RT–11 System Release Notes

AA–5286I–TC

July 1986

This manual summarizes the features that differentiate the RT–11 V5.0, V5.1, V5.2, V5.3, and V5.4 operating systems from RT–11 V4.0.

This manual supersedes *RT–11 System Release Notes*, AA–5286H–TC.

Operating System and Version: RT–11 Version 5.4

digital equipment corporation · maynard, massachusetts

First Printing, March 1980
Updated, June 1980
Revised, March 1981
Revised, March 1983
Revised, December 1983
Updated, July 1984
Revised, April 1985
Revised, November 1985
Revised, July 1986

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

The following are trademarks of Digital Equipment Corporation:

DEC	MASSBUS	RT-11
DECmate	PDP	UNIBUS
DECnet	P/OS	VAX
DECsystem-10	Professional	VMS
DECSYSTEM-20	Q-BUS	VT
DECUS	Rainbow	Work Processor
DECwriter	RSTS	
DIBOL	RSX	digital

CONTENTS

PREFACE

This manual describes how the RT-11 V5.4 operating system differs from RT-11 V4.0, V5.0, V5.1, V5.2, and V5.3.

Changes made to RT-11 after the publication of the V5.3 RT-11 System Release Notes, AA-5286H-TC, and changes for V5.4 are highlighted by change bars. In software version numbers, a whole number (X) represents a major release and all its updates, and a mixed number (X.X) represents a major release or a specific update of a major release. For example, references to V5 apply to V5.4 as well, but references to V5.4 apply only to V5.4.

References to RT-11 documentation other than parts of this manual do not apply to new material (indicated by change bars).

The system manager and system programmers should be thoroughly familiar with the contents of this manual before performing a system generation as described in the RT-11 System Generation Guide.

This manual describes new features and corrected problems and presents a comparison between this release of RT-11 and the previous releases:

- Chapter 1 - NEW FEATURES

  This chapter describes the new processors, devices, software components, and documentation.

- Chapter 2 - CHANGES AND ADDITIONS TO EXISTING COMPONENTS

  This chapter describes software components that have been improved through changed or added capabilities.

- Chapter 3 - CURRENT RESTRICTIONS AND CORRECTED PROBLEMS

  This chapter describes current software restrictions and documentation problems and lists software problems that have been corrected.

- Chapter 4 - INSTALLATION, BOOTSTRAP, AND HARDWARE SETUP PROCEDURES

  This chapter tells you how to access on-line information that describes customizations you can use to improve system performance. The chapter also describes formatting, installation, bootstrap, and hardware setup procedures you may need depending on your configuration.

- Chapter 5 - NATIVE TRANSFER UTILITY (Unsupported)

  This chapter tells you how to use the unsupported native TRANSFER utility. TRANSFER, released with RT-11 V5.3, copies files between a processor running RT-11 and another processor running RSX or VAX/VMS.

- Appendix A lists V5.4 and V5.3 error messages, with causes and solutions for each.

- Appendix B explains the guidelines for submitting a software performance report (SPR) to DIGITAL.

- Appendix C describes how to run RT-11 from the virtual memory device (VM). Many processors are capable of creating a large enough VM device to run RT-11. Running RT-11 from VM can greatly increase the performance of your processor.

NOTE

A machine-readable addendum to this document, V5NOTE.TXT, is provided on the distribution kit. V5NOTE.TXT describes changes that have occurred since this document was printed.

Information on unsupported utilities, previously in the file UNSUP.TXT, is now in Appendix C of the RT-11 System Utilities Manual.

The term unsupported as used in reference to a distributed module means that DIGITAL does not guarantee that future releases will be necessarily compatible with previous versions of that module, and does not guarantee that module will appear in future releases of RT-11. However, SPRs (Software Performance Reports) are answered for any module distributed with RT-11.

CHAPTER 1

NEW FEATURES

The primary goals of RT-11 V5 are to support additional processors and devices, improve and extend the functionality of existing software components, correct existing software problems, and increase the ease of RT-11 installation and maintenance.

This chapter summarizes new hardware that RT-11 supports and its new software components. Refer to the rest of the manuals in the documentation set for detailed descriptions of how to use these new features.

1.1 NEW PROCESSORS

RT-11 V5 runs on four new processors:

T-11 Falcon (SBC-11/21 and SBC-11/21 PLUS)

F-11 Microprocessor Based Board and Systems

Professional 300 Series (PC325, PC350, and PC380)

J-11 Microprocessor Based Boards and Systems

1.1.1 T-11 Falcon (SBC-11/21 and SBC-11/21 PLUS)

The new T-11 processor chip is supported by the RT-11 single-job (SJ) and foreground/background (FB) monitors in the SBC-11/21 PLUS, single-board computer configuration. The original SBC-11/21 is also supported by the SJ and FB monitors.

1.1.2 F-11 Microprocessor Based Board and Systems

The PDP-11/23 PLUS is supported by RT-11 V5, including full 22-bit addressing (up to 4M bytes) under the XM monitor. The PDP-11/23 PLUS processor is available in both the standard PDP-11/23 PLUS system configuration and in the new MicroPDP-11/23 configuration.

1.1.3 Professional 300 Series (PC325, PC350, and PC380)

The Professional 300 series computers are supported by RT-11. Although the foreground/background (FB) monitor runs on Professional computers with some restrictions (see Chapter 3), the extended memory

(XM) monitor is better suited for running RT-11 on Professional computers. The XM monitor supports full RT-11 functionality, including 22-bit addressing, on Professional computers. The SJ and BL monitors are not supported on Professional 300 series computers.

1.1.4 J-11 Microprocessor Based Boards and Systems

RT-11 supports the J-11 microprocessor, including full 22-bit addressing (up to 4M bytes) under the XM monitor.

The KDJ11-A (J-11 CPU board) is a supported upgrade option to the PDP-11/23. Also supported are the MicroPDP-11/73 and MicroPDP-11/83 Q-BUS processor systems and the PDP-11/84 UNIBUS processor system.

RT-11 V5.4 supports the new MicroPDP-11/53 (KDJ11-DA) processor system. See Section 1.2.4 for a description of the new RX33 diskette drive, and Section 4.3.1 for a description of the installation procedure you must follow to install RT-11 on the MicroPDP-11/53.

1.2 NEW MASS STORAGE DEVICES

RT-11 V5 supports the following new mass storage devices:

RA80/RA81 disks
RC25 disk
RD31/RD50/RD51/RD52/RD53 disks
RX33/RX50 diskette
TK25 magtape
TK50 magtape
TS05 (TSU05/TSV05) magtapes

1.2.1 RA80/RA81 Disks (DU)

RA80

The RA80 disk is a 124M-byte fixed hard disk supported by the MSCP disk class handler, DU.

Because the RA80 disk contains more than 64K blocks, it must be divided into multiple 64K-block partitions. When running under RT-11, each partition operates as a separate disk. Partition 0 can be used as a system volume or for data storage. All other partitions can be used for data storage only.

The RA80 disk is now supported for Q-BUS processors using the KDA50 controller.

RA81

The RA81 is a 406M-byte fixed hard disk. Because the RA81 contains more than 64K blocks, it must be divided into multiple 64K-block partitions. Each partition operates as a separate disk. Partition 0 can be used as a system volume or for data storage; all other partitions can be used for data storage only. You can divide the RA81 into a maximum of eight partitions. Therefore, at any given time you

can directly address up to 262M bytes using that method. You must use the JREAD and JWRITE subroutines to read and write to the rest of the disk.

The RA81 is supported for Q-BUS and UNIBUS processors.

1.2.2 RC25 Disk (DU)

The RC25 disk is a 26M-byte disk supported by the MSCP handler, DU. RC25 disk drives are always paired. The even-numbered drive accepts a removable disk, and the odd-numbered drive contains a fixed RC25 disk for a total storage of 52M bytes per pair. RC25 disks can be used as the system volume or for data storage.

1.2.3 RD31/RD50/RD51/RD52/RD53 Disks (DU or DW)

The RD31/RD50/RD51/RD52/RD53 disks are fixed hard disks available for the MicroPDP-11 and Professional 300 series processors.

The MicroPDP-11 can include a 10M-byte RD51 disk, a 20M-byte RD31 disk, a 33M-byte RD52 disk, or a 71M-byte RD53 disk. Each disk is an MSCP device supported by the MSCP disk class handler, DU.

Because the RD53 disk contains more than 64K blocks, it must be divided into multiple 64K-block partitions. Each partition operates as a separate disk. Partition 0 can be used as a system volume or for data storage. All other partitions can be used for only data storage.

Professional 350 and Professional 380 systems can include a 5M-byte RD50 disk, 10M-bytes only support for the 20M-byte RD31 disk, a 10M-byte RD51 disk, or a 33M-byte RD52 disk. The RD31, RD50, RD51, and RD52 are supported by the DW handler. The RD53 is not supported by the DW handler.

RD31/RD50/RD51/RD52/RD53 disks can be used as system or data volumes.

1.2.4 RX33 Diskette Drive (DU)

The RX33 is a half-height diskette drive that supports RX33 double-sided 1.2M-byte diskettes and RX50 single-sided 400K-byte diskettes. The RX33 diskette drive is available only for Q-BUS processors such as the MicroPDP-11 and requires the RQDX3 controller.

RX33 and RX50 diskettes are supported on the RX33 drive as both data and system media.

For RT-11 V5.4, the FORMAT utility does not support the RX33 diskette drive. You must format RX33 diskettes before you initialize them, and you must use the formatting diskette that is shipped with your drive to format RX33 diskettes. See Section 4.3.2 for information on that procedure.

RX33 and RX50 diskettes are physically similar. Use the color of the metal diskette oxide surface to distinguish them from each other. RX33 diskettes have a black oxide surface, while the surface of the RX50 diskette is brown. The difference is apparent when you compare them.

CAUTION

DO NOT FORMAT RX50 DISKETTES IN THE RX33 DRIVE

It is physically possible to format RX50 diskettes as RX33 diskettes. However, RX50 diskettes do not contain the same oxide (storage medium) as RX33 diskettes and therefore cannot safely store 1.2M bytes of data. RX50 diskettes therefore can randomly lose data if they are formatted as RX33 diskettes.

1.2.5 RX50 Diskette (DU or DZ)

The RX50 is a diskette subsystem available on Q-BUS processors such as the MicroPDP-11, UNIBUS processors, and Professional 300 series processors. The subsystem consists of two drives, each of which holds one 5 1/4-inch diskette. Each diskette provides a storage capacity of 400K bytes (800 blocks).

RX50 diskettes on the Q-BUS processors are MSCP (mass storage communication protocol) devices supported by the MSCP disk class handler, DU. On Professional 300 series processors, RX50 diskette drives are supported by the DZ handler. MicroPDP-11 and Professional computers can read and write each other's diskettes. However, since they are supported by different device handlers, a diskette configured to bootstrap on one cannot be hardware bootstrapped on the other.

RX50 diskettes can be used as system or data storage volumes. The RT-11 V5 distribution kit is available on RX50 diskettes.

1.2.6 TK25 Magtape (MS)

The TK25 is a streaming TS11-compatible cartridge magtape drive, operating on the Q-BUS. Data is recorded on a 600-foot, 10-track (1/4-inch format) magtape. The TK25 stores up to 60M bytes of data in 8K-byte blocks, with a record size of up to 16K bytes. The read/write speed is 55 inches per second while streaming, for a data transfer rate of 55K bytes per second.

1.2.7 TK50 Magtape (MU)

RT-11 V5.4 supports TK50 magtape.

The TK50 is a cartridge tape drive that uses the tape-class mass storage communication protocol (TMSCP). The TK50 stores approximately 95M bytes of data (unformatted) on 600 feet of 1/2-inch magtape at a density of 6667 bits/inch. The tape contains 22 tracks.

The RT-11 V5.4 distribution kit is available on TK50 magtape. You can install RT-11 from the TK50 magtape using the Automatic Installation Procedure. See Section 1.5.1 for information.

You can boot RT-11 only from TK50 magtape controller unit 0. You must install TK50 magtape controller unit 0 (MU0) at CSR address 774500 and at VECTOR 260.

1.2.8 TS05 (TSU05/TSV05) Magtapes (MS)

The TS05 is a TS11-compatible tape drive. The TSU05 runs on UNIBUS processors, the TSV05 on Q-BUS processors.

Data is recorded on a 9-track 1/2-inch format magtape. The TS05 can store up to 28M bytes (in formatted 2K records) on a standard 10 1/2-inch, 2400-foot reel of tape. TS05 tape drives can accept three tape reel sizes: 7 inch, 8-1/2 inch, and 10-1/2 inch.

The TS05 is a streaming tape drive that operates in two modes: TS11 compatible mode and extended features mode. In TS11 mode, the TS05 is a TS11 look-alike that automatically streams at 25 in/s. In extended features mode, which is turned on by setting the hardware extended features switch, the tape can stream at 100 in/s under program control.

1.3 OTHER NEW HARDWARE SUPPORT

RT-11 supports the following new hardware.

1.3.1 EXPANDER BOX

The EXPANDER BOX lets you add one additional RD5x controller and hard disk to a Professional series processor configuration. RT-11 V5.4 supports the EXPANDER BOX in the following manner.

The DW handler supports only a single device connected to a single controller. Therefore, to support the second controller and hard disk, you must COPY the DW handler to another handler name (for example, DV) and use a new SET command to assign the created handler (DV) to the second device.

COPY the DW handler to DV and use the new SET DV SLOT=n command to support the new EXPANDER BOX. (Using device name DV is optional; you can use any device name not recognized by or assigned to RT-11.) Use the SLOT=n option only to support an EXPANDER BOX and only with the created handler.

Use the SLOT=n option to identify the EXPANDER BOX RD5x controller card slot in the Professional's backplane. The six card slots are numbered 0 through 5, from front to back. The first RD5x controller is typically in slot 0. The second RD5x controller typically goes in slot 5. Assuming that arrangement, issue the following commands to support the EXPANDER BOX:

```
.COPY SY:DW[X].SYS DV[X].SYS
.SET DV SLOT=5
.INSTALL DV
```

The SLOT=n option and created handler as support for the EXPANDER BOX are an interim solution. A simpler solution will probably be supplied in a future release. Do not rely on the SLOT=n option being present in future releases of RT-11.

1.3.2 512K-Byte Memory Expansion Card

The 512K-byte memory expansion card is supported for Professional series processors. It adds 512K bytes to the on-board memory and can give the processor up to 1024K bytes of memory.

1.3.3 Mini-Exchange

The Mini-Exchange is a serial switching box that allows up to eight ports to be controlled by character string commands. RT-11 does not currently provide support for printers on the Mini-Exchange.

See the RT-11 System Utilities Manual for information on using the Mini-Exchange with the virtual terminal communication utility (VTCOM).

1.4 NEW DISTRIBUTION KITS

The format of RT-11 distribution kits has changed because of the new automatic installation and verification procedure (see Section 1.5.1). The procedures for RT-11 V5 distribution kits that will not be installed automatically are in Chapter 4.

1.5 NEW SOFTWARE COMPONENTS

Although RT-11 V5's primary goal is to extend the functionality of existing software, V5 also includes the following new operating features:

- Automatic installation (AI) and installation verification procedure (IVP)
- Backup utility program (BUP)
- Concise command language (CCL)
- Global region support in extended memory
- Graphics utility support (PRO/GIDIS)
- Indirect control file processor (IND)
- Virtual KED (KEX)
- Logical disk subsetting handler (LD)
- Ethernet controller hardware handlers (NC/NQ/NU)
- Hardware setup utility (SETUP)
- Single-line editor (SL)
- Transparent spooling utility (SPOOL)
- Native Transfer Utility (TRANSFER) (unsupported)
- User commands first utility support (UCF)

- User command linkage utility (UCL)
- Virtual RUN utility (VBGEXE) (unsupported)
- Virtual terminal communication package (VTCOM)

In addition, RT-11 V5 includes new programmed requests, macros, SYSLIB routines, handlers, error messages, new system generation procedures and options, a new software update process, and new unsupported utilities. The following sections summarize those new components and refer you to more detailed descriptions in the documentation set.

1.5.1 New Automatic Installation and Verification Procedure

The RT-11 V5 automatic installation and verification procedure installs RT-11 by conducting an interactive dialog at the console terminal. As you answer the dialog questions, the system creates and exercises a working RT-11 system.

The following hardware configuration is required for the automatic installation procedure:

- PDP-11 processor with 24K words of memory or Professional 300 series computer (TK50 magtape automatic installation requires 256K bytes of memory)
- A line or Professional clock
- VT100 series, VT200 series, or LA100 series console terminal or a Professional 300 series system
- One of the following mass storage configurations:

  MicroPDP-11 (RX50/RD31/RD51/RD52/RD53) - distribution kit resides on RX50 diskettes. Automatic installation using RX50 diskettes requires using the RX50 diskette drive. Automatic installation using RX50 diskettes is not supported on the RX33 diskette drive. See Section 4.3.1 for information on installing RT-11 using the RX33 diskette drive

  Professional 325, 350, or 380 - distribution kit resides on RX50 diskettes

  TK50 magtape - distribution kit resides on TK50 magtape

  Dual RL02 disks - distribution kit resides on RL02 disk

  Dual RX02 diskettes - distribution kit resides on RX02 diskettes

If you bought an RT-11 distribution kit on any of those media, you should have received an RT-11 Automatic Installation Booklet, which tells you how to start the automatic installation process. If your configuration does not meet the requirements shown above, you must have DIGITAL install your system or install your system by following the procedures described in the RT-11 Installation Guide.

1.5.1.1 **RX02 and RX50 Automatic Installation Information** - RX02 and RX50 distribution kits include two copies of the RT-11 distributed software, so you do not have to back up the distribution kit. Although not supported for the automatic installation procedure, RX01 distribution kits also include two copies of the distributed software.

Store one copy as your master distribution kit and bootstrap the other copy to run the automatic installation procedure. Be careful to keep the volumes from those two (RX02 or RX50) distribution kits separate. Once the installation procedure has been run, volumes AUTO and 1 of the distribution kit you installed will no longer be identical to volumes AUTO and 1 of the master distribution kit you stored.

Automatic installation from RX50 diskettes requires the RX50 diskette drive. Automatic installation from RX50 diskettes is not supported on the RX33 diskette drive.

1.5.1.2 **RL02 Automatic Installation Information** - For RL02 distribution, the only difference between the original distribution volume you received and the installed volume is the bootstrap. The bootstrap on the original distribution kit bootstraps the automatic installation monitor, RT11AI.SYS. The bootstrap on your installed system bootstraps the RT-11 FB monitor, RT11FB.SYS. Therefore, you can return your installed system to its original state by copying the RT11AI.SYS bootstrap to the boot blocks of your disk.

1.5.1.3 **TK50 Automatic Installation Information** - For the TK50 magtape distribution, you must have at least 256K bytes of memory. Automatic installation of the TK50 magtape distribution requires a J-11 microprocessor-based Q-BUS MicroPDP-11 or UNIBUS PDP-11/84 processor with an installed MU boot ROM. You can boot RT-11 only from TK50 magtape controller unit 0. You must install TK50 magtape controller unit 0 (MU0) at CSR address 774500 and at VECTOR 260.

Boot the TK50 magtape and type MDUP.AI in response to the MSBOOT prompt (*). The MDUP.AI file copies a partial bootable working system into the extended memory virtual disk (VM), and VM is booted. The normal automatic installation procedure is then initiated, and you are prompted as to what further actions to take.

See Chapter 4 for procedures for RT-11 V5 distribution kits that do not support automatic installation.

1.5.2 New System Procedures and Utilities

This section describes the new system procedures and utilities provided with RT-11 V5.

1.5.2.1 **Backup Utility Program (BUP)** - The backup utility program provides a quick way to store a large volume or file on a set of smaller volumes.

BUP lets you copy a large volume or file to several specially initialized backup volumes. The file or volume cannot be used, however, while stored on the backup volumes. BUP also lets you initialize backup volumes, obtain directory information about a set of

backup volumes, and restore a volume or file to its original form from a set of backup volumes. In addition, BUP will use the 100 in/sec streaming mode of the TS05 magtape drive under the SJ and FB monitors.

See Chapter 3 in the RT-11 System Utilities Manual for a complete description of BUP.

Changes have been made to the BUP utility. See Chapter 2 for details.

1.5.2.2 **Concise Command Language (CCL)** - Concise command language lets you issue commands directly to utility programs or your own user-written programs on a single command line.

Prior to RT-11 V5, you could issue commands on a single command line only by using DCL keyboard monitor commands or by running (R or RUN) the program and specifying a CSI command line. CCL lets you run the program and specify the input and output files and all accompanying options on a single line.

See the RT-11 System User's Guide for more information on CCL.

1.5.2.3 **DBG-11 Symbolic Debugging Package (Unsupported)** - DBG-11 is a symbolic debugging package that lets you interactively debug a MACRO-11 program. You can use DBG-11 to:

- Control program execution through breakpoint traps and single-step execution of instructions.
- Display the contents of locations in numeric or symbolic form.
- Change the contents of locations.
- Define symbolic names for memory addresses and location values.

The unsupported DBG-11 package consists of four pseudodevice handlers (SDH.SYS, SDHX.SYS, SDS.SYS, and SDSX.SYS) and DBGSYM.SAV, a symbol definition utility. The new DBG-11 Symbolic Debugger User's Guide tells you how to use DBG-11 under RT-11. Also see the DBG-11 restrictions section in Chapter 3.

1.5.2.4 **Ethernet Controller Hardware Support** - The Ethernet handlers provide hardware support for Ethernet class controllers.

- The NC handler supports the DECNA Ethernet controller for the Professional 300 series processors.
- The NQ handler supports the DEQNA Ethernet controller for Q-BUS processors.
- The NU handler supports the DEUNA and DELUA Ethernet controllers for UNIBUS processors.

Ethernet handlers run only under the XM monitor. Each handler supports only one controller. An unsupported program, NITEST.MAC, is included with your distribution kit; use it to verify connection to the Ethernet.

For information on the Ethernet handlers NC and NQ, see the RT-11 Software Support Manual. All information describing NC and NQ also applies to the NU handler. For information on building and running the unsupported program NITEST.MAC, see the RT-11 System Utilities Manual.

1.5.2.5 **Global Region Support in Extended Memory** - Global regions are areas of extended memory that are controlled by the operating system, rather than by a particular program. Global regions can be used by more than one program and can remain in extended memory after the creating program has exited. User programs, system utilities, system device handlers, and monitors can create global regions. You can create up to ten global regions, and a program can attach to a combination of up to six local or global regions.

Creating, attaching to, detaching from, and eliminating global regions are controlled by programmed requests in the same manner as regions local to a program. For complete information on global region support, see the RT-11 Software Support Manual.

1.5.2.6 **Graphics Utility Support (PRO/GIDIS)** - The graphics utility PRO/GIDIS is a foreground job for Professional 300 series processors. PRO/GIDIS runs under the XM monitor, and requires FPU (floating-point unit) hardware and the extended bit-map option (EBO). PRO/GIDIS lets you create applications in which images are described using geometrical entities such as lines, arcs, and shaded areas.

PRO/GIDIS support is provided by the Professional interface handler (PIX.SYS) and the foreground program GIDIS.SAV. RT-11 provides three interfaces between the applications program and the Professional interface handler:

- The .SPFUN programmed request
- The GIDCAL system subroutines
- The ISPFN/ISPFNC/ISPFNF/ISPFNW system subroutines

See the RT-11 System Programmer's Manual for complete details on running PRO/GIDIS under RT-11.

1.5.2.7 **Indirect Control File Processor (IND)** - The indirect control file processor executes indirect control files. IND control files contain IND directives, which control the execution of the indirect control file, and may contain keyboard commands (DCL, CCL, and UCL). You can use indirect control files to access other files, execute keyboard monitor commands, define symbols, pass parameters, and perform logical tests.

When running under the XM monitor, IND stores context information in a region of high memory, resulting in a performance improvement.

KMON is the default command file processor in the distributed monitors. If you want to change the default command file processor to IND, use the monitor command SET KMON IND or apply the customization given in the RT-11 Installation Guide.

Chapter 5 in the RT-11 System User's Guide describes how to create and execute indirect control files.

1.5.2.8 **Virtual KED (KEX)** - The virtual KED program (KEX) is the default editor under the XM monitor and is available for use under the XM monitor only. KEX can be run as a foreground or system job, allowing editing to continue while the system is performing other functions.

KEX editing commands are identical to KED commands. However, KEX maximizes the amount of high memory used, while minimizing the amount of low memory used. Therefore, KEX will continue to operate in many instances where there is insufficient low memory for KED to run. You also maximize the size of paste buffers when you use KEX; you have more memory available for them.

You can create multiple copies of KEX by running KEX as a system job and assigning KEX1, KEX2, KEXn as logical job names to KEX. For example, the following series of commands creates two KEX editors, KEX and KEX1, and runs them as system jobs:

```
.SRUN KEX.SAV/NAME:KEX <RET>

.SRUN KEX.SAV/NAME:KEX1 <RET>
```

That procedure can be useful, for example, in examining the contents of one file using KEX while you are editing another file using KEX1. You can also edit both files concurrently.

Assuming you have created and are running two KEX editors, use the following procedure to examine and/or edit two files concurrently:

- Open the first file (file1.ext), using KEX. Type CTRL/X and specify KEX in response to the system job prompt. Specify file1.ext in response to the KEX prompt.

  ```
  JOB>KEX <RET>
  *file1.ext <RET>
  ```

  The contents of file1.ext appear on your terminal screen.

- Open the second file (file2.ext) using KEX1. Type CTRL/X and specify KEX1 in response to the system job prompt. Specify file2.ext in response to the KEX1 prompt.

  ```
  JOB>KEX1 <RET>
  *file2.ext <RET>
  ```

  The contents of file2.ext appear on your terminal screen.

- You can now return to file1.ext, using the following three steps:

  1. Type CTRL/X
  2. Type KEX in response to the system job prompt and press RETURN
  3. Type CTRL/W to repaint your terminal screen with the contents of file1.ext.

- You can go from one file to the other using that procedure. Simply type CTRL/X, specify which editor you want to use, and repaint your screen using CTRL/R. The cursor position in one file does not change because of editing you are doing in the other file. The editors are completely independent.

- You use the same KED commands and operations when you use multiple KEX editors.

- Because you are running one or more KEX editors as system jobs, you still have access to the monitor prompt using CTRL/B.

- When you want to remove an editor from memory, type CTRL/C in response to that editor's prompt (*).

The number of KEX editors you can run as system jobs is limited only by available memory or the 8-job limit of RT-11.

On systems that include multiterminal support, multiple copies of KEX can be run (using the SRUN and FRUN commands), each from its own terminal.

1.5.2.9 **Logical Disk Subsetting Handler (LD)** - The logical disk subsetting handler lets you define logical disks, which are subsets of physical disks. You define logical disks by assigning a logical disk unit number to a file on a physical disk. You can then use the logical disk as though it were a physical disk.

Logical disk subsetting is particularly useful when you work with large disks, which often run out of directory entry space before the volume is full. Since each logical disk contains its own directory, dividing a physical disk into several logical disks increases directory entry space. Logical disk subsetting also increases the speed of directory operations on large disks.

Chapter 9 in the RT-11 System Utilities Manual details logical disk subsetting.

A method for creating logical disks is shown in Section 3.4.1.

1.5.2.10 **Hardware Setup Utility (SETUP)** - The SETUP utility lets you choose operating characteristics for video terminals, printers, and some system clocks. By typing SETUP commands, you can control terminal characteristics such as background and text color, scrolling, and cursor form (block or underscore), printer characteristics such as length of printed pages, and clock characteristics such as the 12- or 24-hour time reporting format.

SETUP's video characteristics are intended primarily for use with Professional 300 series computers, since they have no hardware terminal setup facility. However, many SETUP commands are also valid for VT100 and VT200 series terminals. Some commands are valid for only VT100 series terminals.

