 Edit the Background Demonstration Program — Use a text editor, for example, EDIT.SAV, to modify the background demonstration program, DEMOBG.MAC. One of the output lines in the program is preceded by a semicolon, which makes the line a comment field. The semicolon prevents the line from being printed; thus, it must be deleted from that line. If DEMOBG.MAC is a protected file, remove the protection before making the edits (UNPROTECT SY:DEMOBG.MAC).

9.10.1.2 Assemble the Background Demonstration Program — The background program, DEMOBG.MAC, is an assembly language source file; it must be assembled and linked before you can execute it. To assemble DEMOBG.MAC and obtain a listing, make sure that your configuration has a line printer that is on-line and ready.

NOTE

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

ASSIGN TT: \_ST: \_
Assemble DEMOBG.MAC as follows:

```assembly
.MACRO /LIST:=LST: DEMOBG
(See Figure 9–2.)

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

.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC

Figure 9–2: DEMOBG Assembly Listing

DEMOBG MACRO V05.01b Friday 09-Dec-93 10:16 Page 1

```

```assembly
1 .TITLE DEMOBG
2 .IDENT "V05.00/"
3 ; DEMONSTRATION PROGRAM TO PRINT DEMONSTRATION MESSAGE, THEN
4 ; RING BELL IF FG JOB S1S SENDS A MESSAGE.
5 .MCALL .RCVDC, PRINT

.3 START: RCVDC AREA=\$BUFFER=\$MSG=\$MSGIN, ISP Request for MESSAGE
9 0 000000
10 0 000034
11 10 00042 000777
12 13 0 000044
14 0 000052
15 16 000106 000207
17 18 0 000110 007 200
19 20 000112 122 124 035
21 22 000147 111 106 040
23 24 0 000225 000
25 0 000226
26 0 000242
27 0 000000
28 0 000242
29 0 000000
30 .END START

DEMOBG MACRO V05.01b Friday 09-Dec-93 10:16 Page 1

```

Symbol table

AREA 000024R BUFFER 000242R MSGIN 000044R ...U1 + 000038 ...U2 + 000027
BELL 000110R MSG 000122R STOP 000000R

- .ABS. 000000 000 (NW1:0D=ABS-DUR)
- 000226 001 (NW1:0D=HEL.COM)

Errors detected: 0

*** Assembler statistics

Work file reads: 0
Work file writes: 0
Size of work file: 7160 Words (36 Pages)
Size of core pool: 16128 Words (63 Pages)
Operating system: RT-11

Elapsed time: 00100000.45

9.10.1.3 Link the Background Demonstration Program — Link the program DEMOBG.OBJ to produce an executable background program, DEMOBG.SAV.

.LINK DEMOBG

9.10.1.4 Run the Background Demonstration Program — Run the program DEMOBG to check the results of the first exercise.

9–10 Installing RT–11 on a MICRO/PDP–11 Processor
.RUN DEMOBG
RT-11 DEMONSTRATION PROGRAM
IF INCORRECTLY EDITED, THIS IS THE LAST LINE.
WELL DONE.

If you did not delete the semicolon character, the last line will not print.
Return to the monitor by typing two CTRL/Cs.

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

.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC

Then, repeat the editing procedure.

9.10.2 Preparing the Foreground Demonstration Program

To execute the foreground program, you must assemble the program DEMOFG.MAC, link it for the foreground, and execute it in conjunction with
DEMOBG.SAV. DEMOFG.MAC is a small foreground program that sends
a message every 2 seconds to DEMOBG (running in the background), tell-
ing it to ring the terminal bell. DEMOBG recognizes these messages and
rings the bell once for each message sent.

Although DEMOFG is always active, sending messages to the background
every 2 seconds, this exercise can execute other programs in the back-
ground besides DEMOBG. The circuit is complete and messages are suc-
cessfully received and honored only when DEMOBG is active. During those
periods when DEMOBG is not running, DEMOFG enters the messages in
the monitor message queue. Once you restart DEMOBG in the background,
the system immediately releases all the messages queued since the last
forced exit, resulting in many successive bell rings. When the queue is
empty, the normal send/receive cycle resumes and the bell rings every 2
seconds as each current message is sent and honored.

9.10.2.1 Assemble the Foreground Demonstration Program — The foreground
demonstration program, DEMOFG.MAC, is an assembly language source
file; it must be assembled and linked before you can use it. Assemble
DEMOFG.MAC as follows:

.MACRO/LIST:LST: DEMOFG
The output resulting from this MACRO command is an object file called DEMOFG.OBJ. This file resides on your system volume.

9.10.2.2 Link the Foreground Demonstration Program — You must link the DEMOFG.OBJ file to produce an executable program. Use the /FOREGROUND option to produce the load module DEMOFG.REL. The .REL file type signifies to the system that the file is a foreground program and is to be run as a priority job.

```
-LINK /FOREGROUND DEMOFG.REL
```

9.10.2.3 Run the Foreground and Background Demonstration Programs — Type the following command to load and start DEMOFG.REL as the foreground job.

```
RUN DEMOFG.REL
```

FOREGROUND DEMONSTRATION PROGRAM
SEND A MESSAGE TO THE BACKGROUND PROGRAM DEMOBG
EVERY 2 SECONDS, TELLING IT TO RING THE BELL.

DEMOFG.REL is now running and queuing the message for DEMOBG every 2 seconds. Now execute DEMOBG.SAV in the background and receive the messages.

```
RUN DEMOBG.RET
```

(The bell rings quickly several times, then once every 2 seconds.)

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

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

```
-DIRECTORY
```

(The bell stops ringing.)

```
-DD MMMM YYYY
```

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

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

```
RUN DEMOBG.RET
```
(The bell rings several times in rapid succession, then rings once every 2 seconds.)

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

(The bell stops ringing.)

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

NOTE

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

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

9.11 Performing the System Generation Process

If you have decided that you need RT-11 features that are available only if you generate your own monitor(s) and handlers, perform the system generation process at this point. You should have thoroughly studied Chapter 1 to make this decision and to establish that you can perform the system generation process on your particular hardware configuration. Read the RT-11 System Generation Guide for guidance in planning and performing system generation.
Chapter 10
Installing RT–11 on a Professional 325 Computer

If you are installing RT–11 to run on a Professional 325 computer, perform the procedures described in this chapter.

Your distribution kit contains two copies of the RT–11 operating system on RX50 diskettes. Store one set of the RX50 diskettes in a safe place and do not modify their contents; they provide a master copy of the distributed RT–11 operating system.

Store the second set of diskettes in a safe place, after installing software updates and creating your working system diskettes. The second set provides a backup copy of your updated RT–11 system. If those diskettes are ever damaged, use the master copy of the distributed RT–11 operating system to reproduce them.

To install your system, perform the procedures summarized below and described in Sections 10.1 through 10.8. Figure 10–1 shows the diskettes you create when you install RT–11.

1. Bootstrap the system diskette.
2. Install software updates.
3. Create the working system from chosen components.
4. Install the bootstrap on any diskettes that must be bootable.
5. Customize the system.
6. Preserve the working system.
7. Test the working system.
8. If appropriate, perform the system generation procedures.
10.1 **Bootstrapping the System Diskette**

The first procedure you perform when installing RT–11 bootstraps the system diskette, distribution diskette 1.

First, turn off the processor (press the power switch to 0).

Insert the distribution diskette labeled 1 in Unit 0 (which has the physical device name DZ0). Unit 0 is the top slot if your Professional 325 sits horizontally on a table top, or the left slot if it sits vertically in a floor stand.

Press the power switch to 1 (on). Wait while the system runs diagnostic tests. If there are no problems, the DIGITAL logo appears on the screen within about 30 seconds.

If a problem occurs, a diagram of the Professional computer appears on your screen. The part of the Professional that has a problem is highlighted in the picture. Write down the numbers you see on the screen. If the top number is 051124, refer to Chapter 7 of the *RT–11 System Message Manual*. Otherwise, refer to your *Professional 300 Series Owner's Manual*.

If nothing happens when you turn the power on, refer to the *Professional 300 Series Owner's Manual* for help.

RT–11 should respond with the following message if you have successfully bootstrapped the diskette:
10.2 Installing Software Updates

You must install updated software modules at this point. Updated modules are critical to system or component operation. They correct software errors discovered since the last release, and may add new functions to the operating system.

Updated software modules are distributed in update kits. Each kit contains complete software module replacements. Software corrections will already have been installed in the new modules and fully tested. Your only requirement is to copy the new modules from the update kit to your system, using the procedures described in the RT-11 Update User's Guide. Updates are distributed on the same medium on which you received the RT-11 operating system and contain modules for the operating system and layered products.

NOTE

If you are installing your system for the first time, you need not update the RT-11 operating system. The RT-11 distribution kit you received contains the most recent modules.

The RT-11 Software Dispatch Review reports serious problems submitted in SPRs and may provide temporary solutions to use until the module containing the errors is corrected by an update kit.

For a complete description of how to update your software, refer to the RT-11 Update User's Guide.

10.3 Creating the Working System from Chosen Components

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

Start by initializing a number of blank diskettes. Insert each diskette in Unit 1 and use the INITIALIZE command. Use the /BADBLOCKS option with INITIALIZE to cover any bad blocks that may be on your diskettes.
(this prevents the system from trying to use them). If a diskette contains bad blocks, the ?DUP-W-Bad blocks detected nnnnnn message appears on the terminal.

NOTE

DIGITAL recommends that you use only diskettes that do not have bad blocks when you build a working system. To ascertain whether an initialized diskette has bad blocks, use the command DIRECTORY/BAD DZn:. You can use diskettes with bad blocks later for temporary storage or as work volumes.

.,INITIALIZE/BADBLOCKS DZ1:
DZ1:/Initialize? Are you sure? Y
?DUP-I-No bad blocks detected DZ1:

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

Repeat this step to create as many initialized blank diskettes as you need for the system you have planned, leaving one initialized, blank diskette in Unit 1.

Then, use the COPY command to copy selected files from distribution diskette 1 to the diskette that will become your working system diskette. The /SYSTEM option is required for copying .SYS files only if wildcards are used in input file specifications.

.,COPY DZ0:filnam.typ DZ1:filnam.typ

You can use the following command to avoid typing numerous file specifications. RT-11 queries you about all the files on the diskette, and you choose the files it copies.

.,COPY/SYSTEM/QUERY DZ0: DZ1:
Files copied:
DZ0:aaaaaaa.ttt to DZ1:aaaaaa.ttt? Y (to include a specific file)
DZ0:bbbbbb.ttt to DZ1:bbbbbb.ttt? N (to exclude a specific file)
(and so on)

To copy files from nonbootable diskettes, you have to alternate diskettes.

Type the following command, where filnam.typ is the name of the file you want to copy.

.,COPY/QUERY/WAIT DZ1:filnam.typ DZ0:filnam.typ

Mount input volume in DZ1; Continue?
Place the diskette containing the file you want to copy in Unit 1.

Files copied: DZ1:aaaaaa.ttt to DZ0:aaaaaa.ttt?

Type Y to include or N to exclude the file.

Mount output volume in DZ0: Continue?

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

Mount system volume in DZ0: Continue?

Replace the diskette in Unit 0 with the system diskette.

Repeat this procedure to copy all the files you planned for the working system diskette.

When you have copied all the files, label the diskette "RT–11 Working System V05.n 1" (where n is the release number of RT–11 V5).

Repeat those procedures to create the other diskettes in the working system.

10.4 Installing the Bootstrap on Diskettes that Must be Bootable

Once you have created your working system, you need to install the bootstrap on any diskettes that must be bootable (that you can use as system diskettes). Generally, any diskette that includes a monitor file and system device handler should be bootable. The diskette also needs SWAP.SYS, and PI(X).SYS to run on a Professional 325.

Insert in Unit 1 the diskette on which you need to install the bootstrap. In the following command, aa is FB or XM.

,COPY/BOD DZ1:RT:1aa.SYS DZ1:RT

This command copies bootstrap information from the monitor file to blocks 0 and 2 through 5 of the same diskette. In the command, you identify the device on which the monitor that contains the bootstrap information resides, the name of the monitor file (RT11FB.SYS or RT11XM.SYS), and the device on which you need to install the bootstrap.
Then, insert working system diskette 1 in Unit 0, and use the hardware bootstrap (turn the system off, then on) to boot your working system.