See the RT-11 System User's Guide for details on how to use SETUP and for a list of SETUP commands. See Section 4.6.1 for a table of SETUP modes and controls for specific line printers.

1.5.2.11 **Single-Line Editor (SL)** - The single-line editor (SL) lets you edit the current keyboard command line typed on a video terminal before you terminate the line. The single-line editor lets you position the cursor anywhere in the current line for editing by using a subset of KED (keypad editor) commands. You can also recall previous and one-before-previous input lines for editing.

See the RT-11 System User's Guide for instructions on using the single-line editor.

1.5.2.12 **Transparent Spooling Package (SPOOL)** - The transparent spooler (SPOOL) is a utility you can use for sending output to the line printer. SPOOL runs as a foreground or system job. Once SPOOL is running, its operations are transparent. Anytime you send output to the line printer, either explicitly by issuing commands (such as COPY and PRINT) or by using commands and options that send output to the line printer by default (such as COMPILE/LIST), SPOOL accepts the output and sends it to the printer. While SPOOL runs in the foreground, you can continue to work on other jobs in the background. SPOOL differs from the Queue Package in that you need not send output to SPOOL as a complete file. Instead, SPOOL accepts output as it becomes available ("pipeline" operation).

Although the line printer is SPOOL's default output device, you can apply a software customization to change the default.

See the RT-11 System Utilities Manual for an explanation of how to use SPOOL.

1.5.2.13 **Native Transfer Utility (TRANSFER)** - TRANSFER lets you copy files between your RT-11 stand-alone system and a PDP-11 processor running RSX or a VAX processor running VMS. No intermediary such as RTEM is required. TRANSFER is an unsupported utility. Complete information on using TRANSFER is located in Chapter 5.

1.5.2.14 **User Commands First Feature (UCF)** - UCF lets you write your own keyboard monitor (KMON) preprocessing utility that intercepts any command line input before KMON attempts DCL, CCL, or UCL parsing. UCF lets you process valid DCL commands in a way you choose, rather than the way KMON would normally process the commands. UCF can intercept and process commands that KMON would treat as invalid.

The distributed file UCL.SAV (user command linkage utility) can be copied and renamed to create a functioning UCF utility.

Command line input can originate from direct console input, a command file, a command line in the chain area from an exiting program, a BATCH file, or IND. UCF can:

- Receive a command line and return it unaltered to KMON for processing.
- Receive a command line with syntax not recognized as valid by KMON and convert the command line to KMON-valid command syntax.
- Intercept and destroy a command you do not want executed.

- Send escape sequences in response to immediate commands, such as CLEAR SCREEN.

See the RT-11 System User's Guide for information on using UCF.SAV.

1.5.2.15 **User Command Linkage Feature (UCL)** - User command linkage (UCL) lets you write your own command parser, so you can create your own commands.

See the RT-11 System User's Guide for instructions on how to define commands using the distributed UCL.SAV. Refer to the RT-11 Software Support Manual for more information on UCL.

1.5.2.16 **Virtual RUN Utility (VBGEXE)** - VBGEXE lets you run some programs under the XM monitor when there is not enough low memory to run the program by using the R or RUN command. VBGEXE is especially useful for running programs on a Professional 300 series computer under the XM monitor.

Some restrictions apply when using VBGEXE, and it is an unsupported utility. See the RT-11 System Utilities Manual for information on using VBGEXE.SAV.

1.5.2.17 **Virtual Terminal Communication Package (VTCOM)** - VTCOM lets you use your PDP-11, PDT-11/150, or Professional computer as a terminal when you connect it to a host computer. VTCOM.REL runs under all monitors. VTCOM.SAV, which runs under the XM monitor, minimizes the amount of low memory used, so more low memory is available for other programs.

When running VTCOM, you can access facilities on an RT-11, RSX-11, RTEM-11, or VAX/VMS host system, and perform ASCII file transfers between the host computer and your stand-alone system. In addition, if the host computer is running RSX or VAX/VMS and TRANSF.TSK or TRANSF.EXE is installed on the host, you can transfer binary files as well. Since VTCOM can run as a foreground or system job, you can continue working under RT-11 while you maintain connection to the host computer.

The RT-11 System Utilities Manual lists requirements for running VTCOM and explains how to use VTCOM. Chapter 5 shows how to run TRANSF.TSK and TRANSF.EXE.

1.5.3 New Programmed Requests

This section describes the new programmed requests available with RT-11 V5. The programmed requests are located in SYSMAC.SML.

1.5.3.1 **.ABTIO** - The .ABTIO programmed request lets a running job stop all outstanding I/O operations on a specified channel without terminating the program under the FB and XM monitors. Under SJ, the request is simulated with a .WAIT directive.

See the RT-11 Programmer's Reference Manual for details and examples of the .ABTIO programmed request.

1.5.3.2 **.ADDR** - The .ADDR macro computes the address of the location you specify in a position-independent manner (independent of its link-time virtual address) and stores that address in a register or on the stack.

See the RT-11 Programmer's Reference Manual for more information on the .ADDR macro.

1.5.3.3 **.ASSUME** - The .ASSUME macro tests, at assembly time, the validity of a condition you specify. If the test is false, MACRO generates an assembly error and prints a message from a comment you supply as an argument to the macro.

See the RT-11 Programmer's Reference Manual for more information on the .ASSUME macro.

1.5.3.4 **.BR** - The .BR macro warns you if code that belongs together is separated during assembly. When you call the .BR macro, you specify a location as an argument. .BR checks that the address of the next instruction matches the address of the location you specified. If the addresses do not match, .BR causes MACRO to print an error message.

See the RT-11 Programmer's Reference Manual for more information on the .BR macro.

1.5.3.5 **.CKXX** - The .CKXX macro generates CK.Rn register checking macros. You specify a register as an argument to .CKXX, and .CKXX creates the CK.Rn checking macro for that register. More than one register can be specified for .CKXX, creating a CK.Rn checking macro for each register. More than one CK.Rn checking macro can be created for a particular register.

Use the CK.Rn macro to simplify the checking of assumptions about registers when those registers are used in autoincrement and autodecrement mode instructions. You can also assign symbols to CK.Rn that can be used to store register contents during autoincrement and autodecrement operations. Those symbols can later be used to verify the position of the stored values.

Macro Call: .CKXX <reg[,reg,...]>

where:

reg — is the register or registers you want .CKXX to define as check registers. Calling .CKXX generates a CK.Rn check macro for each register you include in the reg argument. For example, the macro call:

```
.CKXX <R0>
```

generates the check macro CK.R0. The macro call:

```
.CKXX <R0,R1>
```

generates the check macros CK.R0 and CK.R1. If you want to generate more than one check macro for a single register, for example R1, append a different letter to the register number for each check macro you want to create. For example:

```
.CKXX <R1A,R1B>
```

generates the check macros CK.R1A and CK.R1B.

The check macro CK.Rn, generated by .CKXX, has the following form:

```
CK.Rn[alph]  [label][,change][,result]
```

where:

n — is the register you assigned to this check macro using .CKXX

alph — is an alphabetic character you assigned to this check macro using .CKXX

label — is the value or label you assume equates to the check register. If the value or label does not equate to the check register, a P (phase) error is returned at assembly time. See the PDP-11 MACRO-11 Language Reference Manual, Appendix D, for a description of the P assembly error

change — indicates the check macro increment or decrement. The change value must be preceded by a plus sign (+) to indicate increment or a minus sign (-) to indicate decrement

result — is a new location assigned to the check macro. Use the result argument to assign a symbol to the check macro. When you want to verify (further in your program) that the check macro equates to that symbol, specify that symbol in the label argument

The check macro for a particular register must exactly track the operations done on that register. The check macro for a register must be explicitly incremented or decremented if that register is incremented or decremented. Refer to the following program fragment example.

Example:

This example illustrates the following check macro features:

- Assigning an initial value to the check macro
- Transferring a value from one check macro to another
- Checking the current value of a register pointer and tracking for autoincrement
- Checking the current value of a register pointer and tracking for autodecrement

- Tracking for autoincrement and autodecrement without checking for values

For the purpose of this example, assume the following data block:

```
DBLK:   .BLKW   4
        DEV=:   0
        NAME=:  2
        TYPE=:  6

.MCall  .CKXX                   ; call .CKXX
.CKXX   <R3,R4>                 ; create CK.R3 and CR.R4
        .
        .
        .
        .
MOV     #DBLK,R3                ; point to data block DBLK
        CK.R3=DEV               ; assign initial value to check macro
MOV     R3,R4                   ; copy the pointer
        CK.R4=CK.R3             ; copy the check macro to new one
        CK.R3 DEV,+2            ; check R3 equates to DEV
                                :  and increment
MOV     (R3)+,R0                ; load device name into R0
        CK.R3 NAME,+2           ; check R3 equates to NAME
                                ;  and increment
MOV     (R3)+,NAME+0            ; get first part of file name
        CK.R3 ,+2               ; increment but no check (no label)
MOV     (R3)+,NAME+2            ; get last part of file name
        CK.R3 TYPE              ; check R3 equates to TYPE
                                ;  but no increment
MOV     @R3,R2                  ; filespec extension into R2
        CK.R3 NAME+2,-2         ; decrement and check R3 equates
                                ;  to NAME+2
TST     -(R3)                   ; test last 3 chars of filespec
        CK.R3 DEV,-2-2          ;  are they blank (0 in RAD50)?
CMP     -(R3),-(R3)             ; skip back to device
```

1.5.3.6 **.DREST (Device Handlers Only)** - The .DREST macro places device specific information in block 0 of a device handler. The device specific information includes:

- The device classification
- The variants of a device classification and additional information about some device classifications
- Whether the device handler contains updatable internal data table(s) accessible by SPFUN 372
- The type (device classification) of the updatable internal data table
- Whether the device handler has a table in block 0 that contains bad-block replacement information

That information is used by RT-11 utilities to determine the characteristics of that device handler.

Macro Call: .DREST [class=n][,mod=n][,data=dptr][,type=n][,replace=rptr]

where:

class is the device classification. Specify the device classification symbol (DVC.xx) for n in the class argument. An octal device classification value is stored in byte 20 of block 0 in the device handler. Valid device classification symbols and stored values for the class argument are:

Symbol	Value	Meaning
DVC.CT	6	Cassette (CT-TA60) tape (CT)
DVC.DE	10	DECnet executive pseudohandler
DVC.DK	4	RT-11 file structured disk (DD, DL, DM, DP, DT, DU, DW, DX, DY, DZ, RK)
DVC.DL	12	DECnet port (line) handler
DVC.DP	11	DECnet protocol pseudohandler
DVC.LP	7	Printer (LP, LS, SP)
DVC.MT	5	File structured magtape (MM, MS, MT, MU)
DVC.NI	13	Ethernet port handler (NC, NQ, NU)
DVC.NL	1	NULL handler (NL)
DVC.PS	14	Pseudohandler (PI, SL)
DVC.TP	3	Reserved
DVC.TT	2	Terminal class handler (TT)
DVC.UK	0	Unknown device class
DVC.VT	15	Virtual terminal port handler (XL, XC)

mod indicates a variation or additional information about a device classification. Specify the device modification symbol (DVM.xx) for n in the mod argument. An octal device modification value is stored in byte 21 of block 0 in the device handler. Valid device modification symbols and stored values for the mod argument are:

Symbol	Value	Meaning
0	0	No variant or information (default)
DVM.DM	2	With class=DVC.DK, indicates device has an extra error word prefixed to SPFUN buffers
DVM.DX	1	With class=DVC.DK, indicates device is an RX01-compatible drive

DVM.NF	200	With all class devices, indicates handler can only be loaded, and cannot be fetched. This bit is read-only and cannot be set using the .DREST macro. (This bit is set by the .DRPTR macro with the fetch=*NO* argument.)

data specifies whether the handler has internal updatable data table(s), accessible by SPFUN 372. The data argument information is stored in word 70 of block 0 in the handler. You must include the type argument if you specify the data argument. For the data argument, dptr can be:

0	The default; specifies that the handler does not have an internal data table
dptr	is the starting address of the internal data table(s)

type specifies whether a device type classification exists for the internal data table(s). The device type classification is made up of one to three RAD50 characters and is normally the same as the RT-11 device name. The type argument information is stored in word 72 of block 0 in the handler. You must include the type argument rad if you specify the data argument dptr. For the type argument, n can be:

omitted	The default; the handler does not have a device type classified internal table
rad	The handler has a device type classified internal table, and rad is the RAD50 device type classification

replace specifies whether the handler has a table in block 0 that contains bad-block replacement information. The replace argument information is stored in word 22 of block 0 in the handler. For the replace argument, rptr can be:

0	The default; the handler does not contain a bad-block replacement information table
rptr	is the starting address of a bad-block replacement information table

Although all .DREST arguments are optional, some arguments are paired. For example, the mod argument has no meaning without the class argument. Also, the data argument requires the type argument.

Errors:

None

Example:

```
        .Title  SK -- Handler Skeleton

;+
; .DRPTR/.DREST/.DRSPF - This is an example skeleton handler
; that illustrates using the .DRPTR, .DREST, and .DRSPF requests.
;-

        .MCALL  .DRDEF          ;Get handler definitions
        .MCALL  .ASSUME         ;Checking macro
        .MCALL  .EXIT           ;To finish run

        .MACRO  ...             ;Define ellipsis (allow
                                ; ellipsis to assemble)
        .ENDM

                ;Generate non-executable handler information tables
                ; containing the following information:
                ; Handler is SK
                ; Handler ID is 350 (user written handler)
                ; Handler accepts neither .READ nor .WRITE
                ; Handler accepts .SPFUN requests
                ; Device is 1 block in size
                ; Device has a CSR at 176543
                ; Device has a vector at 20

        .DRDEF  SK,350,RONLY$!WONLY$!SPFUN$,1,176543,20

                ;Handler has .Fetch and $LOAD code to be executed:

        .DRPTR  FETCH=Fetch,LOAD=Load

                ;Handler is for a "Null" class device
                ;Handler has a data table called DATABL
                ;Data table is of the SKL format

        .DREST  CLASS=DVC.NL,DATA=DATABL,TYPE=SKL

                ;Handler accepts the following SPFUN codes:
                ; 372,376,377

        .DRSPF  <372,376,377>

                ;Handler CSR is not to be checked at install,
                ;but is to be displayed:

        .DRINS  -SK
        ...                     ;Here is any installation check code
        RETURN

        .ASSUME . LE 400,MESSAGE=<;Installation area overflow>

                ;Handler accepts SET SK [NO]BONES command:

        .DRSET  BONES,123456,CORPUS,NO

CORPUS:                         ; SET SK BONES
        COM     R3              ;Flip bits
        NOP                     ;Pad code
        .ASSUME . EQ CORPUS+4,MESSAGE=<;No option code in wrong place>
NOCORP:                         ; SET SK NOBONES
        MOV     R3,PICKNT       ;Set value in block 1
        RETURN
```

```
        .ASSUME . LE 1000,MESSAGE=<;Set area overflow>

        .DRBEG  SK              ;Handler Queue Manager Entry point
        BR      START           ;Skip data table
DATABL:
        .RAD50  "SKL"           ;Table ID
WRIST:  .BlkW   1               ;Table contents
ANKLE:  .BlkW   1               ;...

                ;Set up the Vector table:

SK$VTB: .DRVTB  SK,SK$VEC,SKINT,0
        .DRVTB  ,SK$VEC+4,SKINT,1

PICKNT: .BLKW   1               ;Value controlled by Set command
        .ASSUME .-2 LE SKStrt+1000,MESSAGE=<;Set object
                                            not in block 1>

START:                          ;Executable Queue code
        ...
        RETURN

        .DRAST  SK,4,ABORT      ;Interrupt entry point
        BCS     INT2            ;Interrupt from second vector
        ...
        RETURN
INT2:                           ;Second interrupt vector code
        ...
        RETURN

ABORT:                          ;Abort entry point
        ...
        .DRFIN  SK              ;Completion return

                                ;End of memory resident part of handler

        .DRBOT  SK,Entry        ;Boot code

ENTRY:
        ...                     ;Hard boot code to call read routine
        RETURN

READ:
        ...                     ;Read routine
        RETURN

        .DREND  SK              ;End of boot code

        .PSECT  SETOVR          ;Suggested block aligned PSect

FETCH:
        ...                     ;Code executed on FETCH
        RETURN
LOAD:
        ...                     ;Code executed on LOAD
        RETURN

RUN:
        ...                     ;Code executed on RUN
        .EXIT

        .END    RUN
```

1.5.3.7 **.DRINS (Device Handler Only)** - The .DRINS macro sets up the installation code area in block 0 of a device handler. .DRINS defines the addresses that contain the display CSRs used by RESORC and the CSRs checked by the INSTALL keyboard command code and defines the system and data device installation entry points.

See the RT-11 Programmer's Reference Manual for more information on the .DRINS macro.

1.5.3.8 **.DRPTR (Device Handler Only)** - The .DRPTR macro places pointers in block 0 of a device handler that refer to handler service routines located in that handler. Those handler service routines are used by utilities, monitors, and the handler itself. The routines help govern how the handler behaves during the following:

- Bootstrap (BSTRAP) operations (load argument)
- .FETCH and .RELEASE requests
- LOAD and UNLOAD commands
- Job abort (release argument)

Macro Call: .DRPTR [fetch=n][,release=n][,load=n][,unload=n]

where:

fetch specifies whether a handler service routine is called by the .FETCH programmed request. For the fetch argument, n can be:

- 0 The default; the handler does not have a service routine for the .FETCH programmed request. The handler can still be fetched
- n is the starting address of the service routine to be called by .FETCH
- *NO* A literal string; the handler cannot be fetched. The handler must be loaded

release specifies whether a handler service routine is called by the .RELEASE programmed request. For the release argument, n can be:

- 0 The default; the handler does not have a service routine for the .RELEASE programmed request
- n is the starting address of the service routine to be called by .RELEASE

load specifies whether a handler service routine is called when the handler is loaded by bootstrap routine or LOAD command. For the load argument, n can be:

- 0 The default; the handler does not have a service routine to be called when it is loaded
- n is the starting address of the service routine to be called when the handler is loaded

unload specifies whether a handler service routine is called when the handler is unloaded by the UNLOAD command. For the unload argument, n can be:

- 0 The default; the handler does not have a service routine to be called when it is unloaded
- n is the starting address of the service routine to be called when the handler is unloaded

.DRPTR arguments are often paired and argument values are often matched because routines they point to are used together or rely on each other. The fetch and load argument values are often paired. Similarly, the release and unload argument values are often the same.

Errors:

None

Example:

See the .DREST example in this manual.

1.5.3.9 **.DRSPF (Device Handler Only)** - The .DRSPF macro defines the special function codes supported by a handler. .DRSPF builds a table or tables containing the supported special function codes. RT-11 utilities or user programs use the table(s) to determine which special function codes are supported by that handler.

Macro Call: .DRSPF arg[,arg2]

where:

arg can be specified in two ways: the list method and the extension table method. The list method is the simplest but contains a restriction. The extension table method is useful if the restriction imposed by the list method is a problem. Both ways of using arg can be included in the same handler.

- List Method - arg can be a list of one or more special function codes. That list is located in block 0 of the handler at locations 22 through 27.

You must follow certain rules in specifying the list because of the way the special function codes are stored in the handler.

Special function codes consist of three octal digits. You are allowed a total of only three unique ordered combinations of the first two digits of a special function code in all lists or combination of lists, when arg is used in this manner. You can use any octal digit as the third digit of any entry in these lists. That restriction is not a problem for most handlers. You can define all supported special functions for most handlers in one list or series of lists.

Each list must be enclosed in angle brackets (<>). Special function codes are separated by commas (,). The special function codes can be specified in any order.

Do not specify the arg2 argument when using the .DRSPF macro in this manner.

An example list for the MU handler:

```
.DRSPF <360,370,371,372,373,374,375,376,377>
```

The same special functions for the MU handler could also be included in a series of lists:

```
.DRSPF <360>
.DRSPF <370,371,372>
.DRSPF <373,374,375>
.DRSPF <376,377>
```

In both examples, only two unique ordered combinations of the first two digits of the special function codes were used: 36 and 37.

- Extension Table Method - arg can be a pointer to an extension table address. Do not place the extension table in block 0 of the handler.

  The pointer to the extension table address must be preceded by a plus sign (+). The extension table address must have the high bit cleared.

  The extension table contains one or more .DRSPF macros. The arg argument for each .DRSPF macro is a minus sign (-), and the arg2 argument is a list of special function codes. Each of the special function codes in arg2 must have the same first two octal digits. The list must be enclosed by angle brackets (<>). Special function codes are separated by commas (,). The special function codes can be specified in any order. The extension table is terminated by a word containing zero (0).

In the following example, the pointer to the extension table is the symbol EXTTAB:

```
        .DRSPF +EXTTAB
               .
               .
               .
        EXTTAB:
               .DRSPF  -,<340,341>
               .DRSPF  -,<350,351,353>
               .DRSPF  -,<200,202,203,204,205>
               .DRSPF  -,<210,212>
               .WORD 0
```

arg2 is a list of special function codes. Only use the arg2 argument to specify special function codes in an extension table, that is, when the arg argument is a minus sign (-). See the discussion of the arg argument in the extension table method description.

Errors:

None

Example:

```
        .MCALL  .DRSPF                  ; Get macro

        .MACRO  ...                     ; Elision macro
        .ENDM   ...                     ; Elision - "act of dropping
                                        ;  out or suppressing"
                                        ;  allows ellipsis to assemble

        ...                             ; Handler continues...

        .DRSPF  <377,376,360,361>       ; Standard call
        .DRSPF  <350,352,367>           ; defining three groups
                                        ;  of code
                                        ; More are supported so
        .DRSPF  +XSPTAB                 ; point to extension table

        ...                             ; Handler continues beyond
                                        ;  block 0
XSPTAB:                                 ; SPFUN extension table
        .DRSPF  -,<200,202,207,203>     ; 20X group
        .DRSPF  -,<222,224,227,223>     ; 22X group
        .WORD   000000                  ; End of list

        .END
```

See also the .DREST example in this manual.

1.5.3.10 **.DRTAB (Device Handlers Only)** - The .DRTAB macro sets up a pointer to a list of DIGITAL-defined handler data tables that are part of the RT-11 distributed handlers. .DRTAB is included in a distributed handler when that handler contains more than one data table. A relationship exists between the .DRTAB macro and the .DREST macro and is as follows:

- When a distributed handler contains only one handler data table, .DREST is used to describe that table.
- When a distributed handler contains more than one handler data table, the .DRTAB macro is used to describe all those tables. The .DREST macro can be included in a handler that includes the .DRTAB macro because other information can be placed in the handler by .DREST. However, when the .DREST macro is included in a handler that also includes the .DRTAB macro, the .DREST macro does not contain the type and data arguments. (The information placed in the handler by those .DREST macro arguments would be destroyed [overwritten] by the .DRTAB macro type argument.)

Macro Call: .DRTAB type,addr,size

where:

type — is the handler data table format name in one to three RAD50 characters

addr — is a pointer to the handler data table

size — is the octal size in bytes of the handler data table

.DRTAB is invoked once for each handler data table. Each invocation of .DRTAB creates a 3-word area containing the values specified for the type, addr, and size arguments. A call to .DRTAB with no arguments specifies the end of that list of handler data tables.

.DRTAB places the pointer to the list of handler data tables in block 0 of the handler. The list of handler data tables and the data tables themselves are not located in block 0. When first invoked, .DRTAB sets up locations 70 through 74 in block 0 with the following contents:

Location	Contents after .DRTAB is invoked
70	-1 (indicates use of .DRTAB)
72	Pointer to list of handler data tables
74	Size in bytes of total list of handler data tables

1.5.3.11 **.DRUSE (Device Handlers Only)** - The .DRUSE macro sets up a pointer to a list of user-defined handler data tables. Use .DRUSE when you want to define your own handler data table(s).

Macro Call: .DRUSE type,addr,size

where:

type — is the handler data table format name in one to three RAD50 characters

addr — is a pointer to the handler data table

size — is the octal size in bytes of the handler data table

Invoke .DRUSE once for each user-defined handler data table in your handler. Each invocation of .DRUSE creates a 3-word area containing the values you specified for the type, addr, and size arguments. Call .DRUSE with no arguments to indicate the end of the list of handler data tables.

.DRUSE places the pointer to the list of handler data tables in block 0 of your handler. Do not place the list of handler data tables or the data tables themselves in block 0 of your handler. When you first invoke .DRUSE, it sets up location 106 in block 0 with the following contents:

Location	Contents after .DRUSE is invoked
106	Pointer to list of handler data tables

Example:

```
        .DRUSE  JFW,JIMS,SZJIM
        .DRUSE  JBM,JIMS2,SZJIM2
        .DRUSE

JIMS:   .WORD   ^rJFW,1,2,3,4,5,6,7,8
SZJIM=: .-JIMS
JIMS2:  .WORD   ^rJBM,10,1000,1000000
SZJIM2=: .-JIMS2
```

1.5.3.12 **.FPROT** - The .FPROT programmed request sets or resets the protection status of files.

See the RT-11 Programmer's Reference Manual for details and examples of the .FPROT programmed request.

1.5.3.13 **.PEEK** - The .PEEK programmed request returns in register R0 the contents of the specified low memory location (below 28K words) or the I/O page location you specify.

See the RT-11 Programmer's Reference Manual for details and examples of the .PEEK programmed request.

1.5.3.14 **.POKE** - The .POKE programmed request deposits the value you specify into a low memory location (below 28K words) or I/O page location.

See the RT-11 Programmer's Reference Manual for details and examples of the .POKE programmed request.

1.5.3.15 **.PVAL** - The .PVAL programmed request modifies or replaces the monitor fixed offset location you specify.

See the RT-11 Programmer's Reference Manual for details and examples of the .PVAL programmed request.

1.5.3.16 **.SFDAT** - The .SFDAT programmed request lets a program set or change the creation date in a file's directory entry.

See the RT-11 Programmer's Reference Manual for details and examples of the .SFDAT programmed request.

1.5.3.17 **SOB** - The SOB macro simulates the instruction "subtract one and branch if not equal" (SOB). That macro is useful for writing programs to run on any RT-11 supported processor, since the SOB instruction is invalid on some processors.

See the RT-11 Programmer's Reference Manual for more information on the SOB macro.

1.5.4 New System Subroutine Library (SYSLIB) Routines

This section describes the routines that have been added to SYSLIB.

1.5.4.1 **CALL$F** - CALL$F can be called only from a MACRO-11 program.

The CALL$F routine saves the contents of general registers R1 through R4 across a call to another routine that might destroy the contents of those registers. CALL$F saves the contents of R1 through R4 on the stack, calls the other routine, and then restores the saved register contents.

```
Form:  MOV    #rtn,R0
       MOV    #arg,R5
       CALL   CALL$F
```

where:

rtn — is the address of the routine you want to call. The current contents of registers 1 through 4 are preserved during execution of the called routine.

arg — is the starting address of the argument list for the routine you want to call.

Errors:

None (any errors are returned by the routine called by CALL$F)

Example:

```
        .GLOBAL GTLIN, CALL$F

                ; Program code

        MOV     #GTLIN,R0       ; Routine to call
        MOV     #PM,R5          ; Point to parameters
        CALL    CALL$F          ; Call routine and preserve registers

                ; Program code

PM:     .WORD   3
        .WORD   BUF
        .WORD   PROMPT
        .WORD   TYPE
TYP:    .ASCII  \PLAIN\
```

1.5.4.2 **IABTIO** - The IABTIO routine stops all outstanding I/O operations on a specified channel without terminating the job under FB and XM. This routine is simulated under SJ with a .WAIT directive.

See the RT-11 Programmer's Reference Manual for details and examples of IABTIO.

1.5.4.3 **IFPROT** - The IFPROT routine changes a specified file's protection status.

See the RT-11 Programmer's Reference Manual for details and examples of IFPROT.

1.5.4.4 **IFREER** - The IFREER subroutine detaches from a specified global region that has been attached to with the IGETR subroutine. IFREER can also eliminate that global region when you specify the type argument. IFREER does not eliminate a global region that is attached to another job but does detach the calling job from that global region.

See the RT-11 Programmer's Reference Manual for details on using IFREER.

1.5.4.5 **IGETR** - The IGETR subroutine attaches to a specified global region. IGETR can optionally initialize a global region by reading a portion of a file into the global region or by calling a specified subroutine.

See the RT-11 Programmer's Reference Manual for details on using IGETR.

1.5.4.6 **IGTDUS** - The IGTDUS subroutine provides information about a specified MSCP (DU) or TMSCP (MU) class device unit. Issue the IGTDUS subroutine only to MSCP or TMSCP class devices. Information returned by IGTDUS includes:

- Whether the device unit is off line, available, or on line.
- Whether the device media is removable (for example, the RC25).
- Whether the device unit is write protected.
- Whether the device controller supports bad block replacement.
- The octal number of physical addressable blocks in the device unit.
- The media name, such as RC25, RA80, RA81, RD31, RD51, RD52, RD53, RX33, or RX50.

Form: i=IGTDUS (devnam,chan,buf[,unit][,type][,work])

where:

devnam is the Radix-50 device name (MSCP or TMSCP class devices only)

chan is the integer specification for an RT-11 channel to be opened by IGTDUS

buf is the address of a 7-word buffer containing status information returned by IGTDUS. The information returned by IGTDUS in the status buffer includes:

buf+0 is the status information word. The following values can be returned in the status information word:

0 = The device unit is on line

1 = The device unit is available

2 = The device unit is off line

buf+2 is the unit flag word. One or more of the following values can be returned in the unit flag word:

200 = The media is removable

20000 = The media is write protected

100000 = The device controller supports bad block replacement

buf+4
buf+6 are the unit size words. The unit size is returned as a 28-bit octal value (16 bits in buf+4 and 12 bits in buf+6)

buf+10
buf+12
buf+14 is the media device name in 5-bit packed ASCII

unit is the unit number requested. See the type argument

type is the type of unit number requested. Type is a character constant and is specified as 'MSCP' or 'TMSCP'. Valid unit numbers are 0 through 255

work is a recommended 80(decimal)-word work area used internally by IGTDUS. If you do not define a work area, IGTDUS takes 80(decimal) words from the processor stack area. Allow for that when planning stack allocation to avoid stack overflow

Use the IGTDUS subroutine to determine the MSCP or TMSCP unit size before issuing the JREAD and JWRITE subroutines, or anytime you need status information concerning an MSCP or TMSCP device.

Errors:

i = 0 Normal return.

-1. Logic error. Retry the operation. If the error persists, submit an SPR to DIGITAL; see Appendix B.

-2. Logic error. Retry the operation. If the error persists, submit an SPR to DIGITAL; see Appendix B.

-3. Logic error. Retry the operation. If the error persists, submit an SPR to DIGITAL; see Appendix B.

-4. Attempt to read or write past end-of-file or invalid function value.

-5. Hard error occurred on channel.

-6. Channel is not open.

-7. Reserved.

-8. Reserved.

-9. Reserved.

-10. Reserved.

-11. Reserved.

-12. Reserved.

-13. Handler is not loaded.

-14. Handler is not installed.

-15. Channel is already in use.

-16. File indicated was not found on the device.

-17. File already open on a nonshareable device.

-18. Device does not support MSCP or TMSCP. Inappropriate device for IGTDUS subroutine.

-19. First character of type argument is not an 'm' or 'r'.

-20. Required argument is missing (devnam, chan, or buf arguments).

-21. Magtape file sequence number is invalid.

The following errors can be returned by IGTDUS if the .SERR programmed request is in effect:

i = -129. Called USR from completion routine.

-130. No device handler; this operation needs one.

-131. Error doing directory I/O.

-132. .FETCH error. An I/O error occurred while the handler was being used, or an attempt was made to load the handler over USR or KMON.

-133. Error reading an overlay.

-134. No more room for files in the directory.

-135. Reserved.

-136. Invalid channel number; number is greater than actual number of channels that exist.

-137. Invalid EMT, and invalid function code has been decoded.

-138. Reserved.

-139. Reserved.

-140. Invalid directory.

-141. Unloaded XM handler.

-142. Reserved.

-143. Reserved.

-144. Reserved.

-145. Reserved.

-146. Reserved.

Example:

```
C
C The following program uses IGTDUS to get and print
C the media type
C
        INTEGER*2 DEVNAM,ISTAT(7),IWORK(80)
        DATA DEVNAM/3RDU0/
        ICHAN = IGETC()
        IF (ICHAN.LT.0) STOP 'NO CHANNEL'
        ICODE = IGTDUS(DEVNAM,ICHAN,ISTAT,,,IWORK)
        IF (ICODE .LT. 0) STOP 'BAD IGTDUS'
        IF (ISTAT(1) .NE. 0 ) STOP 'DRIVE ERROR'
        TYPE *,'THE MEDIA TYPE FOR THIS DRIVE IS '
        TYPE 100,(ISTAT(J),J= 5,7)
100     FORMAT(' ',3A2)
        CALL EXIT
        STOP
        END
```

1.5.4.7 **IPUT** - The IPUT routine modifies or replaces the value of a monitor fixed offset location.

See the RT-11 Programmer's Reference Manual for details and examples of IPUT.

1.5.4.8 **ISDTTM** - The ISDTTM routine modifies the current date and time values stored in the monitor.

See the RT-11 Programmer's Reference Manual for details and examples of ISDTTM.

1.5.4.9 **ISFDAT** - The ISFDAT routine sets or changes the creation date in a file's directory entry.

See the RT-11 Programmer's Reference Manual for details and examples of ISFDAT.

1.5.4.10 **JREAD/JREADC/JREADW** - The functions JREAD, JREADC, and JREADW use non-file-structured access to transfer into memory a specified number of words from an MSCP device. JREAD, JREADC, and JREADW are especially useful because they use a 28-bit starting block number and can therefore read from any block on any DU device.

Those functions require a queue element. Allow for that queue element when the IQSET function is executed.

JREAD

The JREAD function transfers into memory a specified number of words from an MSCP device associated with the indicated channel. The monitor returns control to the user program immediately after the JREAD function is initiated. No special action is taken when the transfer is completed.

Form: i = JREAD (wcnt,buff,jblock,chan[,area])

where:

wcnt — is the integer number of words to be transferred

buff — is an array to be used as the buffer; that array must contain at least wcnt words

jblock — is the 28-bit starting block number of the MSCP device to be read. The 28-bit starting block number consists of 16 bits in the first jblock word and the low 12 bits in the second jblock word. The high 4 bits in the second jblock word are reserved.

The first block of the MSCP device is physical block 0. The first block of each MSCP device partition is logical block 0. The jblock argument is offset from physical or logical block number 0. The jblock argument refers to a physical block offset when the MSCP disk has not been partitioned, or to a logical block offset when the MSCP disk has been partitioned. When an MSCP disk has been partitioned, the logical block number you address with the jblock argument depends on which partition is assigned to the RT-11 unit (DU0: through DU7:).

The jblock argument must be updated (incremented) when necessary. For example, if the program is reading two blocks at a time, jblock should be incremented by 2 for each read

chan — is the integer specification for the RT-11 channel to be used

area — is a recommended 80(decimal)-word work area that is used internally by JREAD. If you do not define a work area, JREAD takes 80(decimal) words from the processor stack area. Allow for that when planning stack allocation to avoid stack overflow

Issue an IWAIT function when the user program needs to access the data read on the specified channel. IWAIT makes sure that the JREAD operation has been completed. IWAIT indicates if a hard error occurs during the transfer.

Errors:

i = n	Normal return; n equals the number of words requested. If wcnt is greater than the number of words remaining in the MSCP device, n is the number of words remaining in the MSCP device.
= -2	Hardware error occurred on channel.
= -3	Specified channel is not open.
= -4	Invalid device (not DU).

Example:

See the JREADW example in this manual.

JREADC

The JREADC function transfers into memory a specified number of words from the indicated channel. The monitor returns control to the user program immediately after the JREADC function is initiated. When the operation is complete, the monitor enters the specified assembly language routine (crtn) as an asynchronous completion routine.

Form: i = JREADC (wcnt,buff,jblock,chan,[area],crtn)

where:

wcnt — is the integer number of words to be transferred

buff — is an array to be used as the buffer; that array must contain at least wcnt words

jblock — is the 28-bit starting block number of the MSCP device to be read. The 28-bit starting block number consists of 16 bits in the first jblock word and the low 12 bits in the second jblock word. The high 4 bits in the second jblock word are reserved.

The first block of the MSCP device is physical block 0. The first block of each MSCP device partition is logical block 0. The jblock argument is offset from physical or logical block number 0. The jblock argument refers to a physical block offset when the MSCP disk has not been partitioned, or to a logical block offset when the MSCP disk has been partitioned. When an MSCP disk has been partitioned, the logical block number you address with the jblock argument depends on which partition is assigned to the RT-11 unit (DU0: through DU7:).

The jblock argument must be updated (incremented) when necessary. For example, if the program is reading two blocks at a time, jblock should be incremented by 2 for each read

chan — is the integer specification for the RT-11 channel to be used

area — is a recommended 80(decimal)-word work area that is used internally by JREADC. If you do not define a work area, JREADC takes 80(decimal) words from the processor stack area. Allow for that when planning stack allocation to avoid stack overflow

crtn — is the assembly language routine to be activated when the transfer is complete. That routine must be specified in the EXTERNAL statement in the FORTRAN routine that issues the JREADC function

Errors:

See the errors under JREAD.

Example:

See the JREADW example in this manual.

JREADW

The JREADW function transfers a specified number of words from the indicated channel into memory. The monitor returns control to the user program when the transfer is complete or when an error is detected.

Form: i = JREADW (wcnt,buff,jblock,chan[,area])

where:

wcnt — is the integer number of words to be transferred

buff — is an array to be used as the buffer; that array must contain at least wcnt words

jblock — is the 28-bit starting block number of the MSCP device to be read. The 28-bit starting block number consists of 16 bits in the first jblock word and the low 12 bits in the second jblock word. The high 4 bits in the second jblock word are reserved.

The first block of the MSCP device is physical block 0. The first block of each MSCP device partition is logical block 0. The jblock argument is offset from physical or logical block number 0. The jblock argument refers to a physical block offset when the MSCP disk has not been partitioned, or to a logical block offset when the MSCP disk has been partitioned. When an MSCP disk has been partitioned, the logical block number you address with the jblock argument depends on which partition is assigned to the RT-11 unit (DU0: through DU7:).

The jblock argument must be updated (incremented) when necessary. For example, if the program is reading two blocks at a time, jblock should be incremented by 2 for each read

chan — is the integer specification for the RT-11 channel to be used

area — is a recommended 80(decimal)-word work area that is used internally by JREADW. If you do not define a work area, JREADW takes 80(decimal) words from the processor stack area. Allow for that when planning stack allocation to avoid stack overflow

Errors:

See the errors under JREAD.

Example:

```
C
C The following example reads block number 65,540 and writes
C it to block number 65,685.
C
C       JREADW/JWRITW
C
        INTEGER*2 DBLK(4),BUFFER(256),BLKNO(2),BLKNO2(2)
        INTEGER*2 WORKA(80)                ! WORK AREA
        DATA    DBLK/3RDU0,0,0,0/
        DATA    BLKNO/5,1/                 ! INPUT 28-BIT BLOCK
        DATA    BLKNO2/150,1/              ! OUTPUT 28-BIT BLOCK
        ICHAN= IGETC()
        IF (ICHAN.LT.0) STOP 'NO CHANNEL AVAILABLE'
        IF (IFETCH(DBLK).NE.0) STOP 'BAD FETCH'
        IF (LOOKUP(ICHAN,DBLK).LT.0) STOP 'BAD LOOKUP'
        ICODE = JREADW(256,BUFFER,BLKNO,ICHAN,WORKA)
        IF (ICODE+1)300,200,10
10      ICODE = JWRITW(256,BUFFER,BLKNO2,ICHAN,WORKA)
        IF (ICODE+1)300,200,100
100     CALL    PRINT('SUCCESSFUL OPERATION')
        CALL    EXIT
        STOP
200     STOP    'END OF DEVICE'
300     STOP    'HARDWARE ERROR'
        END
```

1.5.4.11 **JWRITE/JWRITC/JWRITW** - The functions JWRITE, JWRITC, and JWRITW use non-file-structured access to transfer a specified number of words from memory to an MSCP (DU) device. JWRITE, JWRITC, and JWRITW are especially useful because they use a 28-bit starting block number and can therefore write to any block on any DU device.

If you are unsure about the physical size of an MSCP device, use the IGTDUS subroutine to determine the size of that device. If you specify a block number higher than the device size returned by IGTDUS, you can corrupt formatting and bad-block replacement information contained on the device. That makes the device unusable, and you must reformat it. Formatting a device removes all information contained on that device. Be very careful to specify a starting block that is lower than the device size returned by IGTDUS.

The JWRITE, JWRITC, and JWRITW functions require a queue element. Allow for that queue element when the IQSET function is executed.

JWRITE

The JWRITE function transfers a specified number of words from memory to the specified channel. The monitor returns control to the user program immediately after queuing the request. No special action is taken upon completion of the operation.

Form: i = JWRITE (wcnt,buff,jblock,chan[,area])

where:

wcnt — is the integer number of words to be transferred

buff — is an array to be used as the output buffer

jblock — is the 28-bit starting block number of the MSCP device to be read. The 28-bit starting block number consists of 16 bits in the first jblock word and the low 12 bits in the second jblock word. The high 4 bits in the second jblock word are reserved.

The first block of the MSCP device is physical block 0. The first block of each MSCP device partition is logical block 0. The jblock argument is offset from physical or logical block number 0. The jblock argument refers to a physical block offset when the MSCP disk has not been partitioned, or to a logical block offset when the MSCP disk has been partitioned. When an MSCP disk has been partitioned, the logical block number you address with the jblock argument depends on which partition is assigned to the RT-11 unit (DU0: through DU7:). Do not specify a block number higher than the physical size of the device (the physical size of an MSCP device is returned by the IGTDUS function).

The jblock argument must be updated (incremented) when necessary. For example, if the program is reading two blocks at a time, jblock should be incremented by 2 for each write

chan — is the integer specification for the RT-11 channel to be used. You must obtain this channel through an IGETC call, or you can use channel 16(decimal) or higher if you have done an ICDFN call

area — is a recommended 80(decimal)-word work area that is used internally by JWRITE. If you do not define a work area, JWRITE takes 80(decimal) words from the processor stack area. Allow for that when planning stack allocation to avoid stack overflow

Errors:

i = n	Normal return; n equals the number of words written.
= -2	Hard error occurred on the channel.
= -3	Channel is not open.
= -4	Invalid device.

Example:

See the JREADW/JWRITW example in this manual.

JWRITC

The JWRITC function transfers a specified number of words from memory to the specified channel. The monitor queues the request and returns control to the user program. When the transfer is complete, the monitor enters the specified assembly language routine (crtn) as an asynchronous completion routine.

Form: i = JWRITC (wcnt,buff,jblock,chan,[area],crtn)

where:

wcnt is the integer number of words to be transferred

buff is an array to be used as the output buffer

jblock is the 28-bit starting block number of the MSCP device to be read. The 28-bit starting block number consists of 16 bits in the first jblock word and the low 12 bits in the second jblock word. The high 4 bits in the second jblock word are reserved.

The first block of the MSCP device is physical block 0. The first block of each MSCP device partition is logical block 0. The jblock argument is offset from physical or logical block number 0. The jblock argument refers to a physical block offset when the MSCP disk has not been partitioned, or to a logical block offset when the MSCP disk has been partitioned. When an MSCP disk has been partitioned, the logical block number you address with the jblock argument depends on which partition is assigned to the RT-11 unit (DU0: through DU7:). Do not specify a block number higher than the physical size of the device (the physical size of an MSCP device is returned by the IGTDUS function).

The jblock argument must be updated (incremented) when necessary. For example, if the program is reading two blocks at a time, jblock should be incremented by 2 for each write

chan is the integer specification for the RT-11 channel to be used. You must obtain this channel through an IGETC call, or you can use channel 16(decimal) or higher if you have done an ICDFN call

area — is a recommended 80(decimal)-word work area that is used internally by JWRITC. If you do not define a work area, JWRITC takes 80(decimal) words from the processor stack area. Allow for that when planning stack allocation to avoid stack overflow

crtn — is the assembly language routine to be activated when the transfer is complete. That routine must be specified in the EXTERNAL statement in the FORTRAN routine that issues the JWRITC function

Errors:

See the errors under JWRITE.

Example:

See the JREADW/JWRITW example in this manual.

JWRITW

The JWRITW function transfers a specified number of words from memory to the specified channel. The monitor returns control to the user program when the transfer is complete.

Form: i = JWRITW (wcnt,buff,jblock,chan[,area])

where:

wcnt — is the integer number of words to be transferred

buff — is an array to be used as the output buffer

jblock — is the 28-bit starting block number of the MSCP device to be read. The 28-bit starting block number consists of 16 bits in the first jblock word and the low 12 bits in the second jblock word. The high 4 bits in the second jblock word are reserved.

The first block of the MSCP device is physical block 0. The first block of each MSCP device partition is logical block 0. The jblock argument is offset from physical or logical block number 0. The jblock argument refers to a physical block offset when the MSCP disk has not been partitioned, or to a logical block offset when the MSCP disk has been partitioned. When an MSCP disk has been partitioned, the logical block number you address with the jblock argument depends on which partition is assigned to the RT-11 unit (DU0: through DU7:). Do not specify a block number higher than the physical size of the device (the physical size of an MSCP device is returned by the IGTDUS function)

The jblock argument must be updated (incremented) when necessary. For example, if the program is reading two blocks at a time, jblock should be incremented by 2 for each write

chan — is the integer specification for the RT-11 channel to be used. You must obtain this channel through an IGETC call, or you can use channel 16(decimal) or higher if you have done an ICDFN call

area — is a recommended 80(decimal)-word work area that is used internally by JWRITW. If you do not define a work area, JWRITW takes 80(decimal) words from the processor stack area. Allow for that when planning stack allocation to avoid stack overflow.

Errors:

See the errors under JWRITE.

Example:

See the JREADW/JWRITW example in this manual.

1.5.5 New Handlers

RT-11 V5 includes the following new handlers.

1.5.5.1 **DU (MSCP Handler)** - The DU handler supports disk systems that implement the mass storage communication protocol (MSCP). Presently, RT-11 supports the following MSCP storage systems: RX33 and RX50 diskettes, RC25 disk, RA80 and RA81 disks, RD31, RD51, RD52, and RD53 disks.

Changes have been made to the DU handler. See Chapter 2 for details.

See Section 10.11 of the RT-11 Software Support Manual for more information on the DU handler.

1.5.5.2 **DW (Professional 350 and 380 Hard Disk Handler)** - The DW handler supports the 5M-byte RD50, 10M bytes of the 20M-byte RD31, the 10M-byte RD51, and the 33M-byte RD52 fixed hard disks. RT-11 supports only drive unit DW0.

1.5.5.3 **DZ (Professional 300 Series Diskette Handler)** - The DZ handler supports RX50 diskette drives for the Professional 300 series computers. RT-11 supports drive units DZ0 and DZ1.

1.5.5.4 **MU (TMSCP handler)** - The MU handler supports magtape systems that use the tape mass storage communication protocol (TMSCP). The MU handler supports up to four units. Each unit requires a separate controller.

You should LOAD the MU handler before you use a TMSCP magtape device. A TMSCP device operates much faster when you LOAD the MU handler. Because of the size of the MU handler, you should UNLOAD the MU handler when you are not using the TMSCP device, for example, if you are only using the TMSCP device for backup operations.

See the RT-11 Software Support Manual for information about MU. See Section 3.4.4 for changes and corrections to MU information found in the RT-11 Software Support Manual.

1.5.5.5 **NC, NQ, and NU (Ethernet Handlers)** - The NC, NQ, and NU handlers provide support for Ethernet-class devices. NC supports the DECNA Ethernet controller for Professional 300 series processors. NQ supports the DEQNA Ethernet controller for Q-BUS PDP-11 processors. NU supports the DEUNA Ethernet controller for UNIBUS processors. Each Ethernet handler supports one Ethernet controller. The NC, NQ, and NU handlers run only under the XM monitor. The new NU handler uses the same .SPFUN function codes and has the same restrictions as the NC and NQ handlers.

See the RT-11 Software Support Manual for complete information on using the NC and NQ Ethernet handlers. Information in that manual for the NC and NQ handlers also applies to the new NU handler. Section 3.4.4 of this manual contains an example program that illustrates Ethernet unit allocation for the NC, NQ, and NU Ethernet handlers.

1.5.5.6 **PI (Professional Interface Handler)** - The Professional interface (PI) handler supports the keyboard, video display, and system clock for Professional 300 series computers. Although you cannot alter PI handler characteristics or direct I/O operations to PI, PI.SYS (or PIX.SYS under the XM monitor) must reside on your system volume if you are running on a Professional 300 series computer.

1.5.5.7 **SD (DBG-11 Debugger Pseudohandler)** - Four variants of the SD handler (SDH, SDHX, SDS, SDSX) provide support for the DBG-11 symbolic debugging package:

SDH.SYS	Hardware I/O version of DBG-11 for use with the SJ and FB monitors
SDHX.SYS	Hardware I/O version of DBG-11 for use with the XM monitor
SDS.SYS	Software I/O version of DBG-11 for use with the SJ and FB monitors
SDSX.SYS	Software I/O version of DBG-11 for use with the XM monitor

Refer to the DBG-11 Symbolic Debugger User's Guide for complete information.

1.5.5.8 **SP (Transparent Spooler Handler)** - The SP pseudohandler supports the transparent spooler (SPOOL). The SP handler intercepts output directed to the line printer and sends it to SPOOL for temporary storage and printing.

1.5.5.9 **XC and XL (Communication Port Handler)** - XC and XL are non-file-structured communications handlers. They are designed to support the virtual terminal communication package, VTCOM. However, their design does not preclude their use in other communication programs. The XC handler supports the Professional 300 series computer communication port. The XL handler supports DL(V)-11 communication ports. XC or XL (depending on your system) is required when you use VTCOM.

XC and XL support the VTCOM utility using .READx, .WRITx, and .SPFUN programmed requests. The following describes information specific to the XC and XL handlers for those requests.

.READx and .WRITx Support

The XC and XL handlers support the .READ, .READC, .READW, .WRITC, .WRITE, and .WRITW requests. You use the .READx and .WRITx requests with XC and XL handlers as described in the RT-11 Programmer's Reference Manual. Note, however, the following additional information:

- You should specify the value 0 in the blk argument for the first request to XC or XL. All subsequent calls should specify a nonzero value for the blk argument.
- NULL characters are ignored by XC and XL during both .READs from and .WRITEs to the handlers.
- XC and XL only pass 7-bit data. The eighth (high-order) bit is stripped from each byte.

.SPFUN Support

In general, the XC and XL handlers support the .SPFUN request as described in the RT-11 Programmer's Reference Manual. Note, however, the following general information:

- You should specify the value 0 in the blk argument for the first request to XC or XL. All subsequent calls should specify a nonzero value for the blk argument.
- NULL characters are ignored by the XC and XL handlers; NULL characters are not stored or sent. However, .SPFUN 203 uses a NULL character to signal the end of available data (see .SPFUN 203 below).
- XC and XL only pass 7-bit data. The eighth (high-order) bit is stripped from each byte.

The XC and XL handlers support the following special function codes. Specific information about using a particular .SPFUN request with XC or XL is included in the description for that request.

Code	Label	Description
201	SP.CLR	Resets the internal flag, indicating a received XOFF. Then sends an XON to the host. Example: .SPFUN #area,#chan,#201,#buf,#wcnt,#blk[,#crtn]
202	SP.BRK	Sets or resets the state of the BREAK bit in the serial interface. Transition of the BREAK bit from 0 to 1 to 0 can get the attention of certain communications devices, such as terminal concentrators. The wcnt argument is a flag that indicates whether the BREAK bit should be set or reset. Specify a value of 1 for the wcnt argument to set the BREAK bit; specify 0 to reset it. DIGITAL recommends you use some time delay between turning the bit on and turning it off; do that by sending one or two characters. Examples: To turn on (set) the BREAK bit: .SPFUN #area,#chan,#202,#buf,#1,#blk[,#crtn] To turn off (reset) the BREAK bit: .SPFUN #area,#chan,#202,#buf,#0,#blk[,#crtn]
203	SP.SRD	Performs a special read from the handler. The wcnt argument specifies the number of bytes to be read. The read is completed when one of the following conditions is met: • The number of bytes specified in the wcnt argument have been transferred. • The available characters have been transferred, when the number of available characters was less than the value specified in the wcnt argument. • One character has been transferred, when no characters were available when the request was issued. The byte following the last transferred character contains a NULL. You must allow for that NULL byte in your buffer.

Example:

The following example reads no more than six (but at least one) characters from XC or XL and places them in the buffer RCVBUF. RCVBUF must be at least seven bytes in length to receive the six characters and the NULL byte.

```
.SPFUN  #area,#chan,#203,#RCVBUF,#6,#blk[,#crtn]
```

204 SP.STS

Returns the driver status in the first word of the specified buffer. .SPFUN 204 always returns one word.

The high byte of the returned word contains the driver support level. The driver support level number will be updated as support is changed in the XC and XL handlers. Programs should verify operation with an established driver support level. The driver support level for RT-11 V5.4 is 17(decimal).

The low byte contains the status of two internal flags and a modem control signal. The significant bits of the low byte are:

Bit	Meaning
0	Set if an XOF has been sent to the host.
1	Set if an XOFF has been received from the host.
2	Set if the CLEAR TO SEND line is set.
3-7	Reserved

Example:

The following example returns the driver support level in the high byte, and the status of internal flags in the low byte of the 1-word buffer STATUS.

```
.SPFUN  #area,#chan,#204,#STATUS,#wcnt,#blk[,#crtn]
```

205 SP.OFF

Sets a flag that disables interrupts when the program exits. DIGITAL recommends you issue .SPFUN 205 before your program exits.

Example:

```
.SPFUN  #area,#chan,#205,#buf,#wcnt,#blk[,#crtn]
```

206 SP.DTR

Sets or resets the state of the DTR modem control signal. Setting (asserting) DTR can cause modems to answer an incoming call. Resetting (deasserting) DTR can cause modems to terminate a current call. DTR can also get the attention of certain communications devices such as the Mini-Exchange. Specify a value of 1 for the wcnt argument to set the DTR control signal; specify 0 to reset the DTR control signal.

Not all interfaces support the DTR control signal. On interfaces that do not support DTR, the setting or resetting of DTR has no effect.

Example:

The following example sets the DTR control signal:

.SPFUN #area,#chan,#206,#buf,#1,#blk[,#crtn]

The following example resets the DTR control signal:

.SPFUN #area,#chan,#206,#buf,#0,#blk[,#crtn]

1.5.5.10 **VM (Memory Disk Handler)** - The VM handler allows memory above 28K words to be used as though it were a disk device. That virtual device can be used as the system volume or as a data volume.

See Appendix C for information on using VM as your system volume.

RT-11 requires using VM to support the use of memory above 256K bytes on UNIBUS machines. Therefore, DIGITAL does not recommend the removal of the VM handler or changing the base on 22-bit UNIBUS systems (11/24, 11/44, 11/70, 11/84).

Refer to Section 10.12 of the RT-11 Software Support Manual for more information on the VM handler.