 RT-11aa VOS.xx
(Followed by any start-up file commands.)

Store the updated distribution diskettes for future updates.

10.5 Customizing the System

You may want to make customizations (described in Section 2.7) to the distributed RT-11 components. At this point, perform the procedures to implement any software customizations. Table 1–3 summarizes the available customizations and directs you to the section in Chapter 2 that describes a particular customization and the procedure for implementing it.

The files on the distribution diskettes have been protected to prevent you from accidentally deleting them. (Note the P that prints next to the file size in the directory. Refer to the RT-11 System User's Guide for a description of file protection.) You need to remove the protection from the files you want to customize. Type the following command to remove protection from the files on distribution diskette 1 (your system diskette):

```
,UNPROTECT/SYSTEM *.ttt
Files unprotected:
DK:aaaaaa.ttt
DK:bbbbbb.ttt
DK:cccccc.ttt
DK:dddddd.ttt

DK:zzzzzz.ttt
```

To remove protection from files on other diskettes, insert each diskette in Unit 1. Then type the following command:

```
,UNPROTECT/SYSTEM DZ1:*.ttt
Files unprotected:
DZ1:aaaaaa.ttt
DZ1:bbbbbb.ttt
DZ1:cccccc.ttt
DZ1:dddddd.ttt

DZ1:zzzzzz.ttt
```

**NOTE**

Later, you can perform the system generation procedures to implement additional customizations.
10.6 Preserving the Working System

Once you build a satisfactory working system, DIGITAL recommends that you preserve the system on backup diskettes.

If you have unprotected files to make software customizations (Section 10.5), use the following command to protect all the files on the system diskette:

```
.PROTECT/SYSTEM *.*
Files protected:
   DK:aaaaaa.ttt
   DK:bbbbbb.ttt
   DK:cccccc.ttt
   DK:dddddd.ttt
   ...
   DK:zzzzzz.ttt
```

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

```
.PROTECT/SYSTEM DZ1:*.*
Files protected:
   DZ1:aaaaaa.ttt
   DZ1:bbbbbb.ttt
   DZ1:cccccc.ttt
   DZ1:dddddd.ttt
   ...
   DZ1:zzzzzz.ttt
```

Next, copy the working system to backup diskettes. Insert blank diskettes in Unit 1 with RT–11 still booted from Unit 0, and initialize the appropriate number of diskettes.

Then, copy all the files in your working system. To copy the system diskette, type:

```
.SQUEEZE/OUTPUT:DZ1: DZ0: *
```

To copy other diskettes, type this command and follow the instructions to change diskettes:

```
.SQUEEZE/WAIT/OUTPUT:DZ1: DZ0: *
```

Remember to copy the bootstrap to any diskettes that need to be bootable.

Store the backup diskettes. If you ever need to restore the working system, you can make copies of the backup working system diskettes.
10.7 Testing the Working System

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

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

SWAP.SYS
RT11FB.SYS or RT11XM.SYS
DZ.SYS or DZX.SYS (system device handler)
PI.SYS or PIX.SYS
LS.SYS or LSX.SYS
KED.SAV
MACRO.SAV
SYSMAC.SML
LINK.SAV
PIP.SAV
DUP.SAV
DIR.SAV
DEMOBG.MAC
DEMOFG.MAC

Insert a blank diskette in Unit 1.

,INITIALIZE/BADBLOCKS DZ1: 50
DZ1: / Initialize; Are you sure? Y 50
?DUP-1-No bad blocks detected DZ1:
,ASSIGN DZ1: DK: 50

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

,DIRECTORY/BRIEF/COLUMNS:1 SY: 85
   dd-mmm-yyyy
SWAP .SYS
RT11xx.SYS
PI .SYS
LS .SYS
DZ .SYS
KED .SAV
MACRO .SAV
SYSMAC.SML
LINK .SAV
PIP .SAV

xxx Files, bbb Blocks
fff Free blocks
NOTE

If the directory scrolls by too quickly, press the HOLD SCREEN key to stop the display. Press HOLD SCREEN again to continue scrolling.

Before you can execute the background and foreground demonstration programs, you must first edit, assemble, and link the background program, DEMOBG.MAC, and then you must assemble and link the foreground program, DEMOFG.MAC.

10.7.1 Preparing the Background Demonstration Program

To prepare the background demonstration program, DEMOBG.MAC, follow these instructions to edit, assemble, link, and run the program.

10.7.1.1 Edit the Background Demonstration Program — Use a text editor, for example KED, to modify the background demonstration program, DEMOBG.MAC. One of the lines in the program contains a message to be printed, but it is preceded by a semicolon (the symbol for a comment field). You must delete the semicolon so the message will print.

If DEMOBG.MAC is a protected file, remove the protection before making the edits (UNPROTECT SY:DEMOBG.MAC).

To start KED, type:

.KED DEMOBG.MAC

Once the file is displayed, press the down-arrow key until the cursor rests on top of the semicolon (;) at the beginning of the line:

; ASCII /WELL DONE,

Delete the semicolon by pressing the comma key (,) on the keypad.

Then, exit the file: Press the PF1 key, then the 7 key on the keypad. When KED prompts Command:, type EXIT and press the ENTER key:

Command: EXIT

10.7.1.2 Assemble the Background Demonstration Program — The background program, DEMOBG.MAC, is an assembly language source file; it must be assembled and linked before you can execute it.

To assemble DEMOBG.MAC and obtain a listing, make sure your configuration has a line printer that is on-line and ready.

.ASSIGN LS: LST:

Installing RT–11 on a Professional 325 Computer  10–9
NOTE
If your configuration does not include a line printer, use the console terminal.

.ASSIGN TT: LST:

Assemble DEMOBG.MAC as follows:

.MACRO/LIST:LST: SY:DEMOBG
(See Figure 10–2.)

If any errors occur when you assemble DEMOBG.MAC, you have incorrectly edited the file and should repeat the edits as described in Section 10.7.1.1. Use the backup demonstration program.

.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC

Figure 10–2: DEMOBG Assembly Listing

 DEMOBG MACRO V05.00 Saturday 08-Jan-83 11:44 Page 1

1 .TITLE DEMOBG
2 .IDENT /V05.00/
3 I DEMONSTRATION PROGRAM TO PRINT DEMONSTRATION MESSAGE.
4 I RING BELL IF FG JOB SENSES A MESSAGE.
5
6 MCD PER 000000
7 V 000035
8 000086: 000777
9 START: .RCVD .AREA: &BUFFER: #4000: &MSG:
10 print #MSG:
11 PRINT DEMONSTRATION MESSAGE
12 BH .
13 AND LOOP
14
15 I COMPLETION ROUTINE ENTERED WHEN FG SENDS MESSAGE
16
17 &MSG:
18 .PRINT #BELL
19 RING BELL IN RESPONSE TO MESSAGE
20 &RCVD .AREA: &BUFFER: #4000: &MSG:
21 IN FG MESSAGE REQUEST
22 RETURN AND RETURN FROM COMPLETION ROUTINE
23
24 &ARCS: 000000
25 I ASCII MESSAGES
26
27 000010 000000
28 .BYTE 0
29
30 &MG2: .BYTE 0
31
32 &ARCS: 000000
33 I ASCII WELL DONE.
34
35 000025 .BYTE 0
36
37 &AREA: .BLANK 0
38 &ARGUMENT AREA
39 &BUFFER: .ASCII 0
40 &ARGUMENT AREA
41 &END START

DEMOBG MACRO V05.00 Saturday 08-Jan-83 11:44 Page 1-1

Symbol table

AREA 000020K BUFFER 000240K &MSG: 000044K .V1 0 000401 .V2 0 00027
REL 000100K MCD 000120K START 00000000K

Errors detected: 0

*** Assembler statistics

Work file read: 0
Work file write: 0
Size of work file: 1008 Words ( 16 Pages)
Size of core pool: 16128 Words ( 64 Pages)
Operating system: RT-11

Elapsed time: 00:00:00.45
DR:DEMOG:DR:DEMOG:DR:DEMOG

10.7.1.3 Link the Background Demonstration Program — Link the program DEMOBG.OBJ to produce an executable background program, DEMOBG.SAV.
10.7.1.4 Run the Background Demonstration Program — Run the program DEMOBG to check the results of the first exercise.

If you did not delete the semicolon character, the last line will not print. Return to the monitor by typing two CTRL/Cs.

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

Then, repeat the editing procedure (see Section 10.7.1).

10.7.2 Preparing the Foreground Demonstration Program

To execute the foreground program, you must assemble the program DEMOFG.MAC, link it for the foreground, and then execute it in conjunction with DEMOBG.SAV.

DEMOFG.MAC is a small foreground program that sends a message every 2 seconds to DEMOBG (running in the background), telling it to ring the terminal bell. DEMOBG recognizes those messages and rings the bell once for each message sent.

Although DEMOFG is always active, sending messages to the background every 2 seconds, this exercise can execute other programs in the background besides DEMOBG. The circuit is complete and messages are successfully received and honored only when DEMOBG is active. During those periods when DEMOBG is not running, DEMOFG enters the messages in the monitor message queue. Once you restart DEMOBG in the background, the system immediately releases all the messages queued since the last forced exit, resulting in many successive bell rings. When the queue is empty, the normal send/receive cycle resumes and the bell rings every 2 seconds as each current message is sent and honored.

10.7.2.1 Assemble the Foreground Demonstration Program — The foreground demonstration program, DEMOFG.MAC, is an assembly language source
file which you must assemble and link before running. Assemble DEMOFG.MAC as follows:

```
.MACRO/LIST: LST: SY:DEMOFG

```

The output resulting from this MACRO command is an object file called DEMOFG.OBJ, which resides on your data volume (DK).

10.7.2.2 Link the Foreground Demonstration Program — You must link the DEMOFG.OBJ file to produce an executable program. Use the /FOREGROUND option to produce the load module DEMOFG.REL. The .REL file type tells the system that the file is a foreground program and should be run as a priority job.

```
.LINK/FOREGROUND DEMOFG

```

10.7.3 Run the Foreground and Background Demonstration Programs

Type the following command to load and start DEMOFG.REL as the foreground job.

```
.FRUN DEMOFG

```

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

DEMOFG.REL is now running and queuing the message for DEMOBG every 2 seconds. Now execute DEMOBG.SAV in the background and receive the messages.

```
.RUN DEMOBG

```

(The bell rings quickly several times, then once every 2 seconds.)

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

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

```
CTRL C

```

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

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

(The bell rings several times in rapid succession, then rings once every 2 seconds.)

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

(The bell stops ringing.)

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

CTRL F
F>
CTRL C CTRL C
B>
UNLOAD F

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

10.8 Performing the System Generation Procedures

If you have decided that you need RT-11 features that are available only if you generate your own monitor(s) and handlers, perform the system generation procedures at this point. You should have thoroughly studied Chapter 1 to make this decision and to establish that you can perform the system generation procedures on your particular hardware configuration.

DIGITAL does not recommend that you perform a system generation on a Professional 325. The process is lengthy and is not automated. However, should you decide to perform a system generation, read the RT-11 System Generation Guide for guidance.
Chapter 11
Installing RT–11 on a Professional 350 Computer

If you are installing RT–11 to run on a Professional 350 computer, perform the procedures described in this chapter.

You will have received the RT–11 operating system on RX50 diskettes, and will be installing it to run on a nonremovable RD50 or RD51 disk.

NOTE

The procedures described in this chapter assume that your Professional 350 system includes an RD50 or RD51 Winchester disk. If your system does not include a Winchester disk, install RT–11 software by following the procedures described in Chapter 5.

Your distribution kit contains two copies of the RT–11 operating system on RX50 diskettes. Store one set of the RX50 diskettes in a safe place, and do not modify their contents; they provide a master copy of the distributed RT–11 operating system.

Store the second set of diskettes in a safe place, after installing software updates and copying their contents onto the nonremovable disk. The second set provides a backup copy of your updated RT–11 system. Use those diskettes if it is ever necessary to copy their contents onto the permanent disk, for example, to create another working system. If those diskettes are ever damaged, use the master copy of the distributed RT–11 operating system to reproduce them.