1.5.6 New Extended Memory Subroutine

A new extended memory subroutine, XDEALC, has been added to the file XMSUBS.MAC.

XDEALC returns a specified section of extended memory to the free memory list maintained by RT-11. The entry point for XDEALC is $P1EXT-18(decimal).

The address and size of the section of extended memory to be returned are specified in units of 32(decimal) words. Load R1 with the starting address divided by 32(decimal), and R2 with the size of the region in units of 32(decimal) words.

In the following example, R0 contains the address of $P1EXT and XDEALC is -18(decimal).

```
mov     #region_address,R1      ; Address in units of 32. words
mov     #region_size,R2         ; Size in units of 32. words
CALL    XDEALC(R0)
```

On return from XDEALC, the carry bit is clear if the memory was returned. If the carry bit is set, the memory was not returned because the free memory has become too fragmented.

XDEALC destroys the contents of R1 and R2. If you want to preserve the contents of those registers across the call, you must save them.

1.5.7 New System Generation Procedures and Options

The system generation process is controlled by the new indirect control file processor, IND, and includes support for new options. See the RT-11 System Generation Guide for a complete description of system generation.

1.5.7.1 **New System Generation Procedures** - The RT-11 V5 system generation procedures are no longer controlled by a FORTRAN IV program (SYSGEN.SAV). Instead, system generation is now run using the IND control file processor. The system generation procedures, however, remain basically the same: system generation produces new monitors and handlers depending on your answers to SYSGEN.COM dialog questions. (SYSGEN.COM is an IND control file that replaces SYSGEN.CND and the device section of SYSTBL.CND.)

The following are the major changes to the system generation procedures:

- You can create an answer file to preserve your responses during a system generation session. You can use this answer file during later system generation sessions to recreate the same system without answering the dialog questions again.

- After answering the system generation dialog questions, you now have the opportunity to change some of your answers. If you are using an answer file, you can also change some responses. The new responses are recorded in the output answer file, if you request one.

- All work files associated with a particular system generation session use the same name as the answer file but have a different file type. For example, if the answer file is NEWSYS.ANS, the work files are named NEWSYS.BLD, NEWSYS.MON, NEWSYS.DEV, NEWSYS.CND, and so on. The default answer file is SYSGEN.ANS.

- The system generation dialog lets you define system conditionals for which there are no system generation questions. For example, you must define the conditional LIGH$T = 1 to enable support for the idle-loop light pattern. Conditional definitions are then included in the answer file and the .CND (conditional) file created during system generation. This new feature lets you define system conditionals without having to edit the .CND file. The .ANS file will contain all system generation information.

  See the RT-11 System Generation Guide for more information on defining system conditionals during system generation.

- You can create the .CND conditional files for the distributed monitors by performing a system generation and using the appropriate monitor answer file (BL.ANS, SJFB.ANS, or XM.ANS) as input.

- Instead of choosing the long or short form of the dialog, you can request explanatory text for individual questions by typing <ESC><RET>.

- SYSGEN.TBL, a new file produced during the system generation session, contains the device tables that were included in the file SYSTBL.MAC. SYSGEN.TBL is included during the assembly and link procedure for all monitors you generate. If you use an answer file during the system generation session, the resulting device tables file will have the same name as the answer file but with the file type .TBL. The new file TRMTBL.MAC, which contains the multiterminal tables, is distributed on the RT-11 V5 distribution kit and is included in the assembly and link procedure only if you request multiterminal support during system generation.

- The system generation procedure for specifying device support has changed. Instead of asking if you want support for each device individually, the system generation dialog now repeats one question to ask you to list all the devices you want to support. If you type a question mark (?) followed by a carriage return, the system lists the codes for all the devices you can support, and marks each device for which you have already included support with an asterisk (*). The system asks appropriate questions about support for additional controllers, and CSR and vector addresses, as you select each device.

  You can also specify your own device handlers during the system generation procedure. They will be included in the build command files generated.

- You can now specify physical or logical device names for the devices to be used for building your system at the end of the system generation process.

- The clock for the Professional 300 series works differently from a PDP-11 line clock. However, RT-11 supports the Professional clock in a mode similar to a 60 hertz line clock. Therefore, regardless of whether you choose 50 or 60 hertz during system generation, if you are running on a Professional computer, a 60 hertz clock rate will be used.

- Optimized PDT execution can be included by setting the conditional PDT$OP to 1. However, monitors generated with PDT$OP will not run on processors with programmable baud interfaces (DLARTS) or on the SBC-11/21.

1.5.7.2 **New System Generation Options** - The system generation dialog includes questions to generate support for the following new options:

- DU handler initiated bad-block replacement
- DU multiple port booting
- High-speed ring buffer
- Up to 9600 baud for DZ11 and DZV11 lines
- User command linkage
- Fetchable handlers under XM
- Global .SCCA support

- Exclusion of Professional printer port support from LS handler
- Generation of support for the following new devices:

  TMSCP magtapes (MU) - TK50

  Ethernet handlers (NC, NQ, and NU)

  TSV05/TSU05 magtape (MS)

  MSCP disks (DU) - RA80/RA81, RC25, RD31, RD51, RD52, RD53, RX33, RX50

  Professional 325, 350, and 380 devices: RD50, RD31, RD51, RD52 (DW), and RX50 (DZ)

  SP transparent spooler handler

  XL communication package handler (PDP-11 and PDT-11/150)

  XC communication package handler (Professional 300 series)

1.5.8 New Procedures to Back up Distribution to Magtapes

The following sections describe the procedures you use to back up the RT-11 distribution to 800 bits/in, 1600 bits/in, or TK50 magtape.

1.5.8.1 Backing up Distribution to 800 bits/in or 1600 bits/in Magtape

If you install a system distributed on magtape to run on hard disk, use the indirect command file MTB.COM or MSB.COM to preserve the distribution magtapes. (If your system is distributed on TK50 magtape, refer to Section 1.5.8.2.)

Use MTB.COM if your magtape is written at 800 bits/in. Use MSB.COM if your magtape is written at 1600 bits/in. MTB.COM or MSB.COM replaces DISMT1.COM and DISMT2.COM.

The new indirect command files simplify the backup procedure. You no longer have to run a complete backup procedure for each magtape. Using MTB.COM or MSB.COM lets you run one indirect command file and switch magtapes in the middle of the procedure.

To use MTB.COM or MSB.COM:

Follow the directions in Section 8.1 of the RT-11 Installation Guide to bootstrap the first distribution magtape. Then, after you copy the files from distribution magtape 1 to the disk, replace distribution magtape 1 with distribution magtape 2. Copy all the files from distribution magtape 2 to the disk as well. Remove distribution magtape 2 from the tape drive. Reboot the disk.

Complete the following steps to preserve the distribution magtapes. (These four steps replace the six steps in Section 8.2 of the RT-11 Installation Guide.)

1. Mount a blank magtape in 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.

2. Copy from the disk to your blank magtape those files that were on distribution magtape 1. To copy those files, invoke the indirect command file MTB.COM if your magtape is 9-track, 800 bits/in. If your magtape is 9-track, 1600 bits/in, invoke MSB.COM. (The commands you enter are shown below.)

   MTB.COM or MSB.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 BIN: to your disk device and the logical name KIT: to your tape drive. In the commands, xx is MT, MS or MM, and yy is your disk.

   If your magtape is written at 800 bits/in, enter:

```
.ASSIGN xxn: KIT:<RET>
.ASSIGN yyn: BIN:<RET>
.$@MTB<RET>
```

   If your magtape is written at 1600 bits/in, enter:

```
.ASSIGN xxn: KIT:<RET>
.ASSIGN yyn: BIN:<RET>
.$@MSB<RET>
```

   (The commands in the indirect file appear on the terminal.)

3. Replace the backup magtape with a blank magtape. When MTB.COM or MSB.COM finishes copying files to the first magtape, it prints the following question on the terminal:

```
xxn:/Initialize; Are you sure?
```

   Before you type a response, rewind the newly created backup magtape, remove it, and label it "Backup RT-11 V05 1/2".

   Then, 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.

4. Copy from the disk to the blank magtape those files that were on distribution magtape 2.

   Type a Y<RET> in response to the "Initialize; Are you sure?" question on the terminal.

   MTB.COM or MSB.COM initializes the blank magtape and copies a duplicate of distribution magtape 2. When the indirect command file finishes executing, the monitor prompt (.) appears on your terminal. Rewind the newly created backup magtape, remove it, and label it "Backup RT-11 V05 2/2".

   Store the distribution magtapes.

1.5.8.2 **Backing up Distribution to TK50 Magtape** - If you install a system distributed on TK50 magtape to run on hard disk, use the new indirect command file MUB.COM to preserve the distribution TK50 magtape.

To use MUB.COM:

Follow the directions in Section 8.1 of the RT-11 Installation Guide to bootstrap the distribution TK50 magtape. Copy the files from the distribution TK50 magtape to the disk. Reboot the disk.

Complete the following steps to preserve the distribution TK50 magtape. (These three steps replace the six steps in Section 8.2 of the RT-11 Installation Guide.)

1. Slide the write-protect switch on the front of a blank TK50 cartridge to the right to enable writing on the tape. Insert the blank TK50 magtape cartridge in the tape drive.

2. Invoke the indirect command file MUB.COM to copy the files from the disk to your blank TK50 magtape. MUB.COM initializes the blank TK50 magtape, writes the primary bootstrap on it, and copies files in their correct order from the disk to the TK50 magtape. Before you invoke MUB.COM, use the ASSIGN command to assign the logical name BIN: to your disk device and the logical name KIT: to your tape drive. In the following commands, yy is your disk.

```
. ASSIGN MUn: KIT:<RET>
. ASSIGN yyn: BIN:<RET>
. $@MUB<RET>
```

   (The commands in the indirect file appear on the terminal.)

3. When MUB.COM finishes executing, the monitor prompt (.) appears on your terminal. Rewind the newly created backup TK50 magtape, remove it, and label it "Backup RT-11 V05 1/1".

   Store the distribution TK50 magtape.

1.5.9 New Procedures to Create a Bootable Magtape

MTB.COM, MSB.COM, and MUB.COM are new procedures that allow you to create a bootable magtape while you run RT-11 from a disk.

Use MTB.COM if your magtape is written at 800 bits/in. Use MSB.COM if your magtape is written at 1600 bits/in. Section 1.5.9.1 describes using MTB.COM and MSB.COM.

Use MUB.COM if your magtape is a TK50. Section 1.5.9.2 describes using MUB.COM.

1.5.9.1 **Creating an 800 bits/in or 1600 bits/in Bootable Magtape** - To use MTB.COM or MSB.COM:

1. Mount a blank magtape in the tape drive. Make sure that the magtape is write enabled and on line.

2. Invoke the indirect command file MTB.COM or MSB.COM. (The commands you enter are shown below.)

   MTB.COM or MSB.COM initializes the blank magtape, writes the primary bootstrap on it, and copies files in their correct order from the disk to the magtape. Before you invoke the indirect command file, use the ASSIGN command to assign the logical name BIN: to your disk device and the logical name KIT: to your tape drive. In the commands, xx is MT, MS, or MM, and yy is your disk.

   If your magtape is written at 800 bits/in, enter:

```
.ASSIGN xxn: KIT:<RET>
.ASSIGN yyn: BIN:<RET>
.$@MTB<RET>
```

   If your magtape is written at 1600 bits/in, enter:

```
.ASSIGN xxn: KIT:<RET>
.ASSIGN yyn: BIN:<RET>
.$@MSB<RET>
```

   (The commands in the indirect file appear on the terminal.)

3. Replace the magtape in the tape drive with a blank magtape. When MTB.COM or MSB.COM finishes copying files to the first magtape, it prints the following question on the terminal:

```
xxn:/Initialize; Are you sure?
```

   Before you type a response, rewind the newly created bootable magtape, remove it, and label it "RT-11 V05 1/2".

   Then, mount another blank magtape, making sure that it is write enabled and on line.

4. Type a Y<RET> in response to the "Initialize; Are you sure?" question on the terminal. MTB.COM or MSB.COM initializes the blank magtape and copies the remaining files in their correct order from the disk to the magtape.

   When the indirect command file finishes executing, the monitor prompt (.) appears on your terminal. Rewind the newly created magtape, remove it, and label it "RT-11 V05 2/2".

1.5.9.2 **Creating a Bootable TK50 Magtape** - To use MUB.COM:

1. Slide the write-protect switch on the front of a blank TK50 cartridge to the right to enable writing on the tape. Insert the blank TK50 magtape cartridge in the tape drive.

2. Invoke the indirect command file MUB.COM to copy the files from the disk to your blank TK50 magtape. MUB.COM initializes the blank TK50 magtape, writes the primary bootstrap on it, and copies a duplicate of the distribution TK50 magtape. Before you invoke MUB.COM, use the ASSIGN command to assign the logical name BIN: to your disk device and the logical name KIT: to your tape drive. In the following commands, yy is your disk.

```
.ASSIGN MUn: KIT:<RET>
.ASSIGN yyn: BIN:<RET>
.$@MUB<RET>
```

   (The commands in the indirect file appear on the terminal.)

3. When MUB.COM finishes executing, the monitor prompt (.) appears on your terminal. Rewind the newly created bootable TK50 magtape, remove it, and label it "RT-11 V05 1/1".

1.5.10 New Error Messages

Error messages have been added for the following software components:

BATCH (job control utility)
BUP (backup utility)
DUP (device utility program)
FORMAT (formatting utility)
IND (indirect control file processor)
KMON (keyboard monitor)
LD (logical disk subsetting facility)
LINK (linker utility)
MDUP (magtape utility program)
PIP (peripheral interchange program)
PRO/GIDIS (graphics utility)
SETUP (hardware setup utility)
SL (single-line editor)
SPOOL (transparent spooling utility)
SYSGEN (system generation procedure)
TRANSF (VTCOM file transfer program)
UCL (user command linkage utility)
VTCOM (virtual terminal communication package)

New error messages for RT-11 V5.4 are located in Section A.1. Error messages that were new for RT-11 V5.3 are located in Section A.2.

1.5.11 New Software Update Process

To maintain RT-11 and layered product software, you no longer need to install mandatory patches. Instead, RT-11 V5 provides a new update process that replaces the software modules that have been changed. Update kits are distributed periodically after RT-11 is released. Each update kit includes an automatic procedure for replacing software modules with the replacement software modules. The update procedure guides you through an interactive dialog to determine which products you want to update and which devices you are using. Then the update software performs the module replacement operations.

Instead of providing binary patches, the RT-11 Software Dispatch now contains only articles describing problems corrected and functionality added by the update kits.

Source kits are still available.

1.5.12 New Unsupported Utilities

A new unsupported debugging package for MACRO-11 programs, DBG-11, has been added to the RT-11 distribution kit. See the new DBG-11 Symbolic Debugger User's Guide for complete information.

An unsupported utility, TRANSFER, has been added to the RT-11 distribution kit. See Chapter 5 for complete information.

See the RT-11 System Utilities Manual for a description of other unsupported utilities and instructions on how to use them. See the Preface to this manual for a definition of the meaning of the term unsupported as it applies to unsupported utilities.

1.6 NEW DOCUMENTATION

Two RT-11 V4 manuals, the RT-11 System User's Guide and the RT-11 Installation and System Generation Guide, have each been split into two manuals to achieve a more modular documentation set. The RT-11 V5 documentation set also includes three new manuals, as well as additions to V4 manuals and changes in the documentation format. The following sections describe the new and changed V5 manuals.

1.6.1 DBG-11 Symbolic Debugger User's Guide

The DBG-11 Symbolic Debugger User's Guide tells you how to use the DBG-11 symbolic debugging package with RT-11 to debug your MACRO-11 programs.

1.6.2 RT-11 for Beginners

This new manual, RT-11 for Beginners, assumes that you do not have any experience with computers. The manual tells you how to start the RT-11 operating system, how to use a subset of RT-11 commands to perform common operations, and how to use the keypad editor to create and edit text files. A quick reference to a subset of RT-11 commands and options is included. The manual also lists a subset of RT-11 system messages, a brief explanation of each message, and suggestions for correcting errors.

1.6.3 RT-11 Automatic Installation Booklets

The RT-11 automatic installation booklets are new to the RT-11 documentation set. Each booklet provides instructions for starting the software that automatically installs RT-11. Your RT-11 documentation includes the booklet appropriate for your distribution media, if your media supports automatic distribution:

RT-11 Automatic Installation Booklet, TK50 Magtape

RT-11 Automatic Installation Booklet, RX02 Diskettes

RT-11 Automatic Installation Booklet, RL02 Disk

RT-11 Automatic Installation Booklet, MicroPDP-11 (RX50 distribution)

RT-11 Automatic Installation Booklet, Professional 300 Series (RX50 distribution)

1.6.4 RT-11 Mini-Reference Manual

This new RT-11 manual provides condensed reference information in a small, portable loose-leaf binder. The RT-11 Mini-Reference Manual covers keyboard commands, utility programs, programmed requests, SYSLIB routines, and monitor fixed offsets. This manual replaces the RT-11 V4 RT-11 Pocket Guide and contains information from the programmed request section of the RT-11 Programmer's Reference Manual as well as information from the RT-11 Pocket Guide.

1.6.5 RT-11 Update User's Guide

This manual, which you receive with your update kit, describes how to use the new RT-11 update process to maintain your software. You can store your RT-11 Update User's Guide in Volume 1 of your RT-11 binder set.

1.6.6 Guide to RT-11 Documentation

This manual, which replaces the V4 RT-11 Documentation Directory, summarizes each manual in the RT-11 documentation set and suggests appropriate reading paths for different users.

1.6.7 RT-11 System User's Guide

The V5 RT-11 System User's Guide describes the RT-11 operating system, system conventions, keyboard monitor commands, and the text editor EDIT. This manual contains the information in Chapters 1 through 5 of the V4 RT-11 System User's Guide and provides a new chapter on the indirect control file processor (IND).

1.6.8 RT-11 System Utilities Manual

This new manual, which describes the RT-11 utility programs, presents the information from Chapters 6 through 24 of the V4 RT-11 System User's Guide. This manual also includes chapters on the new backup utility program (BUP) and logical disk subsetting program (LD).

1.6.9 RT-11 Installation Guide

The RT-11 Installation Guide describes manual procedures for installing RT-11; it also tells you how to customize the distributed RT-11 software. This document contains the information in Chapters 1 through 7 of the V4 RT-11 Installation and System Generation Guide. Two chapters have been added to describe installing RT-11 on diskettes from a hard disk distribution kit, and installing RT-11 on a MicroPDP-11. Information has also been added to Chapter 5 to describe installing RT-11 using RC25 disks.

1.6.10 RT-11 System Generation Guide

This manual, taken from Chapters 8 through 10 of the V4 RT-11 Installation and System Generation Guide, tells you how to run the system generation software to produce monitors and handlers with specialized configurations and characteristics.

1.6.11 PDP-11 TECO User's Guide

Since TECO software has been removed from the RT-11 distribution kit, the PDP-11 TECO User's Guide has been removed from the documentation set.

CHAPTER 2

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

This chapter describes features that have been added to existing RT-11 software components and features that have been changed.

2.1 DISTRIBUTION KIT

Changes have been made to the RT-11 software distribution kit for V5.4.

Sources for all supported and unsupported components except MACRO, CREF, the DBG-11 handlers, DBGSYM.SAV, and GIDIS are provided on the RT-11 distribution source kit. The distributed files TRANSF.EXE, TRANSF.TSK, and KED.SAV (including its variants) cannot be built from the source files, using only RT-11.

The KED, KEX, and K52 sources are now provided on the source distribution kit.

Software kit maps are no longer included in this document; they are now included with the binary distribution kit cover letter.

2.1.1 New Files

The following files have been added to the RT-11 V5.4 software distribution kit:

DBGSYM.SAV	DBG-11 symbol definition utility (unsupported)
MDUP.AI	Bootstrap utility for TMSCP magtape automatic installation
MDUP.MU	Bootstrap utility for TMSCP magtape manual installation
MSB.COM	Indirect command file used to build distribution backup magtape from distributed 1600 bits/inch magtape
MU.SYS	TMSCP handler for SJ and FB monitors
MUB.COM	Indirect command file used to build distribution backup magtape from distributed TK50 magtape
MUX.SYS	TMSCP handler for XM monitor
NU.MAC	DEUNA handler source file for system generation

NUX.SYS	DEUNA handler for XM monitor
SDH.SYS	DBG-11 hardware I/O handler for SJ and FB monitors (unsupported)
SDHX.SYS	DBG-11 hardware I/O handler for XM monitor (unsupported)
SDS.SYS	DBG-11 software I/O handler for SJ and FB monitors (unsupported)
SDSX.SYS	DBG-11 software I/O handler for XM monitor (unsupported)
TU.MAC	TMSCP handler source file for system generation
UM.MAC	MSCP class handler source code file required for TMSCP device handler system generation

2.2 MONITORS

The following features are additions or changes to the RT-11 monitors.

- **Abort code passed from RMON to the handler abort entry point has been modified for RT-11 V5.4 -** Job channel abort information is now passed to the handler in R5. Therefore, you should modify any internally queued handler you have written to correctly process the information passed by R5 in the following manner:

  The contents of R5 now determine the type of abort; whether a handler aborts all queue elements for a job or only those queue elements on a particular channel. When R5 contains zero, the handler should abort all queue elements for the job number matching R4. When R5 is nonzero, it points to the first word of a channel control block (the channel status word), and the handler should abort only the queue elements for that channel.

- **KMON passes unaltered commands -** KMON by default now passes unaltered commands by means of the chain area. This is useful for programs that require command input that is not a file specification (such as SETUP commands). Only RUN filename constructions are valid; that is, RUN filename, RUN filename data, and CCL (concise command language) constructions.

  A new optional third argument for the GTLIN subroutine supports a program getting the unaltered command line. See the RT-11 Programmer's Reference Manual for information.

- **New KMON preprocessor interface (UCF) -** You can create a KMON preprocessor (UCF) that intercepts all command line input after KMON has tried IND command syntax. UCF can be used to process valid DCL commands in a way you choose, rather than the way KMON would normally process the commands. UCF can also be used to process commands that KMON would otherwise refuse. See the RT-11 Software Support Manual for information.

  The distributed file UCL.SAV can be copied and renamed to create a functional UCF.SAV. See the RT-11 System User's Guide for information.

- **New KMON command line processor customization patch** - A new customization patch lets you selectively inhibit the various (DCL, CCL, UCL) KMON command line processors. See Section 2.7.63 in the RT-11 Installation Guide for information.

- **KMON no longer automatically converts lowercase input to uppercase** - KMON no longer automatically converts lowercase characters to uppercase from both terminal input and command files. A program using .GTLIN with the lowercase bit set receives the commands in lowercase if entered in lowercase. However, when type-ahead is used, characters are sometimes transmitted and echoed as uppercase no matter what case is entered at the keyboard.

  See Chapter 2 in the RT-11 Software Support Manual for information.

- **Minimum memory requirement** - The minimum memory requirement has been increased to 16K words for the SJ monitor and to 24K words for the FB monitor. The minimum memory requirement for the XM monitor (32K words) remains the same as in RT-11 V4.

- **New default working monitor** - FB is the default monitor for a working system installed through RT-11 automatic installation procedures for all processors except the Professional 300 series. The XM monitor is installed on the Professional 300 series during automatic installation. If your system does not include the minimum hardware requirements for the FB monitor (24K words of memory and a line-time or Professional series clock), you cannot perform the automatic installation procedure. See Chapter 4 for information on how to install your system when your hardware configuration does not support automatic installation.

- **Distributed XM monitor** - Under RT-11 V4, use of the extended memory feature was available only through system generation. The RT-11 V5 distribution kit includes an XM monitor, RT11XM.SYS. The RT-11 Installation Guide provides a description of all the features included in the distributed XM monitor. The most notable features include device timeout support and system job support.

- **BATCH excluded from distributed monitors** - BATCH support is excluded from the distributed monitors. BATCH support is available only through system generation.

- **FPU support included in distributed monitors** - FPU support is included in the distributed SJ, FB, and XM monitors.

- **New monitor fixed offsets** - The following monitor fixed offsets have been added to the resident monitor data base:

  SPSTAT — Word offset 414; SPOOL status word (formerly reserved for DECnet)

  EXTIND — Byte offset 416; the stored error byte for IND

  INDSTA — Byte offset 417; the IND control status byte

  $MEMSZ — Word offset 420; contains the total amount of memory available (in 32-word blocks) to the monitor currently executing

$TCFIG Word offset 424; contains the address of the current console configuration word (SET option status); points to TTCNFG in single terminal systems and T.CNFG in the current console's terminal control block (TCB) in multiterminal systems

$INDDV Word offset 426; pointer to INDDEV (the ASCII device name and unit number from which IND will be run)

MEMPTR Word offset 430; contains the offset to memory control block pointers

P1EXT Word offset 432; contains the address of the kernel PAR1 externalization routine for fetchable handlers under the XM monitor; contains the value 0 when running under the SJ or FB monitor

Word offset 444 is reserved

IMPLOC Word offset 446; under the SJ monitor, IMPLOC contains a value of 0. Under the FB and XM monitors, IMPLOC contains a pointer to word following last word in table of pointers to the impure area

Under FB and XM monitors, IMPLOC-2 contains a pointer to the foreground job impure area if a foreground job is loaded, or a 0 if no foreground job is loaded

The last word in the table of pointers to impure areas contains -1. The word above the word containing -1 is the pointer to the background job impure area. The pointers to the impure area for any system jobs supported by the monitor are located between the pointers to the foreground and background impure areas

KMONIN Word offset 450; indicates if KMON is the background job. If KMONIN contains nonzero value, then KMON is background job; if KMONIN contains zero, then KMON not background job

PROGDF Byte offset 452; indicates default editor for EDIT command. The values in PROGDF that indicate the default editor are:

Value	Default Editor is:
222	KED
223	K52
224	KEX
225	EDIT
226	TECO

Byte offset 453 is reserved

WILDDF Byte offset 454; indicates if wildcards (*) are implicit or must be supplied explicitly. A value of one indicates implicit wildcarding; zero indicates explicit wildcarding

Byte offset 455 is reserved

See Section 3.6.1 in the RT-11 Software Support Manual for more information on monitor fixed offsets.

- **New monitor floating offsets** - The following monitor floating offsets have been added to the resident monitor data base:

  CLITYP — High byte of word located one word below INDDEV (see $INDDV word offset 426 in this section); indicates which command processor is running

  CLIFLG — Low byte of word located one word below INDDEV (see $INDDV word offset 426 in this section); indicates which command processors can be run

- **New bit definitions** - The following new bit masks are defined for fixed offset locations:

Offset	Bit Mask	Meaning
CONFG2	LDREL$ = 20	A handler has been unloaded or released.
	BUS$ = 100	0 = RT-11 is running on UNIBUS or Professional (CTI-BUS) machine. If 20000 = 0, then UNIBUS machine. If 20000 = 1, then CTI-BUS machine. 1 = RT-11 is running on Q-BUS machine.
	PROS$ = 20000	0 = RT-11 is not running on Professional 300 series system. 1 = RT-11 is running on Professional 300 series system.
INDSTA	CC$IND = 4	Status of .ENABLE/.DISABLE ABORT.
	CC$GBL = 10	Status of global .SCCA.
	LN$IND = 40	Indicates current line from IND.
	IN$RUN = 100	KMON has issued a RUN IND command.
	IN$IND = 200	IND has returned control to KMON.
SYSGEN	FPU11$ = 400	Indicates that FPU support has been chosen as a system generation option.