To install your system, perform the procedures summarized below and described in Sections 11.1 through 11.11. Figure 11–1 shows the diskettes you create when you install RT–11.

1. Bootstrap the system diskette.
2. Install software updates.
3. Copy the system diskette onto the system disk.
4. Install the bootstrap on the system disk.
5. Copy the distribution diskettes onto the system disk.
6. Create the working system from chosen components.
7. Customize the system.
8. Compress the system disk.
9. Preserve the working system.
10. Test the working system.
11. If appropriate, perform the system generation procedures.

Figure 11–1: Sample Backup Diskettes

11.1 Bootstrapping the System Diskette

The first procedure you perform when installing RT–11 bootstraps the system diskette.

First, turn off the processor (press the power switch to 0).

Make sure the system diskette, labeled 1, is write-protected (with a write-protect tape over the notch), and insert it in diskette Unit 0 (which has the physical device name DZ0:). Unit 0 is the top slot if your Professional 350 sits horizontally on a table top, or the left slot if it sits vertically in a floor stand.
Press the power switch to 1 (on). Wait while the system runs diagnostic tests. If there are no problems, the DIGITAL logo appears on the screen within about 30 seconds.

If a problem occurs, a diagram of the Professional computer appears on your screen. The part of the Professional that has a problem is highlighted in the picture. Write down the numbers you see on the screen. If the top number is 051124 refer to Chapter 7 of the **RT–11 System Message Manual**. Otherwise, refer to your **Professional 300 Series Owner’s Manual**.

If nothing happens when you turn the power on, refer to the **Professional 300 Series Owner’s Manual** for help.

RT–11 should respond with the following message if you have successfully bootstrapped the system diskette.

```
RT-11XM V05.xx
.TYPE V5USER.TXT

RT-11 V5.xx

Installation of RT-11 ....(followed by V5USER.TXT message)....
```

### 11.2 Installing Software Updates

To make sure that RT–11 operates correctly, you must install updated software modules at this point. Update modules are critical to system or component operation. They correct software errors discovered since the last release, and may add new functions to the operating system.

Updated software modules are distributed in update kits. Each kit contains complete software module replacements. Software corrections will already have been installed in the new modules and fully tested. Your only requirement is to copy the new modules from the update kit to your system, using the procedures described in the **RT–11 Update User’s Guide**. Updates are distributed on the same medium on which you received the RT–11 operating system and contain modules for the operating system and layered products.

**NOTE**

If you are installing your system for the first time, you need not update the RT–11 operating system. The RT–11 distribution kit you received contains the most recent modules.
The *RT–11 Software Dispatch Review* reports serious problems submitted in SPRs and may provide temporary solutions to use until the module containing the errors is corrected by an update kit.

For a complete description of how to update your software, refer to the *RT–11 Update User's Guide*.

### 11.3 Copying the System Diskette onto the System Disk

Once you have installed software updates, copy the system diskette onto the nonremovable system disk (RD50 or RD51).

You must first use the INITIALIZE command to initialize the nonremovable system disk. Use the `/BADBLOCKS` option with INITIALIZE to cover any bad blocks that may be on your disk (this prevents the system from trying to use them). If the disk contains bad blocks, the `?DUP–W–Bad blocks detected nnnnnn` message appears on the terminal.

```
.INITIALIZE/BADBLOCKS DW0:
.DW0:/Initialize; Are you sure? Y
?DUP–I–No bad blocks detected DW0:
```

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

```

```

Then, copy the contents of the system diskette onto the disk.

```
.COPY/SYSTEM/NOPROTECT DZ0: DW0:
```

### 11.4 Installing the Bootstrap on the System Disk

Once you have copied the contents of the system diskette onto the system disk, you need to install the bootstrap on the system disk.

```
.COPY/BOOT DW0:RT11XM.SYS DW0:
```

This command copies bootstrap information from the monitor file to blocks 0 and 2 through 5 of the same volume. In the command, you identify the device (DW0:) on which resides the monitor that contains the bootstrap information, the name of the monitor file (RT11XM), and the device on which you need to install the bootstrap (DW0:).

Remove the system diskette from Unit 0, and use the hardware bootstrap to boot the system disk. RT–11 should respond with the following message if you have successfully bootstrapped the system disk:

```
RT–11XM V05,xx
(Followed by any start-up file commands.)
```

11–4 Installing RT–11 on a Professional 350 Computer
11.5 Copying the Distribution Diskettes onto the System Disk

The next operation you perform with the running RT-11 system copies the remaining distribution diskettes onto the system disk that will serve as your working system disk.

Insert distribution diskette 2 in Unit 0, and copy all its files to the system disk by typing the command:

```
.COPY/SYSTEM/NOPROTECT DWO: DW0: RET
```

Remove the distribution diskette from Unit 0 and store it. Then repeat this procedure to copy the remaining distribution diskettes onto the system disk. When you have completed copying all distribution diskettes, store them in a safe place.

11.6 Creating the Working System from Chosen Components

Once you have chosen your system components (see Section 2.3), you can create the working system by deleting selected components from your system disk.

You can use the following command to avoid typing numerous file specifications. RT-11 queries you about all the files on the disk, and you choose the files it deletes.

```
.DELETE/SYSTEM/QUERY DW0: RET
Files deleted:
DW0:aaaaaaa.ttt Y RET (to delete a specific file)
DW0:bbbbbbb.ttt N RET (to retain a specific file)
```

(and so on)

Repeat this procedure to retain only the files you planned for the working system disk.

11.7 Customizing the System

You may want to make customizations (described in Section 2.7) to the distributed RT-11 components. At this point, perform the procedures to implement any software customizations. Table 1-3 summarizes the available customizations and directs you to the section in Chapter 2 that describes a particular customization and the procedure for implementing it.

**NOTE**

Later, you can perform the system generation procedures to implement additional customizations.

11.8 Compressing the System Disk

DIGITAL recommends that you compress the working system disk to make its free space contiguous. Consolidating free space lets you use space on the disk that would otherwise be too fragmented to be usable.
Use the SQUEEZE command to compress free space. (The volume must be write-enabled.)

```
.SQUEEZE DWO:
DWO:/Squeeze: Are you sure? Y
RT-11xx V05.xx
```

The system automatically reboots when you compress a system disk.

### 11.9 Preserving the Working System

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

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

```
.PROTECT/SYSTEM DWO:.*
Files protected:
DWO:aaaaaa.ttt
DWO:bbbbbb.ttt
DWO:cccccc.ttt
.
.
DWO:zzzzzz.ttt
```

Insert a blank RX50 diskette in diskette Unit 0 or 1 so you can initialize it. You can use the INITIALIZE command with the /BADBLOCKS option to initialize the diskette and to detect any bad blocks that may be on it. If the diskette contains bad blocks, the *DUP-W-Bad blocks detected nnnnnn message appears on the terminal. In the command, n is the device unit number.

**NOTE**

DIGITAL recommends that you use only diskettes that have no bad blocks. To ascertain whether an already initialized diskette has bad blocks, use the command DIRECTORY/BAD DZn: where n is the drive number. If bad blocks exist on a diskette, copy the contents of the diskette to an error-free diskette. You can use diskettes with bad blocks later for temporary storage or as work volumes.

```
.INITIALIZE/BADBLOCKS DZn:
DZn:/Initialize: Are you sure? Y
?DUP-I-No bad blocks detected DZn:
```

There will be a delay while the system scans the diskette for bad blocks and creates a new directory. The monitor dot appears when this process is complete.
Now, remove the newly initialized diskette and initialize an adequate number of blank diskettes, leaving one initialized, blank, write-enabled (write-protect notch uncovered) diskette inserted in Unit 0 or 1.

Copy all the files in your working system, using the COPY/MULTIVOLUME command below. In the command, n is the device unit number. The system copies all the files from the disk (the input volume) that will fit on the diskette (the output volume). When no more files will fit on the diskette, the system prompts you to mount another output volume and prints the Continue? message.

Mount another blank, initialized diskette, and type Y to continue. Continue to mount diskettes in this fashion until all the files are copied.

```
.COPY/MULTIVOLUME/SYSTEM DW0: DZn:FF
    Files copied:
    DW0:aaaaaa.ttt to DZn:aaaaaa.ttt
    DW0:bbbbbb.ttt to DZn:bbbbbb.ttt
    (and so on)

Mount next output volume in DZn:;  Continue? Y
    (Log of files copied)

Mount next output volume in DZn:;  Continue? Y
    (Log of files copied)
    (and so on)
```

Then, install the bootstrap on any backup working system diskettes that need to be bootable. In the following command, aa is FB or XM.

```
.COPY/FILE DZn:FF11aa DZn:FF
```

Remove each diskette from Unit 0 or 1, and label it "Backup RT–11 V05 x" (where x is the diskette number). Use a soft-tipped pen when you label diskettes. Then, store them in a safe place.

If you ever need to restore the working system, you can copy the backup working system diskettes to DW::

As long as the diskette you intend to copy is bootable and contains certain system utility programs, you can boot RT–11 from that diskette and copy the diskette. Distribution diskette 1 is bootable, but the rest of the diskettes in your kit are not bootable because they lack the necessary components. A bootable diskette needs an appropriate monitor file, a bootstrap, a system device handler, the SWAP.SYS file, and the Professional interface PI(X).SYS.

11.10 Testing the Working System

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

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

SWAP.SYS
RT11FB.SYS or RT11XM.SYS
DW.SYS or DWX.SYS (system device handler)
PI.SYS or PIX.SYS
LS.SYS or LSX.SYS
KED.SAV
MACRO.SAV
SYSMAC.SML
LINK.SAV
PIP.SAV
DUP.SAV
DIR.SAV
DEMOBG.MAC
DEMOFG.MAC

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

```
.DIRECTORY/BRIEF/COLUMNS:1 SY:
  dd-mm-yy
SWAP .SYS
RT11FB.SYS
LS .SYS
DW .SYS
PI .SYS
KED .SAV
MACRO .SAV
SYSMAC.SML
LINK .SAV
PIP .SAV

xxx Files, bbb Blocks
fff Free blocks
```

**NOTE**

If the directory scrolls by too quickly, press the HOLD SCREEN key to stop the display.Press HOLD SCREEN again to continue scrolling.

Before you can execute the background and foreground demonstration programs, you must first edit, assemble, and link the background program, DEMOBG.MAC, and then you must assemble and link the foreground program, DEMOFG.MAC.
11.10.1 Preparing the Background Demonstration Program

To prepare the background demonstration program, DEMOBG.MAC, follow these instructions to edit, assemble, link, and run the program.

11.10.1.1 Edit the Background Demonstration Program — Use a text editor, for example, KED.SAV, to modify the background demonstration program, DEMOBG.MAC. One of the lines in the program contains a message to be printed, but it is preceded by a semicolon (the symbol for a comment field). You must delete the semicolon so the message will print.

If DEMOBG.MAC is a protected file, remove the protection before making the edits (UNPROTECT SY:DEMOBG.MAC).

To start KED, type:

.KED DEMOBG.MAC

Once the file is displayed, press the down-arrow key until the cursor rests on top of the semicolon (;) at the beginning of the line:

; ASCII /WELL DONE./

Delete the semicolon by pressing the comma key (,.) on the keypad.

Then, exit the file: press the PF1 key, then the 7 key on the keypad. When KED prompts Command:, type EXIT and press the ENTER key:

Command: EXIT

11.10.1.2 Assemble the Background Demonstration Program — The background program, DEMOBG.MAC, is an assembly language source file; it must be assembled and linked before you can execute it.

To assemble DEMOBG.MAC and obtain a listing, make sure that your configuration has a line printer that is on-line and ready.

.ASCII

,ASSIGN LS: LST:

NOTE

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

,ASSIGN TT: LST:

Assemble DEMOBG.MAC as follows:

,MACRO/LIST: LST: DEMOBG
(See Figure 11–2.)
If any errors occur when you assemble DEMOBG.MAC, you have incorrectly edited the file and should repeat the edits as described in Section 11.10.1.1. Use the backup demonstration program.

.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC

Figure 11–2: DEMOBG Assembly Listing

DEMOBG MACRO V05.00 Saturday 08-Jan-03 11:44 Page 1

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29

DEMOBG MACRO V05.00 Saturday 08-Jan-03 11:44 Page 1-1

Symbol Table

AREA 00022A8 BUFFER 0002428 MSGIN 0000448 ...
BELL 000110R MSG 000120R START 0000000D

Errors detected: 0

BBE Assembler statistics

Work file name: 0
Work file write: 0
Size of work file: 988 Words (36 Pages)
Size of core map: 16128 Words (63 Pages)

Operating system: RT-11

Elapsed time: 00:00:05.43
DR (DEMOBG:DR:DEMOBG:DR:DEMOBG)

11.10.1.3 Link the Background Demonstration Program — Link the program DEMOBG.OBJ to produce an executable background program, DEMOBG.SAV.

.LINK DEMOBG

11.10.1.4 Run the Background Demonstration Program — Run the program DEMOBG to check the results of the first exercise.

.RUN DEMOBG

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

If you did not delete the semicolon character, the last line will not print. Return to the monitor by typing two CTRL/Cs.

CTRL/C  CTRL/C

11–10 Installing RT-11 on a Professional 350 Computer
If you incorrectly edited the file, you can repeat the exercise, although you can continue without correcting the file. If you want to repeat the exercise, begin by using the backup demonstration program.

.RENAME SY:DEMOBG.BAK SY:DEMOBG.MAC

Then, repeat the editing procedure (see Section 11.10.1.1).

11.10.2 Preparing the Foreground Demonstration Program

To execute the foreground program, you must assemble the program DEMOFG.MAC, link it for the foreground, and then execute it in conjunction with DEMOBG.SAV.

DEMOFG.MAC is a small foreground program that sends a message every 2 seconds to DEMOBG (running in the background), telling it to ring the terminal bell. DEMOBG recognizes those messages and rings the bell once for each message sent.

Although DEMOFG is always active, sending messages to the background every 2 seconds, this exercise can execute other programs in the background besides DEMOBG. The circuit is complete and messages are successfully received and honored only when DEMOBG is active. During those periods when DEMOBG is not running, DEMOFG enters the messages in the monitor message queue. Once you restart DEMOBG in the background, the system immediately releases all the messages queued since the last forced exit, resulting in many successive bell rings. When the queue is empty, the normal send/receive cycle resumes and the bell rings every 2 seconds as each current message is sent and honored.

11.10.2.1 Assemble the Foreground Demonstration Program — The foreground demonstration program, DEMOFG.MAC, is an assembly language source file which you must assemble and link before running. Assemble DEMOFG.MAC as follows:

.MACRO LIST:LST: DEMOFG

The output resulting from this MACRO command is an object file called DEMOFG.OBJ which resides on your system volume.

11.10.2.2 Link the Foreground Demonstration Program — You must link the DEMOFG.OBJ file to produce an executable program. Use the /FOREGROUND option to produce the load module DEMOFG.REL. The .REL file type tells the system that the file is a foreground program and should be run as a priority job.

.LINK FOREGROUND DEMOFG

Installing RT–11 on a Professional 350 Computer   11–11
11.10.3 Run the Foreground and Background Demonstration Programs

Type the following command to load and start DEMOFG.REL as the foreground job.

```
.FRUN DEMOFG.REL
> FOREGROUND DEMONSTRATION PROGRAM
SENG A MESSAGE TO THE BACKGROUND PROGRAM DEMOBG
EVERY 2 SECONDS, TELLING IT TO RING THE BELL.
CTRL-B
> DEMOFG.REL is now running and queueing the message for DEMOBG
every 2 seconds. Now execute DEMOBG.SAV in the background and re-
ceive the messages.

.RUN DEMOBG.REL
```

(The bell rings quickly several times, then once every 2 seconds.)

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

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

```
CTRL-C CTRL-C
> (The bell stops ringing.)
.DIRECTORY.REL
  dd-mmm-yyyy
> (The directory of the device DK prints on the terminal.)
```

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

```
.RUN DEMOBG.REL
```

(The bell rings several times in rapid succession, then rings once every 2
seconds.)

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

(The bell stops ringing.)
Now, stop the foreground program and remove it from memory.

CTRL F
F >
CTRL C  CTRL C
B >
UNLOAD F

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

11.11 Performing the System Generation Procedures

If you have decided that you need RT-11 features that are available only if you generate your own monitor(s) and handlers, perform the system generation procedures at this point. You should have thoroughly studied Chapter 1 to make this decision and to establish that you can perform the system generation procedures on your particular hardware configuration. Read the RT-11 System Generation Guide for guidance in planning and performing system generation.
Appendix A

Modifying MDUP to Recognize a Device Handler

The Version 5 MDUP.MM, MDUP.MS, and MDUP.MT programs support the following devices:

- RK05
- RK06
- RK07
- RL01
- RL02
- RP02
- RP03
- RS03
- RS04
- RF11

If you need to create MDUP for a disk or magtape not supported by RT-11 (for which you have written your own handler), use MDUP.SAV as follows.

Use the hardware version of the magtape handler and the standard (distributed) single-job monitor. Your configuration must include at least 28K words of memory. Apply the following customization to the monitor. In the customization, xx is the name of the device on which the monitor resides.

```
.R SIPP
.*xxn:RT11SJ.SYS
Base? 0@F
Offset? 1116@F

Base       Offset       Old       New?
000000   001116       000407   240@F
000000   001120       013704   12784@F
000000   001122       177570   60000@F
000000   001124       042704   ONLY@F

*STR @F
```

Then copy the bootstrap:

```
COPY/BOOT xxn:RT11SJ.SYS xxn:RT
```

This procedure causes the system to use only 12K words of memory when booted.
Next apply the following customization to every handler to be supported by the version of MDUP you are building. In the customization, xx is the name of the device on which the handler resides, and yy is the name of the handler.

```
.R SIPP
*xx:yy,SYN
Base?  063
Offset? 176

<table>
<thead>
<tr>
<th>Base</th>
<th>Offset</th>
<th>Old</th>
<th>New?</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000</td>
<td>000176</td>
<td>?????</td>
<td>0 BE</td>
</tr>
<tr>
<td>000000</td>
<td>000200</td>
<td>?????</td>
<td>0 BE</td>
</tr>
<tr>
<td>000000</td>
<td>000202</td>
<td>?????</td>
<td>CRT/0 BE</td>
</tr>
</tbody>
</table>
```

Now boot the monitor you have modified. Set the USR NOSWAP. Load all the handlers to be supported. Run MDUP.SAV and issue the following command. In the command, xx is the physical device name, and filnam.typ is your version of MDUP (MDUP.??).

```
.R MDUP.SAV
*xx:filnam.typ=/H
```

MDUP.SAV creates the file filnam.typ and exits.

If you make a typing error in the command line, use CRT/C to abort MDUP.SAV and run MDUP.SAV again. If you get an insufficient memory error message, you cannot build MDUP with all the handlers loaded. Unload one of the handlers, and run MDUP.SAV again.
Appendix B
Loading Software Bootstraps

If your hardware configuration procedure does not include a hardware bootstrap, use one of the following procedures for loading a software bootstrap. Find the section for bootstrapping your device, and use the switch register to deposit the bootstrap in memory. If your hardware configuration includes a pushbutton console emulator instead of a switch register, follow the instructions in your hardware manual to load the appropriate bootstrap loader (listed in Tables B–1 through B–9).

B.1 Loading the RK11 (RK05) DECPack Bootstrap

Deposit the basic RK11 (RK05) disk bootstrap loader in memory as follows:

1. Set the ENABLE/HALT switch to HALT.
2. Set the first address, 001000, in the switch register (see Table B–1).
3. Press the LOAD ADDR switch.
4. Set the contents for the first address (from Table B–1) in the switch register.
5. Lift the DEP switch. The computer automatically advances to the next address.
6. Set the contents for the next address (from Table B–1) in the switch register.
7. Lift the DEP switch.
8. Repeat steps 6 and 7 until you have deposited all the instructions.
Table B–1: RK11 (RK05) Bootstrap Loader

<table>
<thead>
<tr>
<th>Address</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>001000</td>
<td>012700</td>
</tr>
<tr>
<td>001002</td>
<td>177406</td>
</tr>
<tr>
<td>001004</td>
<td>012710</td>
</tr>
<tr>
<td>001006</td>
<td>177400</td>
</tr>
<tr>
<td>001010</td>
<td>012740</td>
</tr>
<tr>
<td>001012</td>
<td>000005</td>
</tr>
<tr>
<td>001014</td>
<td>105710</td>
</tr>
<tr>
<td>001016</td>
<td>100376</td>
</tr>
<tr>
<td>001020</td>
<td>005007</td>
</tr>
</tbody>
</table>

Now verify that you deposited the bootstrap loader properly.

1. Set the first address, 001000, in the switch register.

2. Press the LOAD ADDR switch.

3. Press the EXAM switch to display the contents of that address in the data register.

4. Compare the value in the data register with the value for that address in Table B–1.

5. If the values are the same, press EXAM again to display the contents of the next address. If the values are not the same, repeat the entire procedure for depositing the bootstrap. Verify the contents of all the addresses in this way. If any instruction is incorrect, repeat the entire deposit procedure.

Once you have correctly deposited the bootstrap in memory, start the computer as follows:

1. Set the starting address, 001000, in the switch register.

2. Press the LOAD ADDR switch.

3. Set the ENABLE/HALT switch to ENABLE.

4. Press the START switch.

B.2 Loading the TC11 DECTape Bootstrap

Deposit the basic TC11 DECTape bootstrap loader in memory as follows:

1. Set the ENABLE/HALT switch to HALT.

2. Set the first address, 001000, in the switch register (see Table B–2).

3. Press the LOAD ADDR switch.

4. Set the contents for the first address (from Table B–2) in the switch register.

5. Lift the DEP switch. The computer automatically advances to the next address.
6. Set the contents for the next address (from Table B–2) in the switch register.

7. Lift the DEP switch.

8. Repeat steps 6 and 7 until you have deposited all the instructions.

**Table B–2: TC11 Bootstrap Loader**

<table>
<thead>
<tr>
<th>Address</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>001000</td>
<td>012700</td>
</tr>
<tr>
<td>001002</td>
<td>177344</td>
</tr>
<tr>
<td>001004</td>
<td>012710</td>
</tr>
<tr>
<td>001006</td>
<td>177400</td>
</tr>
<tr>
<td>001010</td>
<td>012740</td>
</tr>
<tr>
<td>001012</td>
<td>004002</td>
</tr>
<tr>
<td>001014</td>
<td>005710</td>
</tr>
<tr>
<td>001016</td>
<td>100376</td>
</tr>
<tr>
<td>001020</td>
<td>012710</td>
</tr>
<tr>
<td>001022</td>
<td>000003</td>
</tr>
<tr>
<td>001024</td>
<td>105710</td>
</tr>
<tr>
<td>001026</td>
<td>100376</td>
</tr>
<tr>
<td>001030</td>
<td>012710</td>
</tr>
<tr>
<td>001032</td>
<td>000005</td>
</tr>
<tr>
<td>001034</td>
<td>105710</td>
</tr>
<tr>
<td>001036</td>
<td>100376</td>
</tr>
<tr>
<td>001040</td>
<td>005007</td>
</tr>
</tbody>
</table>

Now verify that you deposited the bootstrap loader properly.

1. Set the first address, 001000, in the switch register.

2. Press the LOAD ADDR switch.

3. Press the EXAM switch to display the contents of that address in the data register.

4. Compare the value in the data register with the value for that address in Table B–2.

5. If the values are the same, press EXAM again to display the contents of the next address. If the values are not the same, repeat the entire procedure for depositing the bootstrap. Verify the contents of all the addresses in this way. If any instruction is incorrect, repeat the entire deposit procedure.