	TSXP$	= 100000	Reserved for TSX PLUS*. This bit should never be set under RT-11.
			*TSX-PLUS is a trademark of S & H Computer Systems.
SPSTAT	NEXT	= 10	Move to start of next file.
	OFF	= 20	Set spooler unit off.
	ON	= 40	Set spooler unit on.
	KILL	= 100	Remove spooled output from work file.
	ACTIVE	= 200	Indicates spooler is active.
	SHOW	= 4000	Display spooler status.
	PRTSCR	= 10000	Print screen (Professional 300 series only).
	DATIME	= 20000	Date and time request (for flag pages).
	INTEN	= 40000	Fake interrupt enable.
	ERROR	= 100000	Error bit (set by SPOOL).

See Section 3.6 in the RT-11 Software Support Manual for more information on fixed offset bit masks.

- **Support for 22-bit addressing (on Q-BUS and CTI-bus processors only)** - The XM monitor now supports 22-bit addressing to allow each job (up to eight with system job support) to have a program logical address space (PLAS) limited only by available memory, using virtual overlays and/or virtual .SETTOP. The job PLAS may be up to 4M bytes (minus 64K bytes RT-11 overhead), using explicit programmed requests.

  All monitors also support up to 4M bytes of memory through the VM handler, which treats memory above 28K words as though it were a random access device.

  When using the XM monitor on any 22-bit system, 22-bit addressing will be enabled. However, since RT-11 does not support the UNIBUS map hardware, memory above 256K bytes cannot be used for direct memory access (DMA) I/O on UNIBUS processors. Any attempt to do so will return a hard error from the device handler. To avoid this situation, the VM handler may be installed such that its base is at the 18-bit/22-bit boundary (SET VM BASE=10000). No such restrictions apply to Q-BUS systems.

  See Chapter 4 of the RT-11 Software Support Manual for more information on 22-bit addressing.

- **KMON size** - The size of the keyboard monitor (KMON) for SJ has been increased to 20000(octal) bytes. In RT-11 V4, KMON was 17000(octal) bytes.

2.3 KEYBOARD MONITOR COMMANDS

This section summarizes the changes to existing keyboard monitor commands and describes the new keyboard monitor commands. For details on these changes and new features, see Chapter 4 in the RT-11 System User's Guide.

2.3.1 Changed Keyboard Monitor Commands and Options

The following keyboard monitor commands and options have changed.

2.3.1.1 Commands

BASIC

BASIC has been removed from the list of keyboard commands. To run BASIC, you must now type the entire word BASIC (rather than the abbreviation BAS), then a carriage return:

```
.BASIC<RET>
```

COMPILE

The /BUFFERING option has been eliminated.

The /PASS:1 option has been eliminated.

COPY

When you use the /DELETE option with the COPY command, you are no longer prompted for confirmation of the deletion. If you want this prompt, you must now use the /QUERY option.

When you copy files, the protection status of the output file will be the same as the protection status of the input file, unless you use the /PROTECTION or /NOPROTECTION option.

The /SETDATE option now accepts an optional date argument. When you specify /SETDATE[:date], the system puts the specified date on all files you copy.

You no longer need to use the /SYSTEM option to copy .SYS files unless you use wildcards in the input file type.

You can now use the /VERIFY option for files as well as for entire volumes. The /VERIFY option is invalid with the /ASCII and /BINARY options.

You can now use the /WAIT option with the /DOS and /INTERCHANGE options.

DELETE

The DELETE command no longer prompts you for confirmation, unless you use wildcards in the file specification.

You no longer need to use the /SYSTEM option to delete .SYS files unless you use wildcards in the input file type.

You can now use the /WAIT option with the /DOS and /INTERCHANGE options.

DIBOL — The /BUFFERING option has been eliminated.

DIFFERENCES — The DIFFERENCES command now accepts wildcards to let you compare several files with one command. See Chapter 4 of the RT-11 System User's Guide for more details on using wildcards with the DIFFERENCES command.

The /SLP option now accepts a file specification argument. When you specify /SLP:filespec, you can later use the resulting file you specify as input to SLP. You can also use the /OUTPUT:filespec option in the same command line to produce a differences listing and an SLP command file simultaneously. If you use the /OUTPUT:filespec option without the /SLP:filespec option, a differences listing is generated and the SLP command file is printed on the console. In RT-11 V4, you had to use /OUTPUT:filespec with /SLP to specify a command file; you could not produce a differences listing and a command file simultaneously.

DIRECTORY — The /BEGIN option is now position independent. The command DIRECTORY/BEGIN is now valid.

You can now use the /WAIT option with the /DOS and /INTERCHANGE options.

You can also now use the /VOLUMEID[:ONLY] option with /INTERCHANGE to print the volume ID of an interchange diskette.

EXECUTE — The /BUFFERING option has been eliminated.

The /PASS:1 option has been eliminated.

FORMAT — You can now format volumes while a foreground job is loaded or when the volume to be formatted contains protected files. If you try such an operation, the system gives you a warning message, then asks you whether you want to continue the operation.

The table of verification patterns valid for the /PATTERN option has been increased to 16 patterns. The last 4 of the 16 patterns are reserved for future use.

INITIALIZE — You can now use the /VOLUMEID[:ONLY] option with the /INTERCHANGE option to write a volume identification on an interchange diskette.

MACRO — The /PASS:1 option has been eliminated.

PRINT

The PRINT command is affected when both QUEUE and SPOOL are running. KMON assigns precedence to SPOOL for any PRINT command, so take care if you run both QUEUE and SPOOL. PRINT options /PROMPT and /NAME are specific only to QUEUE. If both QUEUE and SPOOL are running, KMON treats those PRINT options as assigned to SPOOL, and returns an invalid option error.

When SPOOL or both SPOOL and QUEUE are running, the /FLAGPAGE:n option, when a value is specified for n, overrides the SET SP FLAG=n command. When no value is specified for n with the /FLAGPAGE:n option, the value for n is set by the SET SP FLAG=n command. The default /NOFLAGPAGE option inhibits flag pages under all circumstances.

When only the QUEUE package is running, the default number of banner pages printed when you use the /FLAGPAGE:n option is determined by the default number of banner pages set with the QUEMAN /P option. If the default set with the /P option is 0, the default for /FLAGPAGE:n is 1. If the QUEMAN /P option is not used, the default is /NOFLAGPAGE.

RENAME

The /SETDATE option now accepts an optional date argument. When you specify /SETDATE[:date], the system puts the specified date on all files you rename.

You no longer need to use the /SYSTEM option to rename .SYS files unless you use wildcards in the input file type.

REMOVE

When running under the XM monitor only, the REMOVE command can now be used to remove a global region from extended memory and return the memory allocated to that global region to the free memory list.

RESET

The RESET command now resets the console terminal ring buffers and command buffers.

RUN

You can now execute virtual jobs from devices other than the system device. Therefore, you can use the RUN command to execute virtual jobs.

SET

SET command conditions are now individually parsed. As a result, you no longer need to enter a completely valid command for modifications to occur. For example, if a SET command specifies four condition modifications and the third condition is invalid, the first two conditions are modified and an error is returned when the third condition is parsed. All conditions following the invalid condition are not modified. Unless you know which condition is invalid, you should reenter the entire command to be sure all conditions are modified.

The SCOPE option for TT is now the default for all monitors.

The DUn [NO]WRITE options are eliminated.

The DZ [NO]WRITE options are eliminated because RX50 diskettes provide hardware write protection.

You can now use SET TERM or SET TT to set console characteristics.

SETUP

The SETUP [NO]CURSOR command now applies to VT200 series terminals.

SHOW

SHOW DEVICES:DU displays new DU status information. The port display now indicates an invalid port by displaying an asterisk (*) before the port number. For example, port = *n, where n = 0,1,2,3, indicates that port n was not installed. If SHOW DEVICES:DU displays port = n, where n = 0,1,2,3, then port n is installed (that is as documented).

The SHOW command can now list the contents of the UCL data file, using the new SHOW COMMANDS command. You can send output from SHOW COMMANDS to any RT-11 output device. The /PRINTER option sends output to the printer. The /OUTPUT:filespec options sends output to the device filespec (where filespec is the device name). The options are position dependent and must follow COMMANDS. For example, the command SHOW COMMANDS/PRINTER sends the contents of the UCL data file to your printer. The default output of SHOW COMMANDS is the console terminal.

/PRINTER and /OUTPUT:filespec options are position dependent in the same manner when used with the SHOW ERRORS command.

The shortest valid abbreviation for COMMANDS is COM.

The SHOW command now includes logical disk subsetting assignments.

The SHOW ALL command now displays the organization of physical memory and logical disk subsetting assignments.

The SHOW ALL command now displays the region type in the extended memory map.

The SHOW CONFIGURATION command now also displays the following system attributes:

- Total amount of memory
- Active command file processor: KMON or IND
- SL status: on or off
- Default editor for EDIT command
- Status of .SCCA support and the .SCCA flag (enabled or disabled)
- EAE (extended arithmetic element) hardware option

The shortest valid abbreviation for CONFIGURATION is CON. Any indirect files that contain a shorter abbreviation must be updated.

The SHOW DEVICES command lets you obtain information about a specific device by using the command SHOW DEVICES:xx. The variable xx represents the two-letter permanent device name. The CSR and vectors for each device displayed are also given.

The SHOW MEMORY command shows the location of each low memory component and, under the XM monitor, each extended memory region as well.

SHOW MEMORY now displays a new fourth column in the extended memory map that indicates the type of region being listed.

The SHOW ERRORS command displays errors recorded by the error logger while running under the SJ monitor.

The SHOW QUEUE command shows the contents of the queue for SPOOL or QUEUE, or for both if both are running. The SPOOL status report shows whether the SPOOL output device is active or inactive, the number of blocks spooled for output, and the number of free blocks in SPOOL's work file.

The SHOW QUEUE command is now performed by the RESORC /Q option, rather than by the QUEMAN /L option. However, the QUEMAN /L option is still valid for compatibility. If the SPOOL utility is running, SHOW QUEUE requires RESORC.SAV be on device SY:. If the QUEUE utility is running, SHOW QUEUE requires RESORC.SAV and QUEMAN.SAV be on device SY:.

SRUN

The default file type for the SRUN command is .REL. The SRUN command now defaults to the system device (SY:).

TYPE

Executing a CTRL/O now stops the display of a file when multiple files are displayed at the terminal. Executing a CTRL/O discards output until another CTRL/O is executed, or until the beginning of the next file is reached, whereupon output to the terminal is resumed.

2.3.1.2 /WAIT Options

You can now abort a /WAIT operation. Refer to the following commands in the RT-11 System User's Guide for more information.

BOOT
COPY
DELETE
DIRECTORY
FORMAT
INITIALIZE
PRINT
PROTECT
RENAME
SQUEEZE
TYPE
UNPROTECT

2.3.2 New Keyboard Monitor Commands

The following keyboard monitor commands are new. Options for these commands are listed in the RT-11 System User's Guide.

ABORT	The ABORT command lets you abort, from the shared console, a foreground or system job assigned to a private console terminal with the FRUN or SRUN /TERMINAL:n option. The ABORT command cannot abort a job with SCCA in effect.
BACKUP	The BACKUP command provides a quick means of backing up a file or an entire volume for storage.
DISMOUNT	The DISMOUNT command is used for logical disk subsetting to dissociate a logical disk unit from the file to which it was assigned.
MOUNT	The MOUNT command is used to associate a file with a logical disk unit for logical disk subsetting.
PROTECT	The PROTECT command assigns a protection status that prevents deletion of a file until you remove the protection.
UNPROTECT	The UNPROTECT command removes protection from a file so you can delete it.

2.3.3 New or Changed Keyboard Monitor Command Options

This section contains new or changed options for old and new keyboard monitor commands. Complete descriptions for the options not described here are located in the RT-11 System User's Guide.

BACKUP

/DEVICE
/FILE
/RESTORE
/VERIFY

COMPILE

/LOG
/PAGE:n
/TABLES

COPY

/BEFORE[:date]
/DATE[:date]
/INFORMATION
/MULTIVOLUME
/PROTECTION
/NOPROTECTION
/RETAIN
/SINCE[:date]

DELETE

/BEFORE[:date]
/DATE[:date]
/INFORMATION
/SINCE[:date]

DIBOL

/LOG
/PAGE:n
/TABLES

DIFFERENCES

/DEVICE

DIRECTORY

/BACKUP
/PROTECTION
/NOPROTECTION

EDIT

/KEX

EXECUTE

/DUPLICATE
/GLOBAL
/LOG
/PAGE:n
/TABLES

INITIALIZE

/BACKUP

LINK

/DUPLICATE
/GLOBAL
/LIMIT:n

MOUNT

/WRITE
/NOWRITE

PRINT

/BEFORE[:date]
/DATE[:date]
/INFORMATION
/SINCE[:date]

PROTECT

/BEFORE[:date]
/DATE[:date]
/EXCLUDE
/INFORMATION
/LOG
/NOLOG
/NEWFILES
/QUERY
/SINCE[:date]
/SYSTEM
/WAIT

RENAME

/BEFORE[:date]
/DATE[:date]
/INFORMATION
/SINCE[:date]

SET

dd CSR=n
RETRY=n
SUCCES
NOSUCCES
VECTOR=n

DU CSR0=n — Modifies the DU handler to use n as the CSR address for controller 0. Equivalent to SET DU CSR=n.

DU VEC0=n	Modifies the DU handler to use n as the vector address for controller 0. Equivalent to SET DU VECTOR=n.
DU CSR1=n	Modifies the DU handler to use n as the CSR address for controller 1.
DU VEC1=n	Modifies the DU handler to use n as the vector address for controller 1.
DU CSR2=n	Modifies the DU handler to use n as the CSR address for controller 2.
DU VEC2=n	Modifies the DU handler to use n as the vector address for controller 2.
DU CSR3=n	Modifies the DU handler to use n as the CSR address for controller 3.
DU VEC3=n	Modifies the DU handler to use n as the vector address for controller 3.

DUn PART=x
PORT=x
UNIT=x

DW WCHECK
NOWCHECK
WRITE
NOWRITE

DXn WRITE
NOWRITE

DYn WRITE
NOWRITE

EDIT KEX

EL LOG
NOLOG
PURGE

EXIT SWAP
NOSWAP

KMON IND
NOIND

LD EMPTY

LDn CLEAN
WRITE
NOWRITE

LP BIT8
NOBIT8
ENDPAG=0
ENDPAG=n

LS BIT8
NOBIT8
ENDPAG=0
ENDPAG=n

LS GRAPH — Sets the LS handler to output all characters regardless of the width of the line. Use this command for plotters such as the LVP-16 or any other graphics devices that use the LS handler. When you SET LS GRAPH, any width limit you specified using the SET LS WIDTH command is ignored.

LS NOGRAPH — Sets the LS handler to ignore all characters that do not fit on a line. Excess characters are not printed. The length of the line is determined by SET LS WIDTH=nnnnn. SET LS NOGRAPH is the default setting.

LS SPEED=n

MS RETRY=n — Allows you to change the number of times the MS handler attempts to recover from an error when the Error Logger is running.

MS CSR=n — Modifies the MS handler to use n as the CSR address for controller 1.

MS VECTOR=n — Modifies the MS handler to use n as the vector address for controller 1.

MS CSR2=n — Modifies the MS handler to use n as the CSR address for controller 2.

MS VEC2=n — Modifies the MS handler to use n as the vector address for controller 2.

MS CSR3=n — Modifies the MS handler to use n as the CSR address for controller 3.

MS VEC3=n — Modifies the MS handler to use n as the vector address for controller 3.

MS CSR4=n — Modifies the MS handler to use n as the CSR address for controller 4.

MS VEC4=n — Modifies the MS handler to use n as the vector address for controller 4.

MS CSR5=n — Modifies the MS handler to use n as the CSR address for controller 5.

MS VEC5=n — Modifies the MS handler to use n as the vector address for controller 5.

MS CSR6=n — Modifies the MS handler to use n as the CSR address for controller 6.

MS VEC6=n — Modifies the MS handler to use n as the vector address for controller 6.

MS CSR7=n	Modifies the MS handler to use n as the CSR address for controller 7.
MS VEC7=n	Modifies the MS handler to use n as the vector address for controller 7.
MS CSR8=n	Modifies the MS handler to use n as the CSR address for controller 8.
MS VEC8=n	Modifies the MS handler to use n as the vector address for controller 8.
MU RETRY=n	Modifies the MU handler to use n as the number of times MU attempts to recover from an error. The value n must be an integer in the range 1 through 8. The default value for n is 8.
MU [NO]SUCCES	Modifies the MU handler to [not] log successful I/O transfers as well as errors when the Error Logger is running. SUCCES is the default mode.
MU CSR0=n	Modifies the MU handler to use n as the CSR address for controller 0. Equivalent to SET MU CSR=n.
MU VEC0=n	Modifies the MU handler to use n as the vector address for controller 0. Equivalent to SET MU VECTOR=n.
MU CSR1=n	Modifies the MU handler to use n as the CSR address for controller 1.
MU VEC1=n	Modifies the MU handler to use n as the vector address for controller 1.
MU CSR2=n	Modifies the MU handler to use n as the CSR address for controller 2.
MU VEC2=n	Modifies the MU handler to use n as the vector address for controller 2.
MU CSR3=n	Modifies the MU handler to use n as the CSR address for controller 3.
MU VEC3=n	Modifies the MU handler to use n as the vector address for controller 3.
NC SHOW	
NQ CSR=n	Modifies the NQ handler to use n as the CSR address.
NQ SHOW	
NQ VECTOR=n	Modifies the NQ handler to use n as the vector address.
NU CSR=n	Modifies the NU handler to use n as the CSR address.
NU SHOW	Displays the station physical address for the DEUNA controller.

NU VECTOR=n	Modifies the NU handler to use n as the vector address.

SL ASK
KMON
LEARN
NOLEARN
LET
NOLET
OFF
ON
SYSGEN
TTYIN
NOTTYIN
VT52
VT62
VT100
VT101
VT102
WIDTH=n

SP ENDPAG=0
ENDPAG=n
FLAG=n
FORM0
NOFORM0
KILL
NEXT
WAIT
NOWAIT
WIDE
NOWIDE

VM BASE=nnnnnn

VM SIZE=nnnnnn	Lets you select the memory size, in blocks (512 bytes for each block), of the virtual device. If you SET VM SIZE=0, VM allocates all available memory from the SET VM BASE value to the top of physical memory. VM SIZE=0 is the default.

XC SPEED=n

SETUP

480INTERLACE

COLOR
MONO

COMPOSE
NOCOMPOSE

DATA
TYPE

LANDSCAPE
PORTRAIT

```
  RETAIN
  SETCOLOR color [pcolor:value,pcolor:value,pcolor:value]
  SETCOLOR color FACTORY

SHOW
  COMMANDS
  /OUTPUT:filespec
  /PRINTER
  MEMORY
  SUBSET

TYPE
  /BEFORE[:date]
  /DATE[:date]
  /INFORMATION
  /SINCE[:date]

UNPROTECT
  /BEFORE[:date]
  /DATE[:date]
  /EXCLUDE
  /INFORMATION
  /LOG
  /NOLOG
  /NEWFILES
  /QUERY
  /SINCE[:date]
  /SYSTEM
  /WAIT
```

2.4 DEVICE CODES

The following device codes have been assigned.

Name	Code	Device
MU	60	TMSCP magtape class handler
NC/NQ/NU	61	Ethernet class handler
SD	62	DBG-11 handler
ST	63	DBG-11 symbol table handler

2.5 HANDLERS

The following handlers, distributed on the RT-11 V5 distribution kit, are no longer supported by RT-11 and are therefore no longer documented in the RT-11 documentation set. You can still use these handlers as documented in previous versions of RT-11.

CR	DT
CT	PC
DP	PD
DS	RF

Also, support for VT11 and VS60 graphics terminals is no longer included by default in the system monitor. You must perform system generation procedures to include that support.

XM versions of all supported handlers are now included on the RT-11 distribution kit.

All RT-11 handlers are now linked using the /NOBITMAP option.

The following sections describe the changes made to RT-11 handlers.

2.5.1 DD

The following changes have been made to the DD handler.

- The DD handler now operates on the SBC-11/21 and SBC-11/21 PLUS.
- The following SET commands are valid for the DD handler:

```
SET DD RETRY=n
       SUCCES
       NOSUCCES
```

  See the RT-11 System User's Guide for a description of those commands.

2.5.2 DL

The following changes have been made to the DL handler.

- The following SET commands are valid for the DL handler:

```
SET DL CSR=n
       RETRY=n
       SUCCES
       NOSUCCES
       VECTOR=n
```

  See the RT-11 System User's Guide for a description of those commands.

- The DL handler now maintains device size information in a unit-specific table. That feature reduces the number of controller operations required in a system with multiple DL units.

- The DL handler now reports write-lock and write-gate errors to the error logger.

- The DL handler supports 22-bit DMA with the RLV12 controller.

2.5.3 DM

The following changes have been made to the DM handler.

- The following SET commands are valid for the DM handler:

```
SET DM RETRY=n
       SUCCES
       NOSUCCES
```

  See the RT-11 System User's Guide for a description of those commands.

- The DM handler now reports the following errors to the error logger:

  Cylinder overflow

  Data late

  Transfer to or from nonexistent drive

  Transfer to or from memory address higher than existing memory

  Write-lock

2.5.4 DU

The following changes have been made to the DU handler.

2.5.4.1 **Changes to the DU Translation Tables** - Changes have been made in the structure of the DU handler translation table. The starting label, labels for elements of the table, and the size of the table have changed for RT-11 V5.4. All programs you write to access the information contained in the table should use the following information. All programs you have written should be changed to use the following information.

Previously, the DU translation table started at label UT.UNIT. UT.UNIT was the 1-word MSCP unit number for a particular RT-11 unit. Now, the DU translation table starts at label DU.ID. Label DU.ID is a 1-word table identification and contains the Radix-50 value for the characters DU.

DU.ID is followed by a 1-byte count of the number of entries in the table. That byte, DU.NUM, is the low-byte of the word DU.ID+2. For RT-11 V5.4, DU.NUM contains the value 10(octal). The high-byte of DU.ID+2 is reserved.

The structure of the rest of the table remains as before. However, the labels you should use to specify elements of the table have changed. The following is the structure of the table with the changed labels:

Offset	Label	Meaning
buf+0	DU.ID	Radix-50 value for characters DU.
buf+2	DU.NUM	Byte containing number of entries in table
buf+3		Reserved
buf+4	DU.ENT	4-byte entry consisting of:
	DU.UNI	Physical MSCP unit number
buf+6	DU.PAR	Byte containing partition number
buf+7	DU.POR	Byte containing MSCP port (controller) number
buf+10	DU.ENT	The next 4-byte entry.....

2.5.4.2 **Other Changes to the DU Handler**

- Some DU system conditionals used by SYSGEN have been changed for RT-11 V5.4. See Section 2.34.3.

- DU now has a status word containing information about the last operation performed by the handler. The status word is called STATU$, and for V5.4 is located at offset 16 from the base of DU. The offset of STATU$ may change with each release of RT-11. The low 5 bits of STATU$ contain the status information. All other bits are reserved.

Octal Value	Meaning when set
00	Success
01	Invalid command
02	Command aborted
03	Unit off line
04	Unit available
05	Medium format error
06	Write protected medium
07	Compare error
10	Data error
11	Host buffer access error
12	Controller error
13	Drive error

  Use ODT/VDT or the E keyboard command to examine the contents of STATU$. You will need to perform customization patch 2.7.32 located in the RT-11 Installation Guide to use the E command. Use the SHOW MEMORY command display to find the base of the DU handler and add the offset to that base.

- The low-memory size of the DU handler when run under the XM monitor has been reduced from approximately 550 words to approximately 114 words. Data and code that was previously resident in low memory has now been placed in a global region in extended memory. Using the same technique, bad-block replacement support adds only approximately 16 words to the low-memory portion of the DU handler. Approximately 1900 words required for bad-block replacement support are placed in extended memory.

- You can now specify an option during SYSGEN that lets you boot RT-11 from any DU port. If you do not specify DU multiport booting during SYSGEN, you can boot RT-11 from DU port 0 only. Use the following procedure to enable multiport DU booting:

  1. Perform a SYSGEN and request support for multiple DU ports and multiple DU port booting. (See Section 2.34.1 for a reproduction of the SYSGEN dialog for DU multiport booting.)

     The distribution kit provides DU.SYS and DUX.SYS. After SYSGEN, you will have two additional files: DU.SYG and DUX.SYG.

  2. Rename DU.SYS to DU.DIS, and rename DUX.SYS to DUX.DIS. This step saves the original distributed files. Rename DU.SYG to DU.SYS, and rename DUX.SYG to DUX.SYS.

3. Use the SET DUn commands to map the particular DU device to the MSCP unit, port, and partition numbers. For example:

```
.SET DU3 UNIT=0, PORT=1
.SET DU4 UNIT=1, PORT=1
.SET DU5 UNIT=2, PORT=1
```

For the SET commands to take effect, you must UNLOAD and then LOAD the handler if it is a data device, or reboot it if it is a system device.

See Section 10.11 of the RT-11 Software Support Manual for information on mapping an MSCP disk.

4. Copy the resulting DU handler to the port on the DU devices you want to be able to boot. For example:

```
.COPY DUX.SYS DU3:
```

5. To hard-boot the DU unit on a new port, copy the bootstrap using the COPY/BOOT command to the desired DU port. The DU unit on that port will also support the soft-boot BOOT DUn: command.

- DU has had support added for .SPFUN and ISPFN 376 and 377. For DU, .SPFUN/ISPFN 376 performs a write to the specified sector, and .SPFUN/ISPFN 377 performs a read from the specified sector. Those writes and reads are not absolute; bad-block replacement and block vectoring remain in force.

  .SPFUN/ISPFN 376 and 377 are especially useful because they can return status information in the first word of the return buffer. Status information includes any occurrence of a bad-block error, forced error, or drive error. No notification of such errors is returned by a .WRITE or .READ request.

  The DU support for .SPFUN/ISPFN 376 and 377 is the same as for DM with the following exceptions:

  - DU supports an additional error code described in Sections 2.35 and 2.36.
  - For DU, bad-block replacement and block vectoring remain in force.

- The DU handler supports bad-block replacement. DU bad-block replacement is supported under the XM monitor only. Do not change from the XM monitor to another monitor, for example FB, when using a DU device with bad-block replacement support. You lose bad-block replacement support when running the FB monitor, which can result in the loss of data.

Bad-block replacement is a technique in which substitute blocks are provided for blocks that have caused a read or write error. The replacement blocks appear to occupy the disk positions of the original blocks, and the disk appears to contain only good blocks. All MSCP (DU) hard disk systems support bad-block replacement, performed either by the disk controller or as an optional feature of the DU handler selected during system generation. There is no bad-block replacement support for RX50 devices.

In MSCP systems that use an RQDX1, RQDX2, or RQDX3 controller, bad-block replacement is performed by the controller. In those systems, bad-block replacement is done automatically by the hardware and is transparent to RT-11.

In MSCP systems that use a KDA50, UDA50, KLESI-QA, or KLESI-UA controller, bad-block replacement can be performed by the DU handler. In those systems, bad-block replacement is an option that you select during system generation. If you specify bad-block replacement, the DU handler performs the replacements automatically.

NOTE

DIGITAL recommends that you perform a SYSGEN and request DU bad-block replacement support if you are using an RC25 or the RA80/RA81 series MSCP disks. Failing to provide DU bad-block replacement support for those devices can cause the loss of data or excessive bad blocks on those devices.

The following table lists the MSCP controllers and drives supported by RT-11 and indicates whether bad-block replacement is performed by the controller or the DU handler.

MSCP Bad-Block Replacement

MSCP Controller	Bad-Block Replaced by:	MSCP Drive
RQDX1	controller	RD31-RD51-RD52
RQDX2	controller	RD31-RD51-RD52-RD53
RQDX3	controller	RD31-RD51-RD52-RD53
KLESI-QA	handler	RC25-RCF25
KLESI-UA	handler	RC25-RCF25
UDA50	handler	RA80-RA81
KDA50	handler	RA80-RA81

Whether bad-block replacement is performed by the controller or the handler, it has the effect of making a disk appear to be error free. In certain cases, however, an I/O operation, a verification procedure, or a bad-block search may report the presence of bad blocks on a disk with replaced blocks. In such cases, any block identified as a bad block should be considered to be a good block with bad data. This means that the controller or handler provided a replacement block for a defective block but was unable to recover the data it contained.