Once you have correctly deposited the bootstrap in memory, start the computer as follows:

1. Set the starting address, 001000, in the switch register.

2. Press the LOAD ADDR switch.

3. Set the ENABLE/HALT switch to ENABLE.

4. Press the START switch.
B.3 Loading the RX11 Bootstrap

The procedure to deposit the RX11 disk bootstrap loader in memory depends on the type of processor you have. If your computer is a PDP-11V/03, PDP-11/03, or LSI-11, see the PDP-11/03 User's Manual for instructions. Otherwise deposit the RX11 disk bootstrap loader in memory as follows:

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

Table B–3: RX11 Bootstrap Loader

<table>
<thead>
<tr>
<th>Address</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>001000</td>
<td>005000</td>
</tr>
<tr>
<td>001002</td>
<td>012701</td>
</tr>
<tr>
<td>001004</td>
<td>177170</td>
</tr>
<tr>
<td>001006</td>
<td>105711</td>
</tr>
<tr>
<td>001010</td>
<td>001776</td>
</tr>
<tr>
<td>001012</td>
<td>012711</td>
</tr>
<tr>
<td>001014</td>
<td>000003</td>
</tr>
<tr>
<td>001016</td>
<td>005711</td>
</tr>
<tr>
<td>001020</td>
<td>001776</td>
</tr>
<tr>
<td>001022</td>
<td>100405</td>
</tr>
<tr>
<td>001024</td>
<td>105711</td>
</tr>
<tr>
<td>001026</td>
<td>100004</td>
</tr>
<tr>
<td>001030</td>
<td>116120</td>
</tr>
<tr>
<td>001032</td>
<td>000002</td>
</tr>
<tr>
<td>001034</td>
<td>000770</td>
</tr>
<tr>
<td>001036</td>
<td>000000</td>
</tr>
<tr>
<td>001040</td>
<td>005000</td>
</tr>
<tr>
<td>001042</td>
<td>000110</td>
</tr>
</tbody>
</table>

Now verify that you deposited the bootstrap loader properly.
1. Set the first address, 001000, in the switch register.

2. Press the LOAD ADDR switch.

3. Press the EXAM switch to display the contents of that address in the data register.

4. Compare the value in the data register with the value for that address in Table B–3.

5. If the values are the same, press EXAM again to display the contents of the next address. If the values are not the same, repeat the entire procedure for depositing the bootstrap. Verify the contents of all the addresses in this way. If any instruction is incorrect, repeat the entire deposit procedure.

Once you have correctly deposited the bootstrap in memory, start the computer as follows:

1. Set the starting address, 001000, in the switch register.

2. Press the LOAD ADDR switch.

3. Set the ENABLE/HALT switch to ENABLE.

4. Press the START switch.

B.4 Loading the TJU16 or TM11 Magtape Bootstrap

Deposit the basic TJU16 or TM11 magtape bootstrap loader in memory as follows:

1. Set the ENABLE/HALT switch to HALT.

2. Set the first address, 001000 or 010000, in the switch register (see Table B–4 or B–5).

3. Press the LOAD ADDR switch.

4. Set the contents for the first address (from Table B–4 for TJU16 magtape or from Table B–5 for TM11 magtape) in the switch register.

5. Lift the DEP switch. The computer automatically advances to the next address.

6. Set the contents for the next address (from Table B–4 or B–5) in the switch register.

7. Lift the DEP switch.

8. Repeat steps 6 and 7 until you have deposited all the instructions.
Table B-4: TJU16 Bootstrap Loader

<table>
<thead>
<tr>
<th>Address</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>001000</td>
<td>012700</td>
</tr>
<tr>
<td>001002</td>
<td>172440</td>
</tr>
<tr>
<td>001004</td>
<td>012710</td>
</tr>
<tr>
<td>001006</td>
<td>000021</td>
</tr>
<tr>
<td>001010</td>
<td>012760</td>
</tr>
<tr>
<td>001012</td>
<td>001300</td>
</tr>
<tr>
<td>001014</td>
<td>000032</td>
</tr>
<tr>
<td>001016</td>
<td>012760</td>
</tr>
<tr>
<td>001020</td>
<td>177777</td>
</tr>
<tr>
<td>001022</td>
<td>000006</td>
</tr>
<tr>
<td>001024</td>
<td>012720</td>
</tr>
<tr>
<td>001026</td>
<td>000031</td>
</tr>
<tr>
<td>001030</td>
<td>105760</td>
</tr>
<tr>
<td>001032</td>
<td>000010</td>
</tr>
<tr>
<td>001034</td>
<td>100375</td>
</tr>
<tr>
<td>001036</td>
<td>012710</td>
</tr>
<tr>
<td>001040</td>
<td>177000</td>
</tr>
<tr>
<td>001042</td>
<td>012740</td>
</tr>
<tr>
<td>001044</td>
<td>000071</td>
</tr>
<tr>
<td>001046</td>
<td>032710</td>
</tr>
<tr>
<td>001050</td>
<td>100200</td>
</tr>
<tr>
<td>001052</td>
<td>001775</td>
</tr>
<tr>
<td>001054</td>
<td>100007</td>
</tr>
<tr>
<td>001056</td>
<td>022760</td>
</tr>
<tr>
<td>001060</td>
<td>001000</td>
</tr>
<tr>
<td>001062</td>
<td>000014</td>
</tr>
<tr>
<td>001064</td>
<td>001403</td>
</tr>
<tr>
<td>001066</td>
<td>000005</td>
</tr>
<tr>
<td>001070</td>
<td>000167</td>
</tr>
<tr>
<td>001072</td>
<td>177704</td>
</tr>
<tr>
<td>001074</td>
<td>005007</td>
</tr>
</tbody>
</table>

Now verify that you deposited the bootstrap loader properly.

1. Set the first address, 001000 or 010000, in the switch register.

2. Press the LOAD ADDR switch.

3. Press the EXAM switch to display the contents of that address in the data register.

4. Compare the value in the data register with the value for that address in Table B-4 or B-5.

5. If the values are the same, press EXAM again to display the contents of the next address. If the values are not the same, repeat the entire procedure for depositing the bootstrap. Verify the contents of all the addresses in this way. If any instruction is incorrect, repeat the entire deposit procedure.
Table B–5: TM11 Bootstrap Loader

<table>
<thead>
<tr>
<th>Address</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>010000</td>
<td>012700</td>
</tr>
<tr>
<td>010002</td>
<td>172524</td>
</tr>
<tr>
<td>010004</td>
<td>005310</td>
</tr>
<tr>
<td>010006</td>
<td>012740</td>
</tr>
<tr>
<td>010010</td>
<td>060011</td>
</tr>
<tr>
<td>010012</td>
<td>105710</td>
</tr>
<tr>
<td>010014</td>
<td>100376</td>
</tr>
<tr>
<td>010016</td>
<td>005710</td>
</tr>
<tr>
<td>010020</td>
<td>100767</td>
</tr>
<tr>
<td>010022</td>
<td>012710</td>
</tr>
<tr>
<td>010024</td>
<td>060003</td>
</tr>
<tr>
<td>010026</td>
<td>105710</td>
</tr>
<tr>
<td>010030</td>
<td>100376</td>
</tr>
<tr>
<td>010032</td>
<td>005710</td>
</tr>
<tr>
<td>010034</td>
<td>100777</td>
</tr>
<tr>
<td>010036</td>
<td>005007</td>
</tr>
</tbody>
</table>

Once you have correctly deposited the bootstrap in memory, start the computer as follows:

1. If the magtape is not positioned at the load point, rewind the magtape manually.
2. Set the starting address, 001000 or 010000, in the switch register.
3. Press the LOAD ADDR switch.
4. Set the ENABLE/HALT switch to ENABLE.
5. Press the START switch.

B.5 Loading the RK06/RK07 DECpack Bootstrap

Deposit the basic RK06 or RK07 disk bootstrap loader in memory as follows:

1. Set the ENABLE/HALT switch to HALT.
2. Set the first address, 001000, in the switch register (see Table B–6).
3. Press the LOAD ADDR switch.
4. Set the contents for the first address (from Table B–6) in the switch register.
5. Lift the DEP switch. The computer automatically advances to the next address.
6. Set the contents for the next address (from Table B–6) in the switch register.
7. Lift the DEP switch.
8. Repeat steps 6 and 7 until you have deposited all the instructions.
Table B–6: RK06/RK07 Bootstrap Loader

<table>
<thead>
<tr>
<th>Address</th>
<th>Contents RK06</th>
<th>Contents RK07</th>
</tr>
</thead>
<tbody>
<tr>
<td>001000</td>
<td>012701</td>
<td>012701</td>
</tr>
<tr>
<td>001002</td>
<td>177440</td>
<td>177440</td>
</tr>
<tr>
<td>001004</td>
<td>012711</td>
<td>012711</td>
</tr>
<tr>
<td>001006</td>
<td>000003</td>
<td>002003</td>
</tr>
<tr>
<td>001010</td>
<td>032711</td>
<td>032711</td>
</tr>
<tr>
<td>001012</td>
<td>100200</td>
<td>100200</td>
</tr>
<tr>
<td>001014</td>
<td>001775</td>
<td>001775</td>
</tr>
<tr>
<td>001016</td>
<td>012761</td>
<td>012761</td>
</tr>
<tr>
<td>001020</td>
<td>177400</td>
<td>177400</td>
</tr>
<tr>
<td>001022</td>
<td>000002</td>
<td>000002</td>
</tr>
<tr>
<td>001024</td>
<td>012711</td>
<td>002021</td>
</tr>
<tr>
<td>001026</td>
<td>000021</td>
<td>000021</td>
</tr>
<tr>
<td>001030</td>
<td>032711</td>
<td>032711</td>
</tr>
<tr>
<td>001032</td>
<td>100200</td>
<td>100200</td>
</tr>
<tr>
<td>001034</td>
<td>001775</td>
<td>001775</td>
</tr>
<tr>
<td>001036</td>
<td>005007</td>
<td>005007</td>
</tr>
</tbody>
</table>

Now verify that you deposited the bootstrap loader properly.

1. Set the first address, 001000, in the switch register.

2. Press the LOAD ADDR switch.

3. Press the EXAM switch to display the contents of that address in the data register.

4. Compare the value in the data register with the value for that address in Table B–6.

5. If the values are the same, press EXAM again to display the contents of the next address. If the values are not the same, repeat the entire procedure for depositing the bootstrap. Verify the contents of all the addresses in this way. If any instruction is incorrect, repeat the entire deposit procedure.

Once you have correctly deposited the bootstrap in memory, start the computer as follows:

1. Set the starting address, 001000, in the switch register.

2. Press the LOAD ADDR switch.

3. Set the ENABLE/HALT switch to ENABLE.

4. Press the START switch.

B.6 Loading the RL01/RL02 Disk Bootstrap

Deposit the basic RL01/RL02 disk bootstrap loader in memory as follows:

1. Set the ENABLE/HALT switch to HALT.

2. Set the first address, 001000, in the switch register (see Table B–7).
3. Press the LOAD ADDR switch.

4. Set the contents for the first address (from Table B–7) in the switch register.

5. Lift the DEP switch. The computer automatically advances to the next address.

6. Set the contents for the next address (from Table B–7) in the switch register.

7. Lift the DEP switch.

8. Repeat steps 6 and 7 until you have deposited all the instructions.

**Table B–7: RL01/RL02 Bootstrap Loader**

<table>
<thead>
<tr>
<th>Address</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>001000</td>
<td>012701</td>
</tr>
<tr>
<td>001002</td>
<td>174400</td>
</tr>
<tr>
<td>001004</td>
<td>012761</td>
</tr>
<tr>
<td>001006</td>
<td>000013</td>
</tr>
<tr>
<td>001010</td>
<td>000004</td>
</tr>
<tr>
<td>001012</td>
<td>012711</td>
</tr>
<tr>
<td>001014</td>
<td>000004</td>
</tr>
<tr>
<td>001016</td>
<td>105711</td>
</tr>
<tr>
<td>001020</td>
<td>100376</td>
</tr>
<tr>
<td>001022</td>
<td>005061</td>
</tr>
<tr>
<td>001024</td>
<td>000002</td>
</tr>
<tr>
<td>001026</td>
<td>005061</td>
</tr>
<tr>
<td>001030</td>
<td>000004</td>
</tr>
<tr>
<td>001032</td>
<td>012761</td>
</tr>
<tr>
<td>001034</td>
<td>177400</td>
</tr>
<tr>
<td>001036</td>
<td>000006</td>
</tr>
<tr>
<td>001040</td>
<td>012711</td>
</tr>
<tr>
<td>001042</td>
<td>000014</td>
</tr>
<tr>
<td>001044</td>
<td>105711</td>
</tr>
<tr>
<td>001046</td>
<td>100376</td>
</tr>
<tr>
<td>001050</td>
<td>005007</td>
</tr>
</tbody>
</table>

Now verify that you deposited the bootstrap loader properly.

1. Set the first address, 001000, in the switch register.

2. Press the LOAD ADDR switch.

3. Press the EXAM switch to display the contents of that address in the data register.

4. Compare the value in the data register with the value for that address in Table B–7.

5. If the values are the same, press EXAM again to display the contents of the next address. If the values are not the same, repeat the entire procedure for depositing the bootstrap. Verify the contents of all the addresses in this way. If any instruction is incorrect, repeat the entire deposit procedure.
Once you have correctly deposited the bootstrap in memory, start the computer as follows:

1. Set the starting address, 001000, in the switch register.
2. Press the LOAD ADDR switch.
3. Set the ENABLE/HALT switch to ENABLE.
4. Press the START switch.

B.7 Loading the RX211 Bootstrap

Deposit the basic RX211 bootstrap loader in memory as follows:

1. Set the ENABLE/HALT switch to HALT.
2. Set the first address, 002000, in the switch register (see Table B–8).
3. Press the LOAD ADDR switch.
4. Set the contents for the first address (from Table B–8) in the switch register.
5. Lift the DEP switch. The computer automatically advances to the next address.
6. Set the contents for the next address (from Table B–8) in the switch register.
7. Lift the DEP switch.
8. Repeat steps 6 and 7 until you have deposited all the instructions.
<table>
<thead>
<tr>
<th>Address</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>002000</td>
<td>012701</td>
</tr>
<tr>
<td>002002</td>
<td>177170</td>
</tr>
<tr>
<td>002004</td>
<td>012700</td>
</tr>
<tr>
<td>002006</td>
<td>100240</td>
</tr>
<tr>
<td>002010</td>
<td>005002</td>
</tr>
<tr>
<td>002012</td>
<td>012705</td>
</tr>
<tr>
<td>002014</td>
<td>000200</td>
</tr>
<tr>
<td>002016</td>
<td>012704</td>
</tr>
<tr>
<td>002020</td>
<td>000401</td>
</tr>
<tr>
<td>002022</td>
<td>012703</td>
</tr>
<tr>
<td>002024</td>
<td>177172</td>
</tr>
<tr>
<td>002026</td>
<td>030011</td>
</tr>
<tr>
<td>002030</td>
<td>001776</td>
</tr>
<tr>
<td>002032</td>
<td>100440</td>
</tr>
<tr>
<td>002034</td>
<td>012711</td>
</tr>
<tr>
<td>002036</td>
<td>000407</td>
</tr>
<tr>
<td>002040</td>
<td>030011</td>
</tr>
<tr>
<td>002042</td>
<td>001776</td>
</tr>
<tr>
<td>002044</td>
<td>100433</td>
</tr>
<tr>
<td>002046</td>
<td>110413</td>
</tr>
<tr>
<td>002050</td>
<td>000304</td>
</tr>
<tr>
<td>002052</td>
<td>030011</td>
</tr>
<tr>
<td>002054</td>
<td>001776</td>
</tr>
<tr>
<td>002056</td>
<td>110413</td>
</tr>
<tr>
<td>002060</td>
<td>000304</td>
</tr>
<tr>
<td>002062</td>
<td>030011</td>
</tr>
<tr>
<td>002064</td>
<td>001776</td>
</tr>
<tr>
<td>002066</td>
<td>100422</td>
</tr>
<tr>
<td>002070</td>
<td>012711</td>
</tr>
<tr>
<td>002072</td>
<td>000403</td>
</tr>
<tr>
<td>002074</td>
<td>030011</td>
</tr>
<tr>
<td>002076</td>
<td>001776</td>
</tr>
<tr>
<td>002100</td>
<td>100415</td>
</tr>
<tr>
<td>002102</td>
<td>010513</td>
</tr>
<tr>
<td>002104</td>
<td>030011</td>
</tr>
<tr>
<td>002106</td>
<td>001776</td>
</tr>
<tr>
<td>002110</td>
<td>100411</td>
</tr>
<tr>
<td>002112</td>
<td>010213</td>
</tr>
<tr>
<td>002114</td>
<td>060502</td>
</tr>
<tr>
<td>002116</td>
<td>060502</td>
</tr>
<tr>
<td>002120</td>
<td>122424</td>
</tr>
<tr>
<td>002122</td>
<td>120427</td>
</tr>
<tr>
<td>002124</td>
<td>000007</td>
</tr>
<tr>
<td>002126</td>
<td>003737</td>
</tr>
<tr>
<td>002130</td>
<td>005000</td>
</tr>
<tr>
<td>002132</td>
<td>005007</td>
</tr>
<tr>
<td>002134</td>
<td>000000</td>
</tr>
</tbody>
</table>
Now verify that you deposited the bootstrap loader properly.

1. Set the first address, 002000, in the switch register.
2. Press the LOAD ADDR switch.
3. Press the EXAM switch to display the contents of that address in the data register.
4. Compare the value in the data register with the value for that address in Table B–8.
5. If the values are the same, press EXAM again to display the contents of the next address. If the values are not the same, repeat the entire procedure for depositing the bootstrap. Verify the contents of all the addresses in this way. If any instruction is incorrect, repeat the entire deposit procedure.

Once you have correctly deposited the bootstrap in memory, start the computer as follows:

1. Set the starting address, 002000, in the switch register.
2. Press the LOAD ADDR switch.
3. Set the ENABLE/HALT switch to ENABLE.
4. Press the START switch.

B.8 Loading the TSV05 Bootstrap Using MICRO-ODT

Deposit the TSV05 bootstrap loader in memory as follows:

1. Turn on your processor; if it is already on, halt it.
2. At the console, type the first address in Table B–9 (7776) followed by a slash (/):
   
   7776/

   The system responds by printing the contents of address 7776 (represented below by xxxxxxx) on the console:

   7776/ xxxxxxx

3. On the same line, type the first contents value from Table B–9 (46523) followed by a line feed.

   7776/ xxxxxxx 46523

   The system deposits the value you type and displays the next memory location.

   10000/

B–12 Loading Software Bootstraps
4. Type the contents for the next memory location (Table B–9) followed by a line feed.

5. Repeat step 4 until you have deposited all the instructions. Then, type a carriage return.

6. Finally, type:

```
100000G
```

The processor reads the software bootstrap from the magtape into memory. The system prints the following prompt when it is finished:

```
MSBOOT V05.xx
*
```

7. Respond to the asterisk (*) as described in Chapter 8. Wherever MM or MT appears in the instructions, substitute MS.

**Table B–9: TSV05 Bootstrap Loader**

<table>
<thead>
<tr>
<th>Address</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>007776</td>
<td>046523</td>
</tr>
<tr>
<td>100000</td>
<td>012701</td>
</tr>
<tr>
<td>010002</td>
<td>172522</td>
</tr>
<tr>
<td>010004</td>
<td>010102</td>
</tr>
<tr>
<td>010006</td>
<td>005000</td>
</tr>
<tr>
<td>010010</td>
<td>105711</td>
</tr>
<tr>
<td>010012</td>
<td>100376</td>
</tr>
<tr>
<td>010014</td>
<td>010704</td>
</tr>
<tr>
<td>010016</td>
<td>112737</td>
</tr>
<tr>
<td>010020</td>
<td>000200</td>
</tr>
<tr>
<td>010022</td>
<td>172523</td>
</tr>
<tr>
<td>010024</td>
<td>005242</td>
</tr>
<tr>
<td>010026</td>
<td>105711</td>
</tr>
<tr>
<td>010030</td>
<td>100376</td>
</tr>
<tr>
<td>010032</td>
<td>005711</td>
</tr>
<tr>
<td>010034</td>
<td>100761</td>
</tr>
<tr>
<td>010036</td>
<td>005007</td>
</tr>
</tbody>
</table>

**B.9 Loading the TSV05 Bootstrap Using the Switch Register**

Deposit the TSV05 bootstrap loader in memory as follows:

1. Set the ENABLE/HALT switch to HALT.

2. Set the first address in Table B–9, 7776, in the switch register.

3. Press the LOAD ADDR switch.

4. Set the contents for the first address (46523 from Table B–9) in the switch register.
5. Lift the DEP switch.
   The computer automatically advances to the next address.

6. Set the contents for the next address (from Table B–9) in the switch register.

7. Lift the DEP switch.

8. Repeat steps 6 and 7 until you have deposited all the instructions.

Now verify that you deposited the bootstrap loader properly.

1. Set the first address, 7776, in the switch register.

2. Press the LOAD ADDR switch.

3. Press the EXAM switch to display the contents of that address in the data register.

4. Compare the value in the data register with the contents value for that address in Table B–9.

5. If the values are the same, press EXAM again to display the contents of the next address. If the values are not the same, repeat the entire procedure for depositing the bootstrap. Verify the contents of all addresses in this way. If any instruction is incorrect, repeat the entire deposit procedure.

Once you have correctly deposited the bootstrap in memory, start the computer as follows:

1. Set the starting address, 10000, in the switch register.

2. Press the LOAD ADDR switch.

3. Set the ENABLE/HALT switch to ENABLE.

4. Press the START switch.
INDEX

Absolute base address p-sects
  changing number of, 2–25
Auto installation file
  list of (table), 2–6

BA.MAC, 2–6
Bad block replacement
  in user-written handlers, 2–50
Banner pages
  changing default width of, 2–48
Base-line monitor
  See BL monitor
BATCH.SAV system utility, 2–5
  modifying to save disk space,
    2–33
BINCOM.SAV system utility, 2–4
BL monitor
  description of, 1–6
  features of (table), 1–3
  vs SJ monitor (table), 1–6
BL.ANS, 2–5
Bootstrap
  installing on diskette, 6–8
  installing on RX02, 7–6
  installing on working system,
    3–9, 4–7, 5–7, 8–8
  loading software, B–1
  magtape
    list of (table), 2–6
Bootstrap loader
  RK05 address and contents, B–2
  RK06 address and contents, B–8
  RK07 address and contents, B–8
  RL01 address and contents, B–9
  RL02 address and contents, B–9
  RX11 address and contents, B–4
  RX211 address and contents,
    B–11
  TC11 address and contents, B–3
  TJU16 address and contents, B–6
  TM11 address and contents, B–7
  TSV05 address and contents,
    B–13
Bootstrap message
  changing, 2–42
  suppressing, 2–41
Bootstraping the system
  in less than 16K words, 2–37
BSTRAP.MAC, 2–6
BUP.SAV system utility, 2–4
CAPS–11
  using to load files, 2–40
Cartridges
  installing system to run on,
    3–1 to 3–16
Clock rate
  specifying 50-cycle, 2–40
Components
  arranging on working system,
    2–15
CONFIG.COM, 2–8
CONFIG.SAV, 2–8
CONSOL.MAC, 2–8
CR.MAC, 2–6
CR.SYS device handler, 2–2
CREF.SAV system utility, 2–4
  changing page listing length,
    2–53
CT.MAC, 2–6
CT.SYS device handler, 2–2
CUSTOM.TXT, 2–8
  customization symbol values,
    2–22
Customizations
  for distributed monitors, 2–22
to 2–58
  software
    getting symbol values for,
      2–22
50-cycle clock rate, 2–40
  restriction for Professional
    300 series, 2–40
D keyboard command
  use of above background job,
    2–45
DATIME.MAC, 2–8
DATIME.SAV, 2–8
DD.MAC, 2–6
DD.SYS device handler, 2–2
DDX.SYS device handler, 2–2
Debuggers
  list of (table), 2–6
DECtape II
  block locations on, 2–18
  changing CSR addresses in, 2–30
  changing vectors in, 2–30
  improving response time on,
    2–17
  installing system to run on,
    3–1 to 3–16

Index–1
Default device
  assigning, 2–15
  changing for EDIT command, 2–44
  changing for FRUN command, 2–43
  changing for indirect command files, 2–42
  changing for QUEMAN, 2–45
DEMOB.G.MAC, 2–6
DEMOED.TXT, 2–6
DEMOF1.FOR, 2–6
DEMOFG.MAC, 2–6
Demonstration programs
  creating volumes for, 2–18
  list of (table), 2–6
DEMOX1.MAC, 2–6
Density
  changing for magtape, 2–33
Device
  assigning default to data, 2–15
Device handlers
  choosing for peripheral devices, 2–11
  choosing for working system, 2–11
  controller installed at
    nonstandard address, 2–31
  list of (table), 2–2
  renaming, 2–30
  user-written
    bad block replacement for, 2–50
    for magtapes, 2–51
    when bootstrap doesn’t install, 2–30
Devices
  installing nonstandard, 2–29
  sizes of (table), 2–9
DF03 modem
  using with VTCOM, 2–39
DIAL command
  using with VTCOM, 2–39
DIBOL
  LINK change requirement, 2–26
DIFFERENCES/CHANGEBAR command, 2–23
DIR.SAV system utility, 2–4
  changing default number of columns in, 2–24
  changing default order of listings, 2–24
DIRECTORY command, 2–9
Directory listings
  changing default number of columns in, 2–24
  reading, 2–9
  sorting
    changing default of, 2–24
Directory segments
  changing default number of, 2–47
Disk distribution kit
  backing up, 5–4, 6–3
  installing
    bootstrapping distribution volume, 5–3, 6–2
    installing on disk, 5–1 to 5–14
    installing on small device, 6–1 to 6–16
Diskettes
  installing system to run on,
    3–1 to 3–16, 6–1 to 6–16
  sizes of (table), 2–9
Disks
  if SYSGEN required, 4–15
  installing system to run on,
    4–1 to 4–15, 5–1 to 5–14
  sizes of (table), 2–9
DISMT1.COM, 2–5
  replacing magtape bootstrap in, 2–52
DISMT2.COM, 2–5
Distribution kit
disk
  installing on disk, 5–1 to 5–14
  installing on small device, 6–1 to 6–16
  files on (table), 2–2
magtape
  installing on disk, 8–1 to 8–16
RX01
  installing on disk, 4–1 to 4–15
  installing on small devices, 3–1 to 3–16
RX02
  installing on RX02, 7–1 to 7–14
RX50
  installing on MICRO/PDP–11, 9–1 to 9–13
  installing on Professional
    325, 10–1 to 10–13
  installing on Professional
    350, 11–1 to 11–12
DL.MAC, 2–6
DL.SYS device handler, 2–2
DLX.SYS device handler, 2–2
DM.MAC, 2–6
DM.SYS device handler, 2–2
DMX.SYS device handler, 2–2
DP.MAC, 2–7
DP.SYS device handler, 2–2
DS.MAC, 2–7
DS.SYS device handler, 2–2
DT.MAC, 2–7
DT.SYS device handler, 2–2
DU.MAC, 2–7
D.U.SYS device handler, 2–2
DUMP.SAV system utility, 2–4
DUP.SAV system utility, 2–4
DUX.SYS device handler, 2–2
DW.MAC, 2–7
DW.SYS device handler, 2–2
DWX.SYS device handler, 2–2
DX.MAC, 2–7
DX.SYS device handler, 2–2
DXX.SYS device handler, 2–3
DY.MAC, 2–7
D.Y.SYS device handler, 2–3
DYX.SYS device handler, 2–3
DZ.MAC, 2–7
D.Z.SYS device handler, 2–3
DZX.SYS device handler, 2–3

Fatal errors
preventing reset from, 2–37

FB monitor
description of, 1–6
features of (table), 1–3
FB.MAC, 2–7

File types
default
changing for FRUN command, 2–44
changing for indirect command files, 2–43

Files
distributed
list of (table), 2–2
loading with CAPS–11, 2–40
size of
setting upper limit on, 2–39
FILEX.SAV system utility, 2–4
Foreground/background monitor
See FB monitor
FORLIB.OBJ
when combining with SYSLIB.OBJ, 2–28
FORMAT.SAV system utility, 2–4
FRUN keyboard command
changing default device for, 2–43
file type
changing default, 2–44
FS.MAC, 2–7

GETSTR.FOR, 2–6

Graphics
list of files for (table), 2–6

Handler
See Device handlers

Handler files
magtape (table), 2–33

Hardware magtapes
installing support for, 2–31
HELP file
changing default device for, 2–49
changing default file name for, 2–49
HELP package, 2–12
Help text
modifying, 2–35
HELP.EXE
creating from HELP.SAV, 2–35
HELP.SAV system utility, 2–4
HELP.TXT
creating from HELP.SAV, 2–35

E keyboard command
use of above background job, 2–45

EDIT keyboard command
changing default file name for, 2–45
file type
changing default, 2–44
EDIT.SAV system utility, 2–4
changing size of text window for, 2–27
customizations for, 2–27
with nonstandard ESCAPE codes, 2–27

Editors
See Text editors
EDTGBL.MAC, 2–7
EL.MAC, 2–7
ELCOPY.MAC, 2–7
ELINIT.MAC, 2–7
ELTASK.MAC, 2–7

Errors
fatal
preventing reset from, 2–37
ERROUT.OBJ, 2–7
ERROUT.SAV, 2–5
ERRTXT.MAC, 2–7

Extended memory monitor
See XM monitor

Index–3
I/O page
  depositing in, 2–45
  examining, 2–45
IND.SAV system utility, 2–4
  changing default device for, 2–50
  changing default to, 2–53
Indirect command files
  changing default device for, 2–42
file type
  changing default, 2–43
nesting
  changing depth, 2–46
start-up
  choosing for working system, 2–14
  suppressing, 2–41
  suppressing echo, 2–42
using IND with, 2–53
Installation
  customization
    features available through, 1–8
disk distribution kit
  backing up, 5–4 to 5–5, 6–3
to 6–5
  bootstrapping, 5–3, 6–2
  copying bootstrap to system volume, 5–5, 6–5
  initializing backup volume, 5–4, 6–3
  unprotecting backup volumes, 5–5, 6–5
disk distribution kit to disk,
  5–1 to 5–14
disk distribution kit to small device, 6–1 to 6–16
magtape distribution kit
  backing up, 8–4 to 8–6
  bootstrapping, 8–2
magtape distribution kit to disk,
  8–1 to 8–16
  if SYSGEN required, 8–16
preparation for, 1–1
procedure, 1–1
reading path for, 1–18
RX01 distribution kit
  backing up, 3–3 to 3–7, 4–3
to 4–5
  bootstrapping, 3–2, 4–2
  copying bootstrap to system disk, 4–5
  copying bootstrap to system volume, 3–4
if SYSGEN required, 3–16
initializing backup disk, 4–3
initializing backup volumes,
  3–3
squeezing system volume, 3–4
unprotecting backup volumes,
  3–6, 4–5
RX01 distribution kit to disk,
  4–1 to 4–15
RX01 distribution kit to small
device, 3–1 to 3–16
RX02 distribution kit
  bootstrapping, 7–2
  unprotecting distribution volume, 7–3
RX02 distribution kit to RX02,
  7–1 to 7–14
RX50
  copying distribution diskettes onto disk, 9–4,
    11–4
  copying system diskette onto disk, 9–3, 11–4
RX50 distribution kit
  bootstrapping, 9–2, 9–4, 10–2,
    11–2, 11–4
survey of procedures, 2–1
test of for MICRO/PDP-11
  working system, 9–8 to 9–13
test of for Professional 325
  working system, 10–8 to 10–13
test of for Professional 350
  working system, 11–7 to 11–12
test of working system on disk,
  4–10 to 4–15, 5–10 to 5–14,
  8–10 to 8–16
test of working system on RX02,
  7–8 to 7–13
test of working system on small device, 3–11 to 3–16, 6–11
to 6–16
Installation worksheet, 2–58
Introduction to RT-11
  creating volumes for, 2–18
  files required for, 2–12
IVP.MAP, 2–6
K52.SAV system utility, 2–4
KED.SAV system utility, 2–4
  modifying default file type,
    2–56
KEK.SAV system utility
limiting memory requested by, 2–54
Keypad editor
See K52.SAV system utility
See KED.SAV system utility
KMON.MAC, 2–7
KMOVLY.MAC, 2–7

LD.MAC, 2–7
LD.SYS device handler, 2–3
Also see Logical disk files
used with magtape, 2–52
LDX.SYS device handler, 2–3
LIBR.SAV system utility, 2–4

Libraries
list of (table), 2–6
LIBRARY command, 2–11

Library modules
increasing number of for LINK, 2–25

Line printer handlers
changing CSR addresses in, 2–30
changing vectors in, 2–30
choosing for working system, 2–13
LINK.SAV system utility, 2–4
changing default output to
132-columns, 2–57
changing size of library module
list, 2–25
changing size of library module
list of, 2–25
modifying listing page length, 2–49
modifying to change default
SYSLIB device, 2–34

Load maps
changing default page length
for, 2–49
Logical disk files
changing default file type, 2–52
LP.MAC, 2–7
LP.SYS device handler, 2–3
choosing, 2–13
renaming, 2–13
LPX.SYS device handler, 2–3
LS.MAC, 2–7
LS.SYS device handler, 2–3
choosing, 2–13
renaming, 2–13
LSX.SYS device handler, 2–3

MACRO assembler
files required for, 2–13
MACRO.SAV system utility, 2–4
changing default
.ENABLE/.DISABLE options, 2–55
changing default .LIST/.NLIST
options, 2–54
changing page listing length, 2–53
Magtape distribution kit
backing up, 8–4
installing
bootstrapping distribution
volume, 8–2
installing on disk, 8–1 to 8–16
Magtapes
changing parity and density of,
2–33
handler files (table), 2–33
hardware
installing support for, 2–31
replacing bootstrap for in
DISMT1.COM, 2–52
user-written handlers for, 2–51
MBOOT.BOT, 2–6
MBOT16.BOT, 2–6
MDUP.MM system utility, 2–4
device support, A–1
MDUP.MS system utility, 2–4
device support, A–1
MDUP.MT system utility, 2–4
device support, A–1
MDUP.SAV system utility, 2–4
nonstandard device
support, modifying for, A–1
Media
calculating sufficient, 2–21
Memory
running in less than available, 2–37
size of
setting upper limit on, 2–41
MICRO/PDP–11
installing on, 9–1 to 9–13
if SYSGEN required, 9–13
MM.SYS device handler, 2–3
MMHD.SYS hardware magtape
handler, 2–32
MMHDXX.SYS hardware magtape
handler, 2–32
MMX.SYS device handler, 2–3
Monitor map files
customization symbol values, 2–22
Monitors
See also BL monitor, FB monitor, SJ monitor, and XM monitor
customizations available for
descriptions of, 2–22 to 2–58
customizations available for
table, 1–8
depositing in, 2–45
distributed
choice of, 1–3
features available in (table), 1–3
list of (table), 2–2
examining, 2–45
selection of for working system,
2–10
system generation options
available for (table), 1–15
MS.SYS device handler, 2–3
MSBOOT.BOT, 2–6
MSCP disks
changing CRS addresses in, 2–30
changing vectors in, 2–30
MSHD.SYS hardware magtape handler,
2–32
MSHDX.SYS hardware magtape
handler, 2–32
MSX.SYS device handler, 2–3
MT.SYS device handler, 2–3
MTHD.SYS hardware magtape handler,
2–32
MTHDX.SYS hardware magtape
handler, 2–32
MTTEM.T.MAC, 2–7
MTINT.MAC, 2–7
MTX.SYS device handler, 2–3

Nesting depth
indirect command files
changing, 2–46
NL.MAC, 2–7
NL.SYS device handler, 2–3
NLX.SYS device handler, 2–3

ODT.OBJ, 2–6
OHANDL overlay handler, 2–11
Output device
changing default to terminal,
2–23
Output ring buffer
increasing space requirement in
to resume job, 2–46
Overlaid utilities
list of, 2–16

P-sects
absolute base address
changing number of, 2–25
Parity
changing for magtape, 2–33
PAS.TAV system utility, 2–4
PC.MAC, 2–7
PC.SYS device handler, 2–3
PD.MAC, 2–7
PD.SYS device handler, 2–3
PI.SYS device handler, 2–3
PI.PAV system utility, 2–4
PI.X.SYS device handler, 2–3
Professional 325
installing on, 10–1 to 10–13
bootstrap, 10–5
creating working system, 10–3
customizing, 10–6
if SYSGEN required, 10–13
Professional 350
installing on, 11–1 to 11–12
if SYSGEN required, 11–12
Program sections
See P-sects
PUTSTR.FOR, 2–6

QUEMAN.SAV system utility, 2–5
changing default device for,
2–45
choosing for working system,
2–13
QUEUE work file
changing size of, 2–26
QUEUE.REL system utility, 2–5
changing size of work file,
2–26
choosing for working system,
2–13
QUEUE.SAV system utility
allowing first form feed, 2–52

RA80 disk
device name, 2–9
size of, 2–9
RC25 disk
device name, 2–9
size of, 2–9
RD50 disk
device name, 2–9
size of, 2–9
RD51 disk
device name, 2–9
size of, 2–9
Reading path
for installation, 1–18
RESORC.SAV system utility, 2–5
RF.MAC, 2–7
RF.SYS device handler, 2–3
Ring buffer
   output
      increasing space requirement
   in to resume job, 2–46
RK.MAC, 2–7
RK.SYS device handler, 2–3
RK05
   loading bootstrap, B–1
RK05 disk
   device name, 2–9
   size of, 2–9
RK06
   loading bootstrap, B–7
RK06/RK07 disk
   device name, 2–9
   size of, 2–9
RK07
   loading bootstrap, B–7
RK11 DECpack
   See RK05
RKX.SYS device handler, 2–3
RL01
   loading bootstrap, B–8
RL01/RL02 disk
   device name, 2–9
   size of, 2–9
RL02
   loading bootstrap, B–8
RMONFB.MAC, 2–8
RMONSJ.MAC, 2–8
RP03 magtape support, 2–33
RT11.AI.SYS monitor, 2–2
RT11.BL.SYS monitor, 2–2
RT11.PL.SYS monitor, 2–2
RT11.SJ.SYS monitor, 2–2
RTBL.MAP, 2–5
RTMON.REL, 2–8
RTSJ.MAP, 2–5
RTXM.MAP, 2–5
RX01 diskette
   device name, 2–9
   initializing and formatting, 6–7
   size of, 2–9
RX01 distribution kit
   backing up, 3–3, 4–3
   installing
      bootstrapping distribution
      volume, 3–2, 4–2
   installing on disk, 4–1 to 4–15
   installing on small devices, 3–1 to 3–16
RX02 diskette
   device name, 2–9
   initializing and formatting, 6–7
   size of, 2–9
RX02 distribution kit
   installing
      bootstrapping distribution
      volume, 7–2
   installing on RX02, 7–1 to 7–14
      if SYSGEN required, 7–14
RX11
   loading bootstrap, B–4
RX211
   loading bootstrap, B–10
RX50 diskette
   device name, 2–9
   initializing, 6–6
   size of, 2–9
RX50 distribution kit
   installing
      bootstrapping system disk, 9–4, 11–4
      bootstrapping system diskette, 9–2, 10–2, 11–2
   copying distribution
   diskettes onto disk, 9–4, 11–5
   copying system diskette onto disk, 9–3, 11–4
   installing on MICRO/PDP–11, 9–1 to 9–13
   installing on Professional 325, 10–1 to 10–13
   installing on Professional 350, 11–1 to 11–12
.SETTOP programmed request
   used with KEX, 2–54
SETUP.SAV system utility, 2–5
Single-job monitor
   See SJ monitor
SIPP.SAV system utility, 2–5
   installing customization symbol
   values, 2–22
SJ monitor
   description of, 1–6
   features of (table), 1–3
SJ.MAC, 2–8
SJPB.ANS, 2–5
SL.SYS pseudohandler, 2–4
SLP.SAV system utility, 2–5
SLX.SYS pseudohandler, 2–4
Software customizations, 2–22 to 2–58

Index–7
choosing, 2–21
getting symbol values for, 2–22
Software kit
distributed components of, 2–2
Software updates
backing up magtape, 8–6
installing, 2–20, 3–7, 4–6, 5–6, 6–5, 7–4, 8–6, 9–3, 10–3, 11–3
Source files
choosing for working system, 2–14
list of (table), 2–6
SP.MAC, 2–8
SP.SYS device handler, 2–3
SPLIT.SAV, 2–8
SPOOL.REL system utility, 2–5
changing output device, 2–57
changing work file size, 2–56
choosing for working system, 2–13
SPOOL.SYS
created by SPOOL.REL, 2–56
SPX.SYS device handler, 2–3
SRCCOM.SAV system utility, 2–5
changing insertion/deletion
characters used by, 2–22
/D option, 2–23
STARTA.COM, 2–6
STARTF.COM, 2–5
STARTS.COM, 2–5
STARTX.COM, 2–5
Subroutines
list of (table), 2–6
SWAP.SYS
need for, 2–10
SYSGEN.COM, 2–5
SYSLIB.OBJ system library, 2–6
changing default device of, 2–34
extracting overlay handlers from, 2–28
need for, 2–11
SYSMAC.SML system MACRO library, 2–5
System generation
features available (table), 1–15
list of files for (table), 2–5
need for, 1–14
System Library
See SYSLIB.OBJ
System utilities
minimum on system volume, 2–20
System volume
creating, 2–18
example of, 2–17
including overlaid utilities, 2–16
limiting components on, 2–20
TC11
loading bootstrap, B–2
TERMID.SAV, 2–6
Terminal
changing default output to, 2–23
Text editors
choosing for working system, 2–14
TJ.MAC, 2–8
TJU16
loading bootstrap, B–5
TM.MAC, 2–8
TM11
loading bootstrap, B–5
TRANSF.SAV system utility, 2–5
TRMTBL.MAC, 2–8
TS.MAC, 2–8
TSV05
loading bootstrap
MICRO–ODT, B–12
switch register, B–13
TT.MAC, 2–8
TT.SYS device handler, 2–3
UCL.SAV system utility, 2–5
Unsupported software
list of (table), 2–8
Updates
backing up magtape, 8–6
installing, 2–20, 3–7, 4–6, 5–6, 6–5, 7–4, 8–6, 9–3, 10–3, 11–3
USR.MAC, 2–8
Utilities volume
creating a, 2–16
example of, 2–17
Utility programs
choosing for working system, 2–15
creating volume for, 2–16
V5NOTE.TXT, 2–8
V5USER.TXT, 2–5
VBGEXE.SAV, 2–8
VDT.OBJ, 2–6
Vectors
VT11/VS60
changing, 2–23
VERIFY.COM, 2–6
VHANDL overlay handler, 2–11
Virtual memory
device name, 2–9
size of, 2–9
VM.MAC, 2–8
VM.SYS device handler, 2–3
VMX.SYS device handler, 2–3
Volumes
system
creating, 2–18
VT11/VS60
changing vector address, 2–23
VTCOM.REL system utility, 2–5
setting default DIAL string, 2–39
VTCOM.SAV system utility, 2–5
setting default DIAL string, 2–39
VTHDLR.OBJ, 2–6
VTMAC.MAC, 2–6, 2–8

Working system
arrangement of components for,
2–15
on disk
backing up, 4–8 to 4–9, 5–9,
8–9 to 8–10
compressing, 4–7, 5–8, 8–8
creating from disk, 5–6
creating from magtape, 8–7
creating from RX01, 4–6
installing bootstrap on, 4–7,
5–7
installing customizations on,
4–7, 5–8, 8–8
protecting, 4–8
test of installation, 4–10 to
4–15, 5–10 to 5–14, 8–10
to 8–16
on MICRO/PDP–11
backing up, 9–6 to 9–8
compressing, 9–6
creating, 9–5
installing customizations on,
9–5
testing installation, 9–8 to
9–13
on Professional 325, 10–3
backing up, 10–7
testing installation, 10–8 to
10–13
unprotecting, 10–6
on Professional 350
backing up, 11–6
compressing, 11–5
creating, 11–5
installing customizations on,
11–5
testing installation, 11–7 to
11–12
on RX02
backing up, 7–8
compressing, 7–7
creating from RX02, 7–4 to
7–6
installing bootstrap on, 7–6
installing customizations on,
7–7
test of installation, 7–8 to
7–13
on small device
backing up, 3–10, 6–10
compressing, 3–10, 6–9
creating from disk, 6–6 to
6–7
creating from RX01, 3–7
installing bootstrap on, 3–9,
6–8
installing customizations on,
3–9, 6–9
test of installation, 3–11 to
3–16, 6–11 to 6–16
selecting components for, 2–8

XC.MAC, 2–8
XC.SYS device handler, 2–4
XCX.SYS device handler, 2–4
XL.MAC, 2–8
XLX.SYS device handler, 2–4
XM monitor
description of, 1–7
features of (table), 1–3
XM.ANS, 2–5
XM.MAC, 2–8
XMSUBS.MAC, 2–8
# HOW TO ORDER
## ADDITIONAL DOCUMENTATION

<table>
<thead>
<tr>
<th>From</th>
<th>Call</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chicago</td>
<td>312–640–5612</td>
<td>Digital Equipment Corporation</td>
</tr>
<tr>
<td></td>
<td>8:15 AM to 5:00 PM CT</td>
<td>Accessories &amp; Supplies Center</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1050 East Remington Road</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Schaumburg, IL 60195</td>
</tr>
<tr>
<td>San Francisco</td>
<td>408–734–4915</td>
<td>Digital Equipment Corporation</td>
</tr>
<tr>
<td>Alaska, Hawaii</td>
<td>603–884–6660</td>
<td>Accessories &amp; Supplies Center</td>
</tr>
<tr>
<td></td>
<td>8:30 AM to 6:00 PM ET</td>
<td>632 Caribbean Drive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sunnyvale, CA 94086</td>
</tr>
<tr>
<td></td>
<td>or 408–734–4915</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8:15 AM to 5:00 PM PT</td>
<td></td>
</tr>
<tr>
<td>New Hampshire</td>
<td>603–884–6660</td>
<td>Digital Equipment Corporation</td>
</tr>
<tr>
<td></td>
<td>8:30 AM to 6:00 PM ET</td>
<td>Accessories &amp; Supplies Center</td>
</tr>
<tr>
<td>Rest of U.S.A.</td>
<td>1–800–258–1710</td>
<td>P.O. Box CS2008</td>
</tr>
<tr>
<td>Puerto Rico*</td>
<td>8:30 AM to 6:00 PM ET</td>
<td>Nashua, NH 03061</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Prepaid orders from Puerto Rico must be placed with the local DIGITAL subsidiary (call 809–754–7575)</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Canada</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>British Columbia</td>
<td>1–800–267–6146</td>
<td>Digital Equipment of Canada Ltd</td>
</tr>
<tr>
<td></td>
<td>8:00 AM to 5:00 PM ET</td>
<td>940 Belfast Road</td>
</tr>
<tr>
<td>Ottawa–Hull</td>
<td>613–234–7726</td>
<td>Ottawa, Ontario K1G 4C2</td>
</tr>
<tr>
<td></td>
<td>8:00 AM to 5:00 PM ET</td>
<td>Attn: A&amp;SG Business Manager</td>
</tr>
<tr>
<td>Elsewhere</td>
<td>112–800–267–6146</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8:00 AM to 5:00 PM ET</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Elsewhere</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Digital Equipment Corporation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A&amp;SG Business Manager*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>*c-o DIGITAL’s local subsidiary or approved distributor</td>
</tr>
</tbody>
</table>


READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

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

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

Did you find errors in this manual? If so, specify the error and the page number.

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

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

- Assembly language programmer
- Higher-level language programmer
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Other (please specify) ________________________________________________

Name __________________________________________ Date ___________________

Organization __________________________________ Telephone __________________

Street __________________________________________

City __________________________ State __________ Zip Code __________ or Country