- A significant loss of performance occurs when running DU under the SJ monitor.

- By-pass recovery for SPFUN 360 (and SPFUN 371) is now enabled when you specify 1 for the blk argument. Enabling by-pass recovery lets the DU handler determine if an error is recoverable and if so, retry the SPFUN 360 operation.

- For compatibility with the MU handler, DU supports the new SPFUN 360. For DU, SPFUN 360 is functionally equivalent to and is the replacement for SPFUN 371 (direct MSCP access). SPFUN 371 is supplied for compatibility purposes only and support for it could be removed in a future release of RT-11.

- If the wcnt argument in SPFUN 360 (special function bypass) is zero, the physical address specified in the command message is used. If wcnt is nonzero, it specifies the virtual address of the data buffer. That virtual address is converted to a physical address and placed in the command message.

- You can determine the status information for a DU (MSCP) device using the new IGTDUS system subroutine. MSCP device status information includes the unit name, the physical device size, the availability of the unit, whether it is removable, and whether it is write protected. See Chapter 1 for details on IGTDUS.

- You can now perform absolute (non-file-structured access) reads and writes to any MSCP device, using the new JREAD and JWRITE system subroutines. JREAD and JWRITE use a 28-bit starting block number. That lets you read and write to any block on any DU device. See Chapter 1 for details on JREAD and JWRITE.

- The following SET commands are changed for the DU handler:

```
SET DU CSR0=n
       VEC0=n
       CSR1=n
       VEC1=n
       CSR2=n
       VEC2=n
       CSR3=n
       VEC3=n
```

  See Section 2.3.3 for details.

- The following SET commands are valid for the DU handler:

```
SET DU RETRY=n
       SUCCES
       NOSUCCES

SET DUn PART=x
        PORT=x
        UNIT=x
```

  See the RT-11 System User's Guide for more information on those commands.

2.5.5 DW

The following changes have been made to the DW handler.

- Before RT-11 V5.3, the DW handler applied the wrong mapping of logical block numbers to physical block numbers on RD5x hard disks. The mapping was corrected in RT-11 V5.3. If you are upgrading your system from RT-11 V5.3 to V5.4, you have already reconfigured your DW hard disk and you do not need to be concerned with this change. If you are upgrading your system from a version of RT-11 previous to V5.3, be sure to read the cover letter included with your RT-11 V5.4 distribution.

  The DW mapping error caused file-structured operations (.READ/.WRITE requests) to map to different locations than non-file-structured operations (.SPFUN requests). File-structured and non-file-structured operations were corrected for RT-11 V5.3; however, see the ISPFN and .SPFUN information in Sections 2.35 and 2.36 for changes to the blk argument.

  Because DW mapping was changed for RT-11 V5.3, all RD5x hard disks that were used by a version of RT-11 before V5.3 must be reconfigured. That is done automatically by the Automatic Installation Procedure. However, any RD5x hard disk not in your processor during Automatic Installation must be initialized using RT-11 V5.4 before use. Copy any files you want to preserve to RX50 diskettes, using the version of RT-11 that wrote those files to the RD5x. Then, initialize the RD5x hard disk using RT-11 V5.4.

  Reconfiguration of two RD52 hard disks changes their usable size in blocks. The following table shows the changes:

Manufacturer	Size Before Reconfiguration	Size After Reconfiguration
CDC	Size is unchanged	
Atasi	55679. blocks	65535. blocks
Quantum	65535. blocks	65407. blocks

  The reduction of 128(decimal) blocks in usable size of the Quantum RD52 is caused by correction of an error in previous versions of RT-11. An area reserved by the disk was incorrectly allocated for data storage.

- DW uses the new .DRPTR macro to reduce its memory-resident size by moving device sizing and initialization code to non-memory-resident FETCH and LOAD code. That makes DW more than 150 words smaller (in memory).

- The following FORMAT commands are now valid for the DW handler:

```
FORMAT DW

FORMAT/VERIFY DW

FORMAT/VERIFY:only DW
```

- The following INITIALIZE command is now valid for the DW handler:

```
INITIALIZE/BAD DW
```

- The following SET commands are valid for the DW handler:

```
SET DW RETRY=n
       SUCCES
       NOSUCCES
       WCHECK
       NOWCHECK
       WRITE
       NOWRITE
```

See The RT-11 System User's Guide for more information on those commands.

2.5.6 DX

The following changes have been made to the DX handler.

The following SET commands are valid for the DX handler:

```
SET DX CSR=n
       RETRY=n
       SUCCES
       NOSUCCES
       VECTOR=n
       WRITE
       NOWRITE
```

See the RT-11 System User's Guide for more information on those commands.

2.5.7 DY

The following changes have been made to the DY handler.

- The following SET commands are valid for the DY handler:

```
SET DY CSR=n
       RETRY=n
       SUCCES
       NOSUCCES
       VECTOR=n
       WRITE
       NOWRITE
```

  See the RT-11 System User's Guide for more information on those commands.

- The DY handler supports only ECO Revision Level F and later controllers.

2.5.8 DZ

The following changes have been made to the DZ handler.

The following SET commands are valid for the DZ handler:

```
SET DZ RETRY=n
       SUCCES
       NOSUCCES
```

See the RT-11 System User's Guide for more information on those commands.

2.5.9 LD

The following changes have been made to the LD handler.

2.5.9.1 **Changes to the LD Translation Tables** - Changes have been made to the LD translation tables for RT-11 V5.4. All programs you write to access the information contained in those tables should use the following information. All programs you have written to access LD translation tables should be changed to use the following information.

The following characteristics of the LD translation tables have changed from those documented in Section 10.13.1 of the RT-11 Software Support Manual:

- Changes have been made to .SPFUN 372 for RT-11 V5.4. All programs you have written that use .SPFUN 372 for LD must be changed to reflect the following:

  The meaning of the wcnt argument is changed for RT-11 V5.4. The meaning for the values in the .SPFUN 372 wcnt argument for LD are now consistent with those for DU. The wcnt argument now takes the following values:

  ```
  wcnt = +1  read from handler to buffer
         -1  write from buffer to handler
  ```

  For RT-11 V5.4, the buf argument must be specified as 0.

- Previously, the LD translation tables started at label HANDLR. Now, the LD translation tables start at label LD.ID. Label LD.ID is a 1-word table identification and contains the Radix-50 value for the characters LD.

- LD.ID is followed by a 1-byte count of the number of entries in the table. That byte, LD.NUM, is the low-byte of the word LD.ID+2, and for RT-11 V5.4 always contains the value 10(octal). The high-byte of LD.ID+2 is reserved.

- There are four LD translation tables. That is unchanged. However, the names of labels you use to reference the tables have changed. Some table contents have also changed.

The following lists the label names, their offset location, and the contents of the four LD translation tables:

LD.FLG (LD.ID+4)

The table beginning at label LD.FLG is the table previously at the label HANDLR. LD.FLG contains one word for each LD unit number; that is, the value stored in LD.NUM. For RT-11 V5.4, there are always eight words in the LD.FLG table. The bits in each word of LD.FLG have the following meaning:

Bits	Label	Meaning
0-5	LD.NDX	An index to the handler tables in RMON for the physical device corresponding to the LD unit number.
6	LD.UNX	A flag that signals the index entry (bits 0-5) may be inaccurate and should be updated. LD sets LD.UNX for all units if, upon entry, the LDREL$ bit in RMON fixed offset CONFG2 is set.
7	LD.UOF	A flag that signals the entry in the LD.OFS table for that LD unit may be inaccurate. LD.UOF is set whenever a volume is squeezed. LD checks LD.UOF each time it uses an LD unit; if set, LD verifies that unit's LD.OFS table entry before proceeding.
8-13	LD.UNT	Contain the unit number of the physical disk assigned to the logical disk unit.
14	LD.RDO	Is the write-lock bit. If LD.RDO set, the LD unit is read-only.
15	LD.ACT	Is the allocation bit. If LD.ACT set, the LD unit is assigned. If LD.ACT clear, the LD unit is not assigned.

LD.OFS (LD.FLG+<2*Contents of LD.NUM>)

The second translation table starts at the label LD.OFS and contains one word for each LD unit number. The count of LD unit numbers is stored in LD.NUM. For RT-11 V5.4, the table starting at label LD.OFS contains eight words. Each word in LD.OFS contains the offset in blocks from the beginning of the assigned physical disk to the start of the area on that physical disk assigned to that LD unit number.

LD.SIZ (LD.FLG+<4*Contents of LD.NUM>)

The third translation table starts at the label LD.SIZ and contains one word for each LD unit number. The count of LD unit numbers is stored in LD.NUM. For RT-11 V5.4, the table starting at label LD.SIZ contains eight words. Each word in LD.SIZ contains the size in blocks of the area on the physical disk assigned to that logical disk unit.

LD.NAM (LD.FLG+<6*Contents of LD.NUM>)

The fourth translation table starts at the label LD.NAM and contains four words for each LD unit number. The count of LD unit numbers is stored in LD.NUM. For RT-11 V5.4, the table starting at label LD.NAM contains 32(decimal) words.

The first word of each 4-word entry contains the Radix-50 2-character name of the physical disk that is assigned to that logical disk unit. That Radix-50 word must be the physical (not logical) device name without any unit number. DL is a valid physical device name; DL1 and DK are not valid.

The second, third, and fourth words of each entry contain the Radix-50 file name and file type assigned as the logical disk.

2.5.9.2 Other Changes to the LD Handler

- Use the following customization patch if you want to change the suffix character for the LD handler to anything other than X or M. (You can use the X and M suffixes without performing a customization patch.) You must also use customization patch D.3 Changing the Handler File-Name Suffix, located in the RT-11 System Generation Guide. The following customization makes known to the monitor bootstrap the suffix character you specified in that customization; you must specify the same suffix character in both.

  In this customization, monitor.SYS is the name of the monitor file you want to modify, and ..BLDS is the value of that symbol from the monitor link map. xxxxxx represents a number that varies but is not important for you to know, and y represents the suffix character you want to add to the LD handler name.

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

     Base          Offset            Old          New?
      000000        ..BLDS          xxxxxx         ;Ay<RET>
      000000        ..BLDS+2        xxxxxx         <CTRL/Y><RET>
*<CTRL/C>
.
```

- The following new SET command is valid for the LD handler:

  ```
  SET LD EMPTY
  ```

  See the RT-11 System User's Guide for more information on that command.

2.5.10 LP

The following changes have been made to the LP handler.

- LP has been converted to a file-structured-device handler to support the new SET LP ENDPAG=n command. Issuing a directory operation (for example, the PRINT command) to LP while the printer is off line can cause the handler (and possibly the system) to appear hung. That is caused by the handler attempting to interrupt a nonexistent device. It also stops the USR from processing directory operations to any other device. Place the printer on line or type CTRL/C twice to free the handler.

- The following SET commands are now valid for the LP handler:

```
SET LP BIT8
       NOBIT8
       ENDPAG=0
       ENDPAG=n
```

  See the RT-11 System User's Guide for more information on those commands.

2.5.11 LS

The following changes have been made to the LS handler.

- The LS handler is now factory set for both Professional series processors (CSR=173400 and VECTOR=220) and traditional PDP-11 processors (CSR=176500 and VECTOR=300).

  The LS handler automatically sets the correct CSR and VECTOR when being run on a Professional processor; the CSR and VECTOR for Professional processors cannot be changed. When the LS handler is being run on a Professional processor, the fixed CSR and VECTOR are displayed by the SHOW DEV command.

  The CSR and VECTOR can be changed for traditional PDP-11 processors with the SET LS CSR=n and SET LS VECTOR=n commands. Use those commands to set the LS handler CSR and VECTOR if your serial line printer is installed at a nonstandard address. The CSR and VECTOR for traditional PDP-11 processors is displayed by the SHOW DEV command when the LS handler is being run on a traditional PDP-11.

  The SHOW DEV command also displays a "temporary" VECTOR (470, 474) for the LS handler on Professional and traditional PDP-11 processors. RT-11 uses that VECTOR while determining the type of processor on which LS is running.

- RT-11 includes support for the LVP-16 plotter printer with the new SET LS [NO]GRAPH command. See Section 2.3.3 for details.

- LS has been converted to a file-structured-device handler to support the new SET LS ENDPAG=n command. Issuing a directory operation (for example, the PRINT command) to LS while the printer is off line or XOFFed can cause the handler (and possibly the system) to appear hung. That is caused by the handler attempting to interrupt a nonexistent device. It also stops the USR from processing directory operations to any other device. Place the printer on line or clear the XOFF condition, or type CTRL/C twice to free the handler.

- The default setting for the LS handler has been changed from NOCTRL to CTRL.

- The following SET commands are now valid for the LS handler:

```
SET LS BIT8
       NOBIT8
       ENDPAG=0
       ENDPAG=n
       SPEED=n
```

  See the RT-11 System User's Guide for more information on those commands.

2.5.12 MM

The following changes have been made to the MM handler.

- For RT-11 V5.4, you can FETCH the MM handler in the XM environment.

- The following SET commands are now valid for the MM handler:

```
SET MM CSR=n
       RETRY=n
       VECTOR=n
```

  For a description of those commands, see the SET dd CSR=n, SET dd RETRY=n, and SET dd VECTOR=n commands in the RT-11 System User's Guide.

2.5.13 MS

The following changes have been made to the MS handler.

- For RT-11 V5.4, you can FETCH the MS handler under the XM monitor.

- For RT-11 V5.4, the MS handler supports 100 in/sec streaming for only the TS05 magtapes under the SJ, FB, and XM monitors. However, currently RT-11 contains no programs (including BUP) that use 100 in/sec streaming for TS05 magtapes under XM.

- The following SET commands are now valid for the MS handler:

```
SET MS CSR=n
       RETRY=n
       VECTOR=n
       CSR2=n
       VEC2=n
       CSR3=n
       VEC3=n
       CSR4=n
       VEC4=n
       CSR5=n
       VEC5=n
       CSR6=n
       VEC6=n
       CSR7=n
       VEC7=n
       CSR8=n
       VEC8=n
```

  See Section 2.3.3 for more information on those commands.

2.5.14 MT

The following changes have been made to the MT handler.

- For RT-11 V5.4, you can FETCH the MT handler in the XM environment.

- The following SET commands are now valid for the MT handler:

```
SET MT CSR=n
       RETRY=n
       VECTOR=n
```

  For a description of those commands, see the SET dd CSR=n, SET dd RETRY=n, and SET dd VECTOR=n commands in the RT-11 System User's Guide.

2.5.15 MU

The following changes have been made to the MU handler.

- For RT-11 V5.4, you can FETCH the MU handler in the XM environment.

- The following SET commands are valid for the new MU handler:

```
SET MU RETRY=n
       SUCCES
       NOSUCCES
       CSR0=n
       VEC0=n
       CSR1=n
       VEC1=n
       CSR2=n
       VEC2=n
       CSR3=n
       VEC3=n
```

See Section 2.3.3 for more information on those commands.

2.5.16 NC, NQ, NU

The following changes have been made to the NC, NQ, and NU handlers.

The following SET commands are valid for the Ethernet handlers NC, NQ, and NU:

For NC:

```
SET NC SHOW
```

For NQ and NU:

```
SET NQ CSR=n
       SHOW
       VECTOR=n

SET NU CSR=n
       SHOW
       VECTOR=n
```

See Section 2.3.3 for information on those commands.

2.5.17 PI

The following changes have been made to the PI handler.

- The RT-11 V5.4 secondary DA (device attributes) reply for the Professional 300 series is now <ESC>[>7;0504c for the FB monitor and <ESC>[>8;0504c for the XM monitor.

- The Professional interface (PI) handler has been modified to support the .SPFUN programmed request and the ISPFN system subroutines for codes 370 (read) and 371 (write). That new PI support is required to support the PRO/GIDIS graphics utility. See the RT-11 Programmer's Reference Manual for a description of the interface between PI and the PRO/GIDIS utility.

2.5.18 RK

The following changes have been made to the RK handler.

The following SET commands are now valid for the RK handler:

```
SET RK CSR=n
       RETRY=n
       SUCCES
       NOSUCCES
       VECTOR=n
```

See the RT-11 System User's Guide for more information on those commands.

2.5.19 SP

The following changes have been made to the SP handler.

The following SET commands are now valid for the SP handler:

```
SET SP ENDPAG=0
       ENDPAG=n
```

See the RT-11 System User's Guide for more information on those commands.

2.5.20 TT

The following change has been made to the multiterminal version of the TT handler.

- The default SET options for the multiterminal TT handler are now:

  ```
  SCOPE,CRLF,PAGE
  ```

  The default SET options for the multiterminal TT handler were previously NOSCOPE,CRLF,PAGE.

2.5.21 VM

The following changes have been made to the VM handler.

- You can now specify the block size for the VM device, using the new SET VM SIZE=nnnnn command. See Section 2.3.3 for details.

- An RT-11 XM monitor can now be booted from the extended memory VM device.

  When you create a bootable VM device, the VM handler you copy to that device must have its base set to the same value as the VM handler used to copy it to the VM device. That is done automatically when you copy the VM handler residing on your system device to the VM device.

2.5.22 XC

The following changes have been made to the XC handler.

- The XC device handler for RT-11 V5.2 cannot be used with earlier versions of VTCOM. Earlier versions of the XC device handler cannot be used with V5.2 VTCOM. Attempts to do so return an error message.

- The following SET command is valid for the XC handler:

  ```
  SET XC SPEED=n
  ```

  See the RT-11 System User's Guide for more information on that command.

2.5.23 XL

The following changes have been made to the XL handler.

- The XL device handler for RT-11 V5.2 cannot be used with earlier versions of VTCOM. Earlier versions of the XL device handler cannot be used with V5.2 VTCOM. Attempts to do so return an error message.

- The following SET commands are valid for the XL handler:

  ```
  SET XL CSR=n
         VECTOR=n
  ```

  See the RT-11 System User's Guide for more information on those commands.

2.6 AUTOMATIC INSTALLATION

The following changes and additions have been made to the automatic installation procedure.

- The automatic installation procedure is now uniform for all supported input and output media. You can use any valid input media to automatically install RT-11 on any valid output media residing on your system configuration. The automatic installation procedure displays a list of valid output devices that reside on your system and queries you as to which output (working system) device you want to use.

  The automatic installation procedure then formats (if necessary) and initializes the output media. It then copies the RT-11 operating system from the input to the output media.

  The following are supported input media: RX02 and RX50 diskettes, TK50 magtape, and RL01 and RL02 disks.

  The following hard disks are the supported output media: RD31 and RD5x (DW), RL01 and RL02 (DL), RK06 and RK07 (DM), and all hard disks supported by the DU handler. Any supported hard disk you want to use as the output medium must contain at least 5600(decimal) blocks of contiguous free space.

- The automatic installation procedure now checks if a fixed output volume is on line, or if a removable output volume is mounted and on line. If it is not, you are referred to the user guide for that device for directions on mounting the volume and/or putting it on line.

- The automatic installation procedure from TK50 magtape checks for sufficient extended memory before attempting the procedure. If there is insufficient extended memory, the automatic installation procedure returns an error message and begins manual installation. That error message is described in Appendix A.

- You can now perform the automatic installation procedure from the DL device without having a second DL device for media backup. However, if your configuration includes only one DL device, you cannot back up the DL distribution medium.

- The automatic installation procedure on Professional 300 series processors no longer queries as to whether you want to configure your working system. If you want to configure your working system, issue the following command after performing the automatic installation procedure and running the installation verification program (IVP):

  ```
  .IND CONFIG
  ```

- If your professional 300 series processor contains an RD52 hard disk manufactured by QUANTUM, you are notified that initializing that disk decreases its available size in blocks. You are referred to the RT-11 On The PRO Cover Letter and queried as to whether you want to continue the procedure.

- The automatic installation start-up command file STARTA.COM now uses FORMAT/VERIFY to format hard disks on Professional 300 series processors before initialization.

- STARTA.COM has been changed to use the new .STRUCTURE directive in IND to determine the file structure of a specified working disk.

  If the .STRUCTURE directive finds a recognized foreign or an unknown file structure, the procedure warns you and gives you the opportunity to continue or stop the procedure.

  If the .STRUCTURE directive finds an RT-11 file structure, the volume is examined for files. If files are found, the installation procedure warns you and gives you the opportunity to continue or stop the procedure.

  See the RT-11 System User's Guide for a description of the .STRUCTURE IND directive.

- Automatic installation is provided for the new TK50 cassette magtape. A new document, the RT-11 Automatic Installation Booklet, TK50 Magtape, describes this procedure. Section 1.5.1 of this manual contains a description of the procedure.

2.7 BINCOM

The following changes have been made to the binary file comparison (BINCOM) program.

- You can now use wildcards with BINCOM to compare multiple binary files.
- The /D option, new for RT-11 V5, compares two entire volumes starting with block 0.

See Chapter 2 of the RT-11 System Utilities Manual for more information on those changes to BINCOM.

2.8 BUP

The following changes have been made to the backup utility (BUP) program.

- By default, BUP successfully handles up to 25 bad blocks on the input device when backing up a disk to a tape or when restoring from a tape to a disk (copy back). If any bad blocks are encountered on the output device, BUP issues a fatal error message and stops the backup operation.

  BUP issues a warning message each time it encounters a bad block on the input device, then continues to backup or restore. If BUP encounters more than 25 bad blocks on the input device, BUP issues a fatal error message and the operation is stopped.

  Use the following customization to set your own value for the number of bad blocks on the input device that BUP will accept.

  In the customization, ..MBAD is the symbol for the value found in the file CUSTOM.TXT, and nnnnn is the new value that you enter. The maximum value you can enter is 77777(octal), that is, you can customize BUP to read 77777 bad blocks on the input device before the next bad block encountered causes a fatal error. The value you enter is also the number of verification mismatches that BUP reads on the input device before the subsequent mismatch causes a fatal error.

```
.RUN SIPP<RET>
*BUP.SAV<RET>
Base?    0<RET>
Offset?  ..MBAD<RET>

     Base          Offset            Old         New?
      000000        ..MBAD            31         nnnnn<RET>
      000000        ..MBAD+2         xxxxx       <CTRL/Y><RET>
*<CTRL/C>
.
```

- BUP can now verify data transferred to or from a device. Use the new /V option for this purpose. BUP prints an error message if the input data does not match the output data.
- BUP can now restore (copy back) a single file from a back up volume. Use the new /F option for this operation. To restore a file from a backed up volume, the volume must have been backed up using the BUP /I option. The /F option must be used with the /X and /I options (/X/I/F).

2.9 CONFIG

The following changes have been made to the unsupported CONFIG utility:

- The following two options are added to the unsupported utility CONFIG:

Option	Function
/D[:yes][:no]	Causes CONFIG to return an error message in addition to setting bits in USERRB. You can assume success if no error message is returned. Using /D:yes once causes CONFIG to continue to return error messages without repeating the /D option. /D:no turns off /D:yes.
/F	Lets you use CONFIG comparison options (/V, /M, and /B) with files. Using /F with a file locks the specified file to CONFIG. That lets you access the file using only the /F option (without specifying the file again), and increases the access speed.

- CONFIG supports a new /T option. Use the /T option to determine if an MSCP device unit contains fixed or removable media and whether that media is available. Use the following command syntax:

```
.RUN CONFIG dev:/T:type
```

where:

dev is the MSCP device unit

type is REM or FIX. Specify REM for type if you want CONFIG to logically assume the MSCP device media is removable. Specify FIX for type if you want CONFIG to assume the MSCP device media is fixed

The following example tests MSCP device DU3 and assumes the media in DU3 is removable:

```
.RUN CONFIG DU3:/T:REM
```

If the media in DU3 is removable, the user error byte (USERRB, byte 53 in the system communications area) is set to 1 for success. If the media is not removable, USERRB is set to 4 for error. If DU3 is not available, USERRB is set to 10 for fatal error.

2.10 DIR

RT-11 V5 includes the following new directory program (DIR) options:

/T includes in the directory listing only those files that are protected against deletion.

/U includes in the directory listing only those files that are not protected against deletion.

See Chapter 4 of the RT-11 System Utilities Manual for more information on those DIR options.

2.11 DUP

The following changes have been made to the device utility program (DUP).

- Warning and error messages for DUP/I/Y (COPY/DEVICE/NOQUERY) and DUP/I/F/Y (COPY/DEVICE/FILES/NOQUERY) operations have been changed for RT-11 V5.4. See Appendix A for information.

- Informational system prompts displayed during DUP/I (COPY/DEVICE) and DUP/I/F (COPY/DEVICE/FILES) operations have been changed for RT-11 V5.4. A system prompt now notifies you if a COPY/DEVICE or COPY/DEVICE/FILES operation will produce a truncated output file or device. A truncated output file or device does not contain the whole input file or device (blocks are not copied and data can be lost).

  A system prompt now also notifies you when a COPY/DEVICE or COPY/DEVICE/FILES operation is from a smaller device or file to a larger device, because the /DEVICE option copies home and directory block information and therefore can reduce the available size of the output device.

  A COPY/DEVICE operation where the output device is smaller than the input device returns the following system prompt:

```
Output device is smaller than input device
Output device will be truncated; Are you sure?
```

  A COPY/DEVICE operation where the output device is larger than the input device returns the following system prompt:

```
Output device is larger than input device; Are you sure?
```

  A COPY/DEVICE/FILES operation where the output file is smaller than the input device returns the following system prompt:

```
Output file is smaller than input device
Output file will be truncated; Are you sure?
```

  A COPY/DEVICE/FILES operation where the output device is smaller than the input file returns the following system prompt:

```
Output device is smaller than input file
Output device will be truncated; Are you sure?
```

A COPY/DEVICE/FILES operation where the output device is larger than the input file returns the following system prompt:

Output device is larger than input file; Are you sure?

- The /H (/VERIFY:only) option when combined with the /K or /B option is now valid (with certain restrictions) for MSCP class devices.

  Use /H to clear bad blocks caused by soft errors. RT-11 reads each block and if it encounters an error, writes the block out and then reads it again. That forces the controller to clear (make usable) blocks that return a soft error. Use /H/K (DIRECTORY/BADBLOCKS/VERIFY:only) only with blank media or with media you have just backed up. This functionality is available only as a CSI-level command and is not implemented as a KMON command option.

  The distributed RT-11 supports the /H option for RD31, RD51, RD52, and RD53 MSCP devices. You must perform a SYSGEN and request DU handler bad-block replacement support to use the /H option with RC25 or RA80 devices.

  The /H option when combined with the /K or /B option is recommended only for MSCP class devices.

- When you use the /I option to copy a larger volume to a smaller one, DUP asks for confirmation before copying the volume.

- You can now abort a /W (WAIT) operation.

- You can now combine the /R and /I options to preserve the output volume's bad-block replacement table. In RT-11 V4, the input volume's bad-block replacement table was always transferred to the output volume.

- It is no longer necessary to customize DUP to use variable-size volumes. Instead, a new handler attribute VARSZ$ can be set to indicate a variable-size volume. Use the stat argument for the .DRDEF programmed request to set VARSZ$ (bit mask 400). Then, you can use the .DSTATUS programmed request to determine whether the handler supports variable-size volumes (and .SPFUN 373).

  See the RT-11 Programmer's Reference Manual for more information.

See Chapter 6 of the RT-11 System Utilities Manual for more information on those changes to DUP.

2.12 EDIT

The following change has been made to the EDIT editor program:

- EDIT is no longer the distributed default editor. To make EDIT your default editor, include the command SET EDIT EDIT in your start-up command file.

2.13 ERROR LOGGER

The following changes have been made to the error logger subsystem.

- The SJ monitor now supports the error logger. Refer to Chapter 16 of the RT-11 System Utilities Manual for information on how to use the error logger under the SJ monitor.
- The RT-11 V5 error logger reports the number of retries for a single error and the final status of the operation (success or failure). The error logger provides separate entries for retries only if the registers differ.
- You can choose to log successful I/O transfers and errors, or only errors. Use the SET dd SUCCES command to log successes as well as errors, and the SET dd NOSUCCES command to log only errors.
- The error logger supports the new DW (RD50/RD31/RD51/RD52 disk) and DZ (RX50 diskette) handlers for the Professional series processors.
- The error logger supports the RC25 and RA80 disks.

See Chapter 16 of the RT-11 System Utilities Manual for more information on changes to the error logger.

2.14 FILEX

The following changes have been made to the file exchange (FILEX) program.

- The default device for all FILEX operations is DK:.
- There are two new FILEX options:

/V[:ONL]	Use with /Z and /U[:n] simultaneously to write a volume identification during initialization of an interchange diskette. Use the [:ONL] argument to change an interchange diskette's volume ID without initializing the diskette. Use with /L or /F to list the volume ID of an interchange diskette when obtaining a directory listing.
/W	Initiates the operation but pauses and waits for you to mount different volumes.

See Chapter 7 of the RT-11 System Utilities Manual for more information on those changes.

2.15 FORMAT

The following changes have been made to the volume formatting (FORMAT) program.

- Formatting RD50, RD31, RD51, and RD52 hard disks is supported for Professional 300 series processors using the commands FORMAT, FORMAT/VERIFY, and FORMAT/VERIFY:ONLY. See Section 4.1 for a description of formatting an unformatted hard disk on Professional 300 series processors.

- The /VERIFY:only option can be used to perform the write/read verify operation on the following devices: DD, DL, DM, DP, DT, DW, DX, DY, DZ, and RK.

- If you try to format a volume that contains protected files, or try to format a volume while a foreground job is loaded, FORMAT warns you and asks you to confirm the operation.

- You can now abort a /W (WAIT) operation.

- The table of verification bit patterns has been increased to 16 patterns. The last 4 of the 16 patterns are reserved for future use.

- Formatting of devices at nonstandard addresses is now supported and will occur automatically, based on the CSR location specified in the device handler.

See Chapter 8 of the RT-11 System Utilities Manual for more information on those changes to FORMAT.

2.16 HELP

The following change has been made in the HELP utility.

- The files HELP.TXT and HELP.EXE, which together make up the program HELP.SAV, are no longer provided on the distribution kit. See Section 2.7.14 in the RT-11 Installation Guide for information.

2.17 IND

The following additions and changes have been made to the indirect control file (IND) processor.

- The .TESTDEVICE and .VOL directives now support the <FILERR> special symbol. That lets you intercept some read-related errors and perform error recovery.

- IND now dynamically determines whether RT-11 is running the XM monitor. If IND is running under the XM monitor, a portion of IND permanently resides in extended memory. That portion is displayed by the SHOW MEMORY command. Use the REMOVE IND command to eliminate that portion of IND in extended memory.

- IND dynamically allocates a region in extended memory for its symbol table when running RT-11 under the XM monitor. You can suppress the dynamic allocation using a customization patch located in the RT-11 Installation Guide.

- The .STRUCTURE directive has been added to IND to determine the file structure of a specified file-structured device from boot block information.

- The .TESTDEVICE directive has been modified to include ATT or NAT (attached or not attached) in field nine of special symbol <EXSTRI>.

- The .ONERR directive has been modified to include processing integer under/over flow conditions.

- The .SETL directive has been modified to support the .AND, .OR, and .NOT logical test directives. When you use the .SETL directive, the logical symbol you specify is set to the value represented by the logical expression.

- The following operating modes have been added to the .ENABLE and .DISABLE directives. (These operating modes also apply to the .IFENABLED and .IFDISABLED directives.)

  .ENABLE/.DISABLE ABORT

  .ENABLE/.DISABLE CONTROL-Z

  .ENABLE/.DISABLE TYPEAHEAD

- If the .ENABLE TIMEOUT directive is issued but the system does not include timer support, IND assigns special symbol <EXSTAT> the value 0, for warning, instead of printing an error message.

See Chapter 5 of the RT-11 System User's Guide for complete information on using IND.

2.18 KED

The following changes have been made to the keypad editor (KED).

- KED (or KEX) now sets bits in the USERRB byte (byte 53) to indicate whether you use the EXIT or QUIT command to leave the editor. If you exit KED, it sets bit 0 in USERRB for success. If you quit KED, it sets bit 1 in USERRB for warning. Therefore, if you have set the error severity level to warning (SET ERROR WARNING) and are running KED from within an indirect file, quitting KED will cause an exit from the indirect file.

- KED is the default editor for the SJ and FB monitors, and KEX is the default editor for the XM monitor.

- <CTRL/R> or <CTRL/W> redisplays the screen.

- <GOLD><CTRL/U> performs the same function as <GOLD><PF4>. (GOLD is the PF1 key.)

- <GOLD><DEL> performs the same function as <GOLD><,>. (GOLD is the PF1 key.)

- <GOLD><LINEFEED> performs the same function as <GOLD><DELETE>. (GOLD is the PF1 key.)

- On terminals with AVO (advanced video option) or its equivalent, select ranges are displayed with reverse background and boldface.

- If the command EDIT/KED filespec is issued and the file you specify is not found, KED prompts you for permission to create the file. If the file is protected, KED prompts you for permission to inspect the file.

- A new version of KED, called KEX, runs as a virtual job under XM. KEX can run as a background job, a foreground job, a system job, or as all three simultaneously.

- The version of KED you are using appears as the prefix for messages displayed (?KED-, ?K52-, or ?KEX-). Formerly, all error messages were prefixed by ?KED-.

- 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 software customization described in the RT-11 Installation Guide.

2.19 LIBR

The following changes have been made to the RT-11 librarian (LIBR) program.

- With RT-11 V5, LIBR continues instead of exiting when an error occurs.

- The default file type for MACRO-11 libraries has been changed to .MLB.

See Chapter 10 of the RT-11 System Utilities Manual for more information on LIBR.

2.20 LINK

The following changes have been made to the RT-11 linker (LINK) program.

- RT-11 V5.3 removes the restriction stating that the total combined size of all virtual overlay segments created by the LINK /V option cannot exceed 96K words. Now, the total combined size is the amount of available memory.

- A problem with LINK's CSI processing has been corrected, so that with RT-11 V5 you can link an increased number of modules at one time.

- The /K:n option is no longer restricted to use for RSTS compatibility. You can now use the /K:n option with RT-11 to limit the number of words allocated by a .SETTOP programmed request.

- RT-11 V5 includes two new LINK options:

  /D Defines the global symbol you specify once in each segment that references that symbol. Such global symbols must be defined in a library module.

  /N Produces in the load map a cross-reference listing of all global symbols defined during the linking process.

See Chapter 11 of the RT-11 System Utilities Manual for more information on those changes to LINK.

2.21 MACRO-11

The following changes have been made to MACRO-11.

- RT-11 supports the new MACRO-11 8-bit functionality for the multinational character set. MACRO-11 supports the 8-bit multinational character set for some directives. Refer to Appendix J of the PDP-11 MACRO-11 Language Reference Manual, Update 1, for information on 8-bit support.

- The default size of the MACRO-11 work file can be increased from 128(decimal) blocks to a maximum of 256(decimal) blocks using a customization patch located in the RT-11 Installation Guide.

2.22 MDUP

The following change has been made to the magtape device utility program.

- The magtape utility (MDUP) includes a new /V:n option. The /V:n option checks that extended memory contains at least the number (n) blocks you specify. If extended memory contains n blocks, MDUP executes commands until another /V is encountered, whereupon it stops executing the commands.

If extended memory contains less than n blocks, MDUP does not execute commands until another /V is encountered, whereupon it starts executing the commands.

2.23 ODT

The following change has been made in the Octal Debugging Technique program.

- The following procedure lets you create a monitor-independent ODT debugger; that is, an ODT that does not require that the operating system be loaded in memory.

  The distributed debugger ODT.OBJ requires that the operating system be loaded in memory. You can modify ODT so it does not require that the operating system be loaded in memory. That could be useful, for example, in debugging the bootstrap.

  Use the following procedure to create a debugger called ODTHWD.OBJ that functions independently of the operating system:

  1. Use KED to create the following patch program. Name it ODTPAT.MAC. In the program, substitute for the symbol ..GVAL, the value for that symbol located in CUSTOM.TXT.

     ```
     .TITLE  ODT
     .PSECT  $ODT$
     BASE=..GVAL
     .=.+BASE
     BR      .+34
     .END
     ```

  2. Assemble the created patch program:

     ```
     .R MACRO<RET>
     *ODTPAT=ODTPAT<RET>
     *<CTRL/C>
     ```

  3. Create the monitor-independent debugger ODTHWD.OBJ by modifying ODT.OBJ. This does not destroy the distributed ODT.OBJ, but modifies a copy of it. Use the utility PAT.SAV in the following manner:

     ```
     .R PAT<RET>
     *ODTHWD=ODT,ODTPAT<RET>
     *<CTRL/C>
     ```

  4. Explicitly specify ODTHWD in the LINK/DEBUG command. If you do not specify ODTHWD, RT-11 by default links the distributed ODT.

     ```
     .LINK/DEBUG:ODTHWD program<RET>
     ```

2.24 PAT

The following change has been made in the Object Module Patch program.

- When PAT finishes executing a command, control returns to the CSI (indicated by the asterisk prompt, *) rather than to the keyboard monitor.

2.25 PIP

The following changes have been made to the peripheral interchange (PIP) program.

- PIP now performs a .RCTRLO request between files, when multiple files are copied to the terminal. This new functionality lets you use CTRL/O to selectively stop the display of files on the terminal screen. Executing a CTRL/O discards output until another CTRL/O is executed, or until the beginning of the next file is reached, whereupon output to the terminal resumes.

- If a PIP command line includes file transfers from magtape, PIP performs all file transfer operations requested on the command line in the order in which the files appear on the volume rather than the order in which they are specified in the command line.

- The /C option now accepts the [:date] argument. Use /C[:date] to include files of a certain date in the operation you specify.

- You can now abort a /E (WAIT) operation.

- In RT-11 V5, you can use the /F (PROTECTION) and /Z (NOPROTECTION) options alone or for copy operations as well as for rename operations. It is no longer necessary to use /R (/RENAME) with /F or /Z.

- The /Q (QUERY) option is no longer the default when deleting files, except when you include wildcards in the file specification.

- The /T (SETDATE) option now accepts the [:date] argument, so you can assign files dates other than the current system date.

- The /Y (SYSTEM) option is now necessary only when you specify wildcards in the input file types.

- The following PIP options are new:

  /H — (VERIFY) Verifies that the output file matches the input file after a copy operation. This option is invalid with /A (ASCII) and /B (BINARY).

  /I[:date] — (SINCE) Includes only those files created on or after the specified date.

/J[:date] (BEFORE) Includes only those files created before the specified date.

/V (MULTIVOLUME) Copies files from one input volume to two or more smaller output volumes.

/X Prints an informational (I) message rather than a fatal (F) message when PIP cannot find a file specified in the command line.

See Chapter 13 of the RT-11 System Utilities Manual for more information on those changes to PIP.

2.26 QUEUE PACKAGE

The following changes have been made to the queue package.

- The PRINT command is affected when you run SPOOL and QUEUE together. See Section 2.3.1.1 (PRINT command).
- QUEUE appends a form-feed character <FF> to the end of each copy of a queued file, whether the output is to a disk, a serial line printer, or a parallel line printer.
- The PRINT/FLAGPAGE:n option, when a value is specified for n, overrides the transparent spooler (SPOOL) SET SP FLAG=n command. When no value is specified for n with the /FLAGPAGE:n option, the value for n is set by the SET SP FLAG=n command.
- The SHOW QUEUE command is now processed by the new RESORC /Q option rather than by the QUEMAN /L option. However, the /L option remains for the sake of compatibility.
- When QUEUE sends a job consisting of more than one input file to an RT-11 file-structured device, QUEUE now copies each input file to a separate output file with the same file name and type. The job name is printed in the JOBNAME field of the banner page. In RT-11 V4, all input files in the same job were concatenated into one output file with the file type .JOB.
- Input files are now protected from deletion while QUEUE is copying them to the output device.
- The default number of banner pages printed when you use the /H command is now determined by the number of banner pages you set as the default with the /P command.
- When the input device for QUEUE operations is MT, to save time MT no longer rewinds between files.
- QUEUE's work file is now SY:QUFILE.WRK. In RT-11 V4, the work file was DK:QUFILE.TMP.
- The following are new QUEMAN options:

  /C[:date] (DATE) Prints only those files with the specified creation date.

/I[:date]	(SINCE) Prints only those files created on or after the specified date.
/J[:date]	(BEFORE) Prints only those files created before the specified date.
/Q	Requests confirmation for each file to be included in the operation. QUEMAN prints the name of each file and pauses; you must respond Y for each file you want to include.
/W	Prints on the console a log of the files included in the operation.
/X	Allows QUEMAN to continue processing instead of halting when it cannot find a file specified in the command line.

See Chapter 17 of the RT-11 System Utilities Manual for more information on those changes to the Queue Package.

2.27 RESORC

The following changes have been made to the resource (RESORC) program.

- RESORC option DU:/D displays new DU status information. The port display now indicates an invalid port by displaying an asterisk (*) before the port number. For example, port = *n, where n = 0,1,2,3, indicates that port n was not installed. If RESORC option DU:/D displays port = n, where n = 0,1,2,3, then port n is installed.

- RESORC includes a new /V option. The /V option displays the release and version numbers for any module in the RT-11 distribution kit.

  Use the /V option to supply the release and version numbers for any RT-11 modules quoted in an SPR (software performance report) submission.

  In the following example, the /V option reports the release and version numbers for the LS handler:

```
*LS.SYS/V
Release = V05, Version(s) = 3
```

- RESORC options /A, /H, and /Z now display processor support for the EAE (extended arithmetic element) hardware option.

- RESORC options /A and /X now display global regions in extended memory.

- RESORC options /A and /X now display a new fourth column in the extended memory map that indicates the type of region being listed.

- The /O option now shows if SYSGEN special feature FPU (floating point unit) support has been selected.

- RESORC displays the following new processor types:

  PDP-11/53
  PDP-11/83
  PC380
  PDP-11/73A
  PDP-11/73B
  PDP-11/84, UNIBUS

- The "KT11 Memory Management Unit" display now reads "Memory Management Unit".

- The /A (ALL) option now provides information about the total amount of memory on the system, logical disk subsetting assignments, and organization of physical memory.

- The /C (CONFIGURATION) option now provides status information for SET KMON [NO]IND, SET EXIT [NO]SWAP, SET EDIT, SET SL ON/OFF, and the global .SCCA flag.

- The /D (DEVICES) option now accepts the optional argument dd (dd:/D), where dd represents the 2-letter permanent device name. You can use the argument dd to obtain information about a specific device.

- The /H option now includes the total amount of memory on the system.

- The /O option now also reports whether global .SCCA support was chosen during system generation.

- The RESORC utility now includes the following new options:

  /Q Lists the contents of the queue for QUEUE or SPOOL, depending on which is running. The SHOW QUEUE keyboard command is performed by this option.

  /S Displays information about logical disk subsetting assignments.

  /X Displays information about the organization of physical memory: where jobs and handlers are loaded and where KMON and the USR will reside.

See Chapter 14 of the RT-11 System Utilities Manual for more information on those changes to RESORC.

2.28 SETUP

The following changes have been made to the hardware characteristics (SETUP) program.

- You can specify the color amber for color1 or color2 in the SETUP [color1] [ON color2] command. You can create and store the exact shade of amber you want using the SETUP SETCOLOR command.

- SETUP now supports two new destinations for only the LANG:code mode: KB and VIDEO. The LANG:code mode is supported only for Professional 300 series processors.

- SETUP now supports the construction of display colors from primary colors using the new SETCOLOR mode. SETCOLOR mode is supported only for Professional 300 series processors.

- SETUP now supports the NRC (national replacement character set) for the Professional 300 series keyboard and video monitor. Support for each NRC keyboard is selected with the SETUP LANG:code command. SETUP also includes the following new modes for more complete NRC support: COMPOSE, DATA, RETAIN, and TYPE. NRC support includes the use of Dead Diacritical keys with various NRC keyboards. There are no valid Dead Diacritical keys on the USA keyboard.

- SETUP now supports the COMPOSE CHARACTER key on the Professional 300 series keyboard with the new modes COMPOSE and NOCOMPOSE. COMPOSE mode lets you use combinations of two keyboard keys to construct characters from the various national replacement character (NRC) sets.

- SETUP now supports 480INTERLACE for Professional 380 series terminals.

- SETUP [NO]INTERLACE is now supported for all Professional 300 series video terminals. The shortest valid abbreviation for INTERLACE is INT.

- SETUP supports new functionality for laser printers with the LANDSCAPE and PORTRAIT modes.

- SETUP RESET now returns the text and background colors to the settings that were set with the most recent SETUP SAVE command.

See the RT-11 System User's Guide for more information on these changes to SETUP.

2.29 SIPP

The following changes have been made to the save image patch (SIPP) program.

- When SIPP is used to patch a file, the creation date of the patched file is changed to the current system date. If no modifications are made, the date remains unchanged.

- When using SIPP to create an indirect command file, the command file contains the command R SIPP rather than RUN SIPP. That lets you run the command file from a volume other than the system volume.

See Chapter 22 of the RT-11 System Utilities Manual for more information on these changes to SIPP.

2.30 SL

The following changes have been made to the Single-Line Editor program.

- Two functions, Get Older, and Save and Get Saved, have been added to the single-line editor (SL).

2.31 SLP

The following changes have been made to the source language patch (SLP) program.

- SLP ignores any characters that precede the start-of-update character (-) in SLP command files. If SLP is unable to find the start-of-update character, SLP prints an error message and returns control to the CSI (indicated by the asterisk prompt).

- You can now update more than one file in a single SLP command file. Type a double slash (//) on a line by itself after the update text for each file. On the next line, type the command line that specifies the next input file to be updated and the command file name (the same command file that contains the update text). Then type the update text on the lines that follow. Type a single slash (/) on a separate line to indicate the end of a series of update texts.

- The SLP utility includes the following new options:

/C[:n]	Determines or validates the contents of the SLP input file or the SLP command file. Use /C to determine the checksum of a file. Use /C:n to verify the contents of a file. SLP computes the checksum for the file and compares the checksum to the value you specify for n.
/N	Suppresses the creation of a backup file when SLP updates the input file.

See Chapter 23 of the RT-11 System Utilities Manual for more information on those changes to SLP.

2.32 SPOOL

The following changes have been made to the transparent spooler (SPOOL) package.

- The PRINT command is affected when you run SPOOL and QUEUE together. See Section 2.3.1.1 (PRINT command).

- The PRINT/FLAGPAGE:n option, when a value is specified for n, overrides the SET SP FLAG=n command. When no value is specified for n with the /FLAGPAGE:n option, the value for n is set by the SET SP FLAG=n command.

2.33 SRCCOM

The following changes have been made to the source file comparison (SRCCOM) program.

- The syntax of the SRCCOM command has changed to:

  ```
  [[out-filespec][,SLP-filespec]=]old-filespec,new-filespec[/options]
  ```

  The new syntax element, [SLP-filespec], lets you create a differences file and a SLP command file in the same command line. With RT-11 V4, you could create only one or the other. Because you can specify both with the RT-11 V5 syntax in your command line, the /P option has been eliminated.

- You can now use wildcards with SRCCOM to compare multiple source files.

See Chapter 15 of the RT-11 System Utilities Manual for more information on those changes to SRCCOM.

2.34 SYSGEN

The following are changes and additions to the system generation procedure:

- Multiport booting. You can select support for multiport booting for DU, which allows you to bootstrap RT-11 from any port. Without multiport booting support, you can bootstrap RT-11 only from port 0.

- DU initiated bad-block replacement. Bad-block replacement is a technique in which substitute blocks are provided for blocks that have caused a read or write error. Controllers for certain MSCP devices do not replace bad blocks. If your system includes the XM monitor, SYSGEN now asks you if you want DU to perform the bad-block replacement.

  DIGITAL recommends that you request DU bad-block replacement support if you are using an RC25 or the RA80/RA81 series MSCP disks. Failing to provide DU bad-block replacement support for those devices can cause the loss of data or excessive bad blocks on those devices.

  This SYSGEN option is available for only the XM monitor.

- SYSGEN now displays the default CSR addresses and vector addresses for each port you want the MSCP disk class handler (DU) to support.

- In system generation dialog questions that ask for CSR and vector addresses, the default values are displayed in parentheses. Those default values are now determined by the CSR and vector addresses that you entered in response to previous questions. The standard defaults are displayed if you accepted the defaults in response to previous questions.

2.34.1 New SYSGEN.COM Dialog Questions

The following dialog questions have been added to SYSGEN.COM for the new DU bad-block replacement and DU multiport support.

109. What is the CSR (register) address for the <nth> port (nnnnnn)?

Enter the CSR address (octal) of the <nth> port. The valid range for the CSR address is 160000 to 177570. The standard address is 172150.

If you request multiple port support for DU, SYSGEN asks questions 109 and 110 once for each port you wish to support. The default (octal) addresses appear in parentheses. If your system includes nonstandard addresses, respond with the address of each DU port. RT-11 can support up to four DU controllers (ports). The following list shows the default CSR addresses and vector addresses that correspond to each port:

Port	CSR (Register) Address	Vector Address
0	172150	154
1	172144	150
2	172140	144
3	172134	140

110. What is the vector address for the <nth> port (nnn)?

The default vector addresses appear in parentheses. If your system includes nonstandard addresses, enter the programmable vector address. The valid range for the vector address is from 100 to 474.

111. Do you want support for multiple port booting (NO)?

Multiple port booting support allows you to boot RT-11 from any DU port. If you do not select support for multiple port booting, you can boot RT-11 from only port 0.

SYSGEN asks this question only if you select support for more than one port.

112. Do you want support for DU initiated bad block replacement (NO)?

Bad block replacement is a technique in which substitute blocks are provided for blocks that have caused a read or write error. If your system includes an RC25, or an RA series device, you should request support for DU-initiated bad block replacement. Controllers for those devices do not replace bad blocks. Enter YES if you want DU to perform the bad block replacement.

SYSGEN asks this question only if you select the XM monitor.

2.34.2 Changed SYSGEN.COM Question Numbers

The following SYSGEN dialog question has been changed from question 110 to question 140:

110. Do you want support for banner pages (Y)?

is now question:

140. Do you want support for banner pages (Y)?

2.34.3 Changed and New System Conditionals

The following system conditionals have been changed:

```
DU$CSR     = nnnnnn  ;STATUS REGISTER OF FIRST MSCP
DU$VEC     = nnn     ;VECTOR OF FIRST MSCP
DU$CS1     = nnnnnn  ;STATUS REGISTER OF SECOND MSCP
DU$VC1     = nnn     ;VECTOR OF SECOND MSCP
DU$CS2     = nnnnnn  ;STATUS REGISTER OF THIRD MSCP
DU$VC2     = nnn     ;VECTOR OF THIRD MSCP
DU$CS3     = nnnnnn  ;STATUS REGISTER OF FOURTH MSCP
DU$VC3     = nnn     ;VECTOR OF FOURTH MSCP

MU$CSR     = nnnnnn  ;STATUS REGISTER OF FIRST TMSCP MAGTAPE UNIT
MU$VEC     = nnn     ;VECTOR OF FIRST TMSCP MAGTAPE UNIT
MU$CS1     = nnnnnn  ;STATUS REGISTER OF SECOND TMSCP MAGTAPE UNIT
MU$VC1     = nnn     ;VECTOR OF SECOND TMSCP MAGTAPE UNIT
MU$CS2     = nnnnnn  ;STATUS REGISTER OF THIRD TMSCP MAGTAPE UNIT
MU$VC2     = nnn     ;VECTOR OF THIRD TMSCP MAGTAPE UNIT
MU$CS3     = nnnnnn  ;STATUS REGISTER OF FOURTH TMSCP MAGTAPE UNIT
MU$VC3     = nnn     ;VECTOR OF FOURTH TMSCP MAGTAPE UNIT
```

The following system conditionals are new or not previously documented:

```
DU$ALT     = nnnnnn  ;STATUS REGISTER FOR FALCON DU HANDLER
                     SUPPORT
DU$BOO     = 1       ;DU HANDLER MULTIPORT BOOTING SUPPORT
DU$BBR     = 1       ;DU HANDLER DYNAMIC BAD BLOCK REPLACEMENT
                     SUPPORT
NU$CSR     = nnnnnn  ;STATUS REGISTER OF UNIBUS ETHERNET
                     ;CONTROLLER
NU$VEC     = nnn     ;VECTOR OF UNIBUS ETHERNET CONTROLLER
```

2.35 SYSTEM SUBROUTINE LIBRARY (SYSLIB)

The following changes have been made to the system subroutine library (SYSLIB).

- GIWRIT - The msglen parameter accepts a value equal to or greater than -1. Specify the -1 value for msglen to reset GIDIS.
- GTLIN - This routine now has an optional third argument that can request special services. The optional third argument is a string constant specified as 'term' or 'plain'. The 'term' argument takes input only from the console terminal, even if the program is running under the control of an indirect command file. The 'plain' argument takes unaltered input from the chain area, and passes the input to the array specified by the GTLIN result argument.
- ICSTAT - This routine can now be used under the SJ monitor.
- IGTDUS - This routine now returns information on TMSCP class devices.
- ISLEEP - This routine can now be used under the SJ monitor.
- IUNTIL - This routine can now be used under the SJ monitor.

- ILUN - This routine now calls a local copy of the $FCHNL routine. That copy of $FCHNL does not assign a logical unit number (LUN) to an available channel if the LUN is not already assigned and thus prevents the channel address table from filling up.

- ISPFN subroutine calls - The ISPFN subroutine calls (ISPFN/ISPFNC/ISPFNF/ISPFNW) have had support added for the DU, DW, DZ, MU, NC, NQ, NU, and PI handlers.

  For DU, ISPFN 376 and 377 can return the following error code (in addition to those returned by DM):

140000	A forced error occurred. The device controller or DU handler discovered bad data on a good (replaced) block. (Bad-block replacement was performed but no data was recovered.)

  For DW, ISPFN 376 and 377 now use logical block numbers rather than physical block numbers in the blk argument. Therefore, to address a write or read to physical block zero, specify -1 for the blk argument. That is necessary because the physical block number of a DW device is one less than the logical block number.

  For MU, ISPFN 374 (write with extended file gap) executes as ISPFN 371 (write).

  The DU handler has had support added for ISPFN 376 and 377. For DU, ISPFN 376 performs a write to the specified sector, and ISPFN 377 performs a read from the specified sector. Those writes and reads are not absolute; bad-block replacement and block vectoring remain in force. See Section 2.5.4 for information.

See the RT-11 Programmer's Reference Manual for complete information on the system subroutine library (SYSLIB).

2.36 SYSTEM MACRO LIBRARY (SYSMAC)

The following programmed requests have been changed:

- .DRBEG - An additional word (word 6) has been added to the handler information set up in block 0 by .DRBEG. The word has the value of a "NOP" instruction. The lowest 5 bits (0 through 4) of the word are used as bit flags to indicate the presence of entry points in block 0 for the fetch, release, load, and unload handler service routines. Those entry points are generated by the .DRPTR request. The following list shows the lowest 5 bits of the new sixth word set up by .DRBEG and their meaning when set:

Bit	Meaning when set
0	Fetch entry point exists in block 0
1	Release entry point exists in block 0
2	Load entry point exists in block 0
3	Unload entry point exists in block 0
4	Reserved

Handlers and programs that interact with .DRBEG in a nonstandard manner or use the size of the code generated by .DRBEG must account for that additional word.

- .DREND - Two optional arguments, FORCE and PSECT, have been added to the .DREND programmed request. The form of the .DREND programmed request is now:

  Macro Call: name[,force=n][,psect]

  where:

  force — causes the specified value(s) from the SYSGEN Features Word (RMON fixed offset 372) to be entered in the handler vector table. For example, specifying force=4 generates the device timeout vector in the table. Generating a vector in the handler vector table does not create support in that handler for a SYSGEN feature. You can create multiple vectors by adding the values from the SYSGEN Features Word; the value is a bit mask. The default value for force is 0.

  psect — forces the .DREND request to be placed in the specified program section at link time. Use this argument when the handler is built from several psects and you want to force location of .DREND code in the memory-resident section of the handler.

- .CRRG - can now return the following new octal error codes:

  12 Global region not found
  13 Too many global regions (none free)
  15 Global region is privately owned

- .CSIGEN - for RT-11 V5.4, .CSIGEN under all monitors does not perform a close on any channels. When you call .CSIGEN, it performs a purge on channels 0 through 10(octal). If CSI errors occur, .CSIGEN purges channels 0 through 10(octal).

- .CSTAT - can now be used under the SJ monitor.

- .DRDEF - A new handler attribute, VARSZ$, is for .DRDEF's stat argument. Setting VARSZ$ (bit mask 400) indicates that the handler supports variable-size volumes and .SPFUN 373.

- .ELRG - can now return the following octal error code:

  14 Global region in use

  The .ELRG programmed request will now concatenate contiguous areas of memory that are segmented in the allocation table, when memory is restored after a region is eliminated.

- .FETCH - can now be used under the XM monitor.

- .GTLIN - now includes a terminal option (.GTLIN ,,TERM=YES), which forces input to come from the terminal rather than from the active command or control file.

- .MAP - now checks to see if the specified window is already mapped. If it is, no unmapping and remapping operations are performed.

- .SCCA - has been modified to include global .SCCA support. Global .SCCA support allows you to prevent double CTRL/C aborts by causing the system to ignore all CTRL/C characters issued from the terminal.

- .SPFUN - has had support added for the DU, DW, DZ, MU, NC, NQ, NU, and PI handlers.

  For DU, .SPFUN 376 and 377 can return the following error code (in addition to those returned by DM):

140000	A forced error occurred. The device controller or DU handler discovered bad data on a good (replaced) block. (Bad-block replacement was performed but no data was recovered.)

  For DW, .SPFUN 376 and 377 now use logical block numbers rather than physical block numbers in the blk argument. Therefore, to address a write or read to physical block zero, specify -1 for the blk argument. That is necessary because the physical block number of a DW device is one less than the logical block number.

  For MU, .SPFUN 374 (write with extended file gap) executes as .SPFUN 371 (write). Write with extended gap functionality is not supported by TMSCP magtapes.

  The DU handler has had support added for .SPFUN 376 and 377. For DU, .SPFUN 376 performs a write to the specified sector, and .SPFUN 377 performs a read from the specified sector. Those writes and reads are not absolute; bad-block replacement and block vectoring remain in force. See Section 2.5.4 for information.

- .TWAIT - can now be used under the SJ monitor.

See the RT-11 Programmer's Reference Manual for complete information on the system macro library (SYSMAC).

2.37 UCL

The following changes have been made to the user command linkage (UCL) utility.

- The data structure for the data file UCL.DAT has been redesigned. Data files created prior to RT-11 V5.2 must be manually recreated.

- UCL supports the new SHOW COMMANDS command, which displays command definitions on the console terminal. SHOW COMMANDS also supports the /PRINTER and /OUTPUT:filespec options.

- The CSI (command string interpreter) interface is not supported for UCL; do not issue the R UCL command.

- The maximum number of user-defined UCL commands in the command table in the UCL data file can be changed (increased or decreased) from 31(decimal) to a maximum you specify.

- You can optimize the UCL utility by combining the save image program (.SAV) file and the data (.DAT) file.

- Informational messages, in the form ?UCL-I-, have been removed from UCL. Warning prompts, in the form Are you sure?, have also been removed. You no longer need to verify your wish to replace or delete a command. Execution of a command to define, redefine, or delete a UCL command is indicated by the appearance of the monitor prompt (.).

See the RT-11 System User's Guide for information on using UCL.

2.38 VTCOM

The following changes have been made to the virtual terminal communication (VTCOM) package.

- Transfer speed enhancements were made to VTCOM for a previous release of RT-11. VTCOM sends data at the interface interrupt speed. That transfer speed can be too fast for the host terminal service to process. A symptom is a beeping terminal. To compensate, slow down the baud rate or let VTCOM adjust the transfer rate, using retries and reduced packet size.

- VTCOM for RT-11 V5.2 cannot be used with earlier versions of the XC and XL device handlers. The XC and XL device handlers for V5.2 cannot be used with earlier versions of VTCOM. Attempts to do so return an error message.

- VTCOM.REL and VTCOM.SAV now support the Mini-Exchange, using the new VTCOM command SELECT.

- The shortest valid abbreviation for the SELECT command is SEL, and for the SEND command is SEN.

- VTCOM includes a new HANGUP command.

- Enhancements made to the file transfer code in VTCOM can increase the speed of a transfer through more efficient use of the serial line.

- The VTCOM SEND command sends ASCII files at two speeds: SLOW or FAST. The distributed default speed is FAST. Use SLOW if the host terminal service does not support XON/XOFF and FAST if it does support XON/XOFF. A new customization in the RT-11 Installation Guide lets you set the VTCOM SEND command speed.

- The customization patch used to set a default dial string for the VTCOM DIAL command now lets you specify the actual character itself, rather than the octal ASCII value for the character.

See the RT-11 System Utilities Manual for a complete description of using VTCOM.

CHAPTER 3

CURRENT RESTRICTIONS AND CORRECTED PROBLEMS

This chapter lists the current restrictions, corrected software problems, and corrections and additions for the RT-11 documentation set.

3.1 CURRENT RESTRICTIONS

When running RT-11 V5.4, observe the following restrictions.

3.1.1 Handler Restrictions

The following restrictions apply to RT-11 handlers.

3.1.1.1 General Handler Restrictions - The following restrictions apply to all RT-11 handlers.

- You must LOAD any device handler that is used by a system job. For example, if your system device (SY) uses a different handler than your data device (DK), you must LOAD the data device handler before running any system job that uses that data device. You should LOAD any device handler that is used by a foreground job.
- Handlers built by RT-11 V5.3 (or subsequent) or distributed with V5.3 (or subsequent) should not be run on and cannot be built by earlier versions of RT-11.
- You can encounter problems when customizing the SD, SL, or ST handler file name suffix. DIGITAL does not recommend the use of any suffixes other than X for the XM monitor or M for RTEM-11 with those handlers.
- Device handlers must now be linked with the /NOBITMAP option.

3.1.1.2 Specific Handler Restrictions - The following restrictions apply to only the indicated handler.

DU

- Support for DU multiport booting requires a V5.3 (or subsequent) DU handler running under a V5.3 (or subsequent) monitor.

- Under the XM monitor, the DU handler (DUX) cannot be renamed and then used concurrently with the renamed handler. You cannot, for example, rename DUX to DAX and use DUX and DAX at the same time.

- DU bad-block replacement is supported under only the XM monitor.

- The distributed monitors do not support the command SET DUn: PORT=x. To generate support for that command, you must perform a SYSGEN and respond to question 108 (How many ports are to be supported (1) ?) with 2, 3, or 4.

LP/LS

- Issuing a PRINT command to the LP or LS handler while the printer is off line or XOFFed (LS only) causes the handler (and possibly the system) to appear hung. Place the printer on line or clear the XOFF condition, or execute CTRL/C twice to free the handler.

- DIGITAL recommends that you do not use the command SET LS NOHANG. When you use SET LS NOHANG, printers with very large buffers can abort before they are through printing.

NC, NQ, and NU

- You must explicitly load the NC, NQ, and NU handlers using the LOAD command. You cannot fetch those handlers.

TT

- Unit numbers other than 0 are not supported for TT devices. I/O requests to TTn, where n is other than 0, return a hard error.

VM

- The VM handler (under SJ and FB) cannot be used on a PDP-11/23 processor with MSV11 memory, strapped for a 2K word I/O page, unless you perform customization patch 2.7.21, described in the RT-11 Installation Guide.

XC and XL

- You should explicitly load the XC and XL handlers using the LOAD command. You should not fetch those handlers. If VTCOM is run as a system job, you must load them.

3.1.2 Monitor Restrictions

The following restrictions apply to RT-11 monitors.

- VECTOR addresses 470 and 474 are reserved for internal use by RT-11.

- FRUNing or SRUNing a privileged foreground job containing extended memory overlays causes the system to crash. Do not FRUN/SRUN privileged foreground jobs containing extended memory overlays.

- Privileged foreground jobs cannot use virtual overlays. Any attempt to do so can crash the system. (That does not apply to virtual foreground jobs; they run correctly with any type of overlay.)

- FORTRAN virtual arrays and VM using extended memory cannot be used concurrently when running under the SJ and FB monitors.

- If you pass more than one command through a special chain exit, you can call an indirect command file only as the last command in that series of commands.

3.1.3 Peripheral Device Restrictions

The following restrictions apply to the indicated peripheral devices.

RC25 Disk Drive

- When using an RC25 dual drive configuration (4 platters), turn on both RC25 drives before booting RT-11. Turning on the second RC25 drive after booting RT-11 requires a reboot to make the second RC25 available.

RQDX3 Disk Controller (DU)

- The RQDX3 disk controller cannot read data written to RD5x disks by the RQDX1 or RQDX2 controller. Therefore, if you upgrade from the RQDX1 or RQDX2 controller to the RQDX3 controller, you must reformat your RD5x disk. Perform the following procedure:

  1. Back up the disk written by the RQDX1 or RQDX2 controller using that controller.

  2. Format the disk using the RQDX3 controller.

  3. Restore your data to the disk.

TK50 Magtape Drive

- The TK50 magtape drive may not immediately recognize a change in the write protection status of a magtape cartridge when that cartridge is loaded in the drive. A change in magtape cartridge write protection is not recognized until a significant event occurs. A significant event would be, for example, a change in the magtape track being read or written.

  You can guarantee recognition of a change in write protection status by unloading the magtape cartridge, changing the write protection on that cartridge, and then loading it. You do not need to remove the magtape cartridge from the drive. Push the red load/unload switch (to unload), wait for the green light to come on, change the write protection on the magtape cartridge, and then push the red load/unload switch (to load).

- You may experience a delay in receiving the monitor prompt (.) when you abort operations on the TK50 magtape drive by issuing a double CTRL/C. The TK50 magtape drive might not relinquish control to the monitor until it finishes performing an operation that it independently initiated. Whether you receive the monitor prompt immediately depends on the operation being performed by the TK50 when you abort.

For example, when you instruct the TK50 to back up 10 or more records, it accomplishes that operation by rewinding to BOT (beginning of tape) and spacing forward to the desired record. If you abort that operation while the TK50 is rewinding to BOT, you will not receive the monitor prompt until the TK50 has reached BOT. However, if you specifically instruct the TK50 to rewind to BOT and then abort that operation, you receive the monitor prompt immediately.

Similarly, the TK50 reads calibration tracks as the first operation on a newly mounted magtape cartridge. If you abort the TK50 while it is reading those tracks, you will not receive the monitor prompt until the calibration is complete.

TS11, TSU05, TSV05, and TK25 Magtape Drives

- Issuing the INITIALIZE MS: command without a tape in the drive can crash the system.

TU58 DECtape II

- If your system load is so great that the TU58 DECtape II cannot be used successfully as the system device on a 38.4K baud line, change your hardware to lower the baud rate.

3.1.4 Processor Restrictions

The following restrictions apply to the indicated processors.

- A system generated to support DU multiport booting cannot be run on an SBC-11/21 (Falcon processor).
- When running RT-11 on an SBC-11/21, the SBC-11/21 must be jumpered for map selection 0, as specified in the SBC-11/21 Falcon User's Guide.
- LS, to be used on the available SLU serial port of an SBC-11/21 or SBC-11/21 PLUS processor, must be rebuilt with LS$PRI = 5.
- XL, to be used on the available SLU serial port of an SBC-11/21 or SBC-11/21 PLUS processor, must be rebuilt with XL$PRI = 5 or XL$SBC = 1.
- DD, to be used on the available SLU serial port of an SBC-11/21 or SBC-11/21 PLUS processor must be rebuilt with DD$PRI = 5.

3.1.5 Programmed Request Restrictions

The following restrictions apply to programmed requests

- The following requests and subroutines perform I/O contiguously from the specified buffer starting address: .READx, .WRITx, IREADx, IWRITx, JREADx, JWRITx, .SPFUN 376 and .SPFUN 377, ISPFN 376 and ISPFN 377. If the specified buffer spans a discontiguously mapped page boundary, I/O is performed outside of the mapped buffer, and no error message is returned.

3.1.6 Utility Restrictions

The following restrictions apply to the indicated RT-11 utilities.

BATCH

- BATCH does not work under the XM monitor when running virtual jobs that use the .GTLIN or .TTYIN programmed requests. An example of this is the CTS-300 compiler (DICOMP).

BUP

- BUP does not support 100 in/sec streaming for the TS05 magtapes under the XM monitor. BUP does support 25 in/s streaming for the TS05 magtapes under the XM monitor.

- Responses for the BUP system utility prompts must be entered at the terminal keyboard. DIGITAL does not support entering BUP prompt responses from command files.

- When you run BUP under the FB monitor, unload any unnecessary foreground jobs to gain more memory. That produces more efficient magtape streaming.

DBG-11

- Before you unload DBG-11, be sure to remove all breakpoints set in permanent areas of memory, such as the monitor and loaded handlers. Unloading DBG-11 before removing breakpoints in those areas could cause unpredictable results.

DUP

- The command COPY/DEVICE cannot be used by RT-11 V5 to copy images from magtapes to disks, when the images were copied to the magtape using RT-11 V4. Use RT-11 V4 for that operation.

- If you use the RT-11 V5 command INIT/VOL to assign a volume ID, that volume ID will not be displayed when you issue the command DIR/VOL under RT-11 V4.

GIDIS

- You cannot make any change in interlace mode while running GIDIS; any change must be made prior to running GIDIS or after you exit from GIDIS.

IND

- Entering an invalid device or file name for the IND .TESTFILE directive returns only a syntax error and terminates IND. IND cannot determine the exact cause of the error.

- IND control files cannot include an indirect command file that calls another IND control file.

- You cannot place CSI commands or DCL commands that require more than one line in IND control files. To execute multiline CSI commands and multiline DCL commands from an IND control file, create an indirect command file that contains the command, and call the command file from the control file by using the $@ syntax.

  In the case of single-line CSI commands, you can use CCL

commands instead of creating and calling the indirect command file. That saves space in the IND control file.

For example, the following IND control file executes a CSI command by creating an indirect command file, then calling the indirect command file:

```
        .IFF PIP .GOTO 1
        .OPEN SECOND.COM
        .DATA R PIP
        .DATA A.MAC=B.MAC
        .DATA ^C
        .CLOSE
        $@SECOND
.1:     .ASKS ...
```

Instead of creating the indirect command file, you can achieve the same result by using the following CCL command example:

```
.IFT PIP PIP A.MAC=B.MAC
```

- You cannot run the BATCH processor and IND simultaneously. You must have SET KMON NOIND in effect to run BATCH. Also, you cannot run BATCH from an IND control file.

LD

- When using recursive logical disks, the outer disk must have a lower unit number than the inner disk. For example, if logical disk B is located within logical disk A, A must have a lower unit number than B. Logical disk A can be LD0 and logical disk B can be LD1, but not vice versa.

  Similarly, you cannot mount a logical device located within the same logical unit. For example, the following mount command attempts (but fails) to mount logical disk LD3:FOO.DSK that is located within logical unit LD3:

```
MOUNT  LD3:  LD3:FOO.DSK
```

  However, the following command will work because it obeys the recursive logical disk rule:

```
MOUNT  LD2:  LD3:FOO.DSK
```

LINK

- The maximum address space of a program without virtual overlays is 65534(decimal) bytes or 32767(decimal) words. That is one word less than 32K words.

  The maximum virtual address space of a program with virtual overlays is 65472(decimal) bytes or 32736(decimal) words. That is 32(decimal) words less than 32K words.

MACRO

- In MACRO-11 programs, all variable names of the form ...Vx are reserved by DIGITAL.
- In MACRO-11 programs, all macro names of the form ...CMx are reserved by DIGITAL.

- Even when .ENABL MCL is in effect, you must manually .MCALL any macros whose names conflict with names in the MACRO-11 permanent symbol table (such as .PRINT).

QUEUE

- QUEUE protects files until it has placed them in the queue. It is a permanent feature of RT-11 that a file cannot be copied to a device that contains a protected file with the same file name. Therefore, QUEUE cannot be used to transfer a file to a device that contains a file with the same file name.

- When both SPOOL and QUEUE are running, QUEMAN options /H and /N return an invalid option error message. Those options conflict with SPOOL (PIP) options, and KMON assigns SPOOL precedence over QUEUE.

  Also when both SPOOL and QUEUE are running, PRINT options /PROMPT and /NAME return an invalid option error message. Those options are specific to QUEUE, and KMON assigns SPOOL precedence over QUEUE (KMON treats the options as SPOOL options).

RESORC

- SHOW MEMORY displays only approximate values. Memory components shown include memory tracking overhead locations, which are not part of the components. Consequently, the values shown can be as much as 32 words in error.

SETUP

- Set an LA210 or LA100 printer to AUTO mode before issuing it SETUP commands.

- Do not issue SETUP COLOR commands to monochrome monitors; such commands will overdrive your monitor, making the screen unreadable. Reboot the system to correct that condition. Do not include SETUP COLOR commands in command files that might be run on processors that have a monochrome monitor.

SL

- The single-line editor (SL) cannot be used with virtual jobs.

- When using the single-line (SL) editor, if you SET TT CRLF you must also set your terminal to wrap. Set your terminal to wrap using the SETUP TT WRAP command or use the terminal hardware setup feature.

- The single-line editor (SL) does not support the use of control characters as input data. Before you run a program that must receive control characters as input data in line mode, you must turn off SL. Use the SET SL OFF command, or have your program set the EDIT$ bit (bit 4) in the JSW.

  That restriction does not apply to programs that use special mode input.

- When the single-line editor (SL) is enabled, the command COPY TT: filespec does not work.

- Problems occur with program prompting with the single-line editor (SL) set ON. SL may not prompt properly when used with various combinations of .PRINT and .TTYIN or when the prompts contain control characters such as line feeds or carriage returns. No correction is available for that problem; however, the DIGITAL-recommended work-around is to SET SL KMON (rather than SET SL ON) or have your program set the EDIT$ bit (bit 4) in the JSW.

- SL and BATCH cannot be used together.

TRANSFER

- TRANSF.TSK (TRANSFER for RSX operating systems) requires that the RMSRES and DAPRES libraries be installed on the host PDP-11 processor.

- Run TRANSFER only from a host that you have logged onto using VTCOM. Attempting to run TRANSFER from a host you have logged onto using DECnet will hang the system. That is, do not use the SET HOST command to log onto a host and then attempt to run TRANSFER from that host (across the DECnet).

UCL

- If you define a UCL command as a save image (.SAV) file, the save image file returns an error message. You must first disable the CCL command processor because it attempts to run the UCL command definition as a CCL command. For example, defining the following UCL command produces the indicated error message:

```
.PIP :== R PIP
?CSI-F-Invalid command
```

  You can disable CCL command processing using customization patch 2.7.63 located in the RT-11 Installation Guide. Carefully study the instructions for using that customization patch because making a mistake can disable other or all command processors.

- If you define a UCL command as a valid DCL command, the keyboard monitor (KMON) returns an error message. KMON tries to execute the UCL command definition as a DCL command. Use UCF if you want to redefine a valid DCL command.

- The CSI (command string interpreter) interface is not supported for UCL; do not issue the R UCL command.

VBGEXE (unsupported)

The following restrictions apply to programs being run by VBGEXE:

- The program cannot contain interrupt service routines.

- The program cannot use the following programmed requests: .FORK, .INTEN, .MFPS, and .MTPS.

- The program can use only the .PEEK, .POKE, .GVAL, and .PVAL programmed requests to access the kernel address space (for example, monitor data and the I/O page).

VTCOM

- Some systems, such as those using MICOM lines, require more than one BREAK command to gain access to the host processor. The current version of VTCOM does not support multiple short BREAK commands.

3.1.7 Current Restrictions When Running FORTRAN under RT-11 V5.4

When running FORTRAN under RT-11 V5.4, observe the following restriction.

Use the following procedure when linking FORTRAN 77 programs, or linking FORTRAN IV programs when the OTS is not built for threaded code support.

You can experience undefined global symbol errors when linking FORTRAN programs that use the GETSTR.OBJ and PUTSTR.OBJ object files (modules). Those modules are located in SYSLIB, the system subroutine library, and require threaded code support in the FORTRAN OTS.

If you are using FORTRAN IV with nonthreaded code, or if you are using FORTRAN 77, and your program is calling GETSTR or PUTSTR:

Compile the distributed files GETSTR.FOR and PUTSTR.FOR, and replace the modules of the same name in SYSLIB with the created GETSTR.OBJ and PUTSTR.OBJ. Perform the following procedure:

```
.FORTRAN PUTSTR.FOR<RET>
PUTSTR

.FORTRAN GETSTR.FOR<RET>
GETSTR

.R LIBR<RET>
*SYSLIB=SYSLIB,PUTSTR.OBJ/R/G<RET>
Global? $OVRH<RET>
Global? <RET>
*SYSLIB=SYSLIB,GETSTR.OBJ/R/G<RET>
Global? $OVRH<RET>
Global? <RET>
*<CTRL/C>
```

Specify SYSLIB and FORLIB (in that order) when linking your program.

3.1.8 Current Restrictions When Using the Multiterminal Special Feature

The following restrictions apply to systems with the multiterminal special feature:

- Support of the DL11-W interface requires the presence of a REV E or later module. Without a REV E module, ECO (Engineering Change Order) number DEC-O-LOG M7856-S0002 must be applied to the M7856 module.

Support of the DLV11-J interface requires the presence of a REV E or later module. Without such a module, ECO M8043-MR002 must be applied to the M8043 module.

- The multiterminal handler can support remote terminals. Modem control is available for both DL11-E and DZ11 interfaces. The DL11 control answers ring interrupts, permitting terminals to dial in to the system. Dial-in is possible with the DZ11 interface, despite lack of a ring interrupt in the DZ11, if the modem is operated in auto-answer mode. This is achieved through a polling routine that periodically checks the status of each line on the multiplexer (see Chapter 5 of the RT-11 Software Support Manual). Dial-up support for DZ interfaces requires BELL 103A-type modems with "common clear to send and carrier" jumpers installed. With this option installed, the modem operates in auto-answer mode.

- The hardware console interface must be a DL interface, and it must be a local terminal. You can use the SET TT: CONSOL command to move the background console to any other local terminal in the system.

- The number of DL interfaces RT-11 supports, both local and remote, is limited to eight. This number includes the hardware console interface.

- The number of DZ11 controllers is limited to two, for a total of 16 lines. The total of DZV11 controllers is limited to four, for the same total of 16.

- The VT11 scroller option is disabled when the multiterminal special feature is present in a system. The commands GT ON and GT OFF are not valid in multiterminal systems. For this reason, EDIT cannot use the display support. The use of graphics is still supported, though, and the display support in TECO works as well.

- The maximum input data rate for a single terminal is 300 baud. The aggregate total input data rate for a system is 4800 baud.

  You can set the output baud rate to any speed; RT-11 sends output as fast as possible, depending on the capacity of the CPU and the nature of its load.

- When you type double CTRL/C in an SJ system, the monitor does a hardware RESET instruction. This causes the DZ multiplexer to reset its status and to drop Data Terminal Ready on all lines, thus hanging them up. This action is part of the general cleanup the system performs after a program aborts.

- If you plan to devote a terminal line to the DD, LS, or XL handler, do not specify that terminal's DL interface in the SYSGEN dialog for a multiterminal system. Do not attempt to attach the terminal from a multiterminal application program, either.

- Setting the baud rate, character length, number of stop bits, and parity by way of the .MTSET programmed request is supported only for DZ interfaces.

3.2 CURRENT RESTRICTIONS FOR PROFESSIONAL 300 SERIES PROCESSORS ONLY

When running RT-11 V5.4 on Professional 300 series processors, observe the following restrictions.

- Smooth screen scrolling is not supported for Professional series processors. Do not issue the command SETUP SMOOTH if using a Professional series processor.

- Professional 300 series processors only support multiterminal monitors that have SYSGEN support for one terminal. Do not attempt to run a multiterminal monitor on a PRO that has SYSGEN support for more than one terminal.

- You must UNPROTECT the PI handler (PI.SYS or PIX.SYS) before executing the SETUP RETAIN command.

- The terminal console display can be changed when changing from one national replacement character (NRC) set to another.

- PRO/GIDIS support is not available under the FB monitor.

- DIGITAL recommends that existing programs being transported to Professional 300 series processors be relinked as virtual jobs to be run under XM. They should use both virtual overlays and virtual .SETTOP as required to minimize low memory use. A useful technique is to create a dummy root that calls the real main program linked as virtual overlay segment 1. That procedure is usually necessary when attempting to run large programs that formerly executed under the SJ or FB monitor.

  If you cannot relink your programs, an alternative technique uses the unsupported virtual run utility VBGEXE to create a simulated SJ/FB environment that is sufficient to execute many programs without relinking. For more information, refer to the RT-11 System Utilities Manual.

- Due to possible contention with the KT-11, split-screen scrolling under the FB monitor refreshes the screen on each line scroll. DIGITAL recommends using the XM monitor on Professional 300 series processors.

- SETUP DATE,TIME should be issued after the first hard boot to set the time-of-year clock. Otherwise, subsequent boots will assume the clock is valid because the battery will have had time to charge. That command is included in the automatic installation procedure by default.

- On Professional series computers, bootstrapping the system causes a terminal restart; that is, whenever you bootstrap the system, terminal status reverts to the last information saved by SETUP.

- Due to memory limitations, the 132-column mode is not available under the FB monitor.

- PRINT SCREEN can be used only under the XM monitor.

- The SJ and AI monitors are not supported.

- ASCII 8-bit supplementary code is not supported.

- Professional 300 series and Q-BUS processors can read and write the same RX50 diskettes. However, bootable system volumes are not interchangeable. A bootable system diskette written for the Professional 300 does not boot a Q-BUS processor.

- You cannot use ODT (on-line debugging technique) on Professional series processors. Use VDT (virtual debugging technique) instead.

- Only one RX50 controller module can be used. No additional RX50 diskette drives can be added.

- FILEX is not supported.

- The BOOT/FOREIGN command is not supported.

- SETUP NOCLICK does not work after a hard boot when issued from a start-up command file.

- No user input/output (write/read) operation can be addressed to the PI (Professional interface) handler; attempts to do so cause a hard error. PI must be present on the system disk and loaded at all times.

- Professional 300 series processors do not support split-screen smooth scrolling.

- RT-11 V5.4 on Professional series processors does not support the following VT100 control character codes, modes, character sets, and tests, adjustments and reports:

  Control Character Codes:

  ETX (end of text)

  EOT (end of transmission)

  Modes:

  DECPEX (printer extent mode)

  DECPFF (printer form-feed mode)

  SRM (send-receive mode)

  MC (auto print mode)

  All VT52 modes

  Character Sets:

  Alternate ROM character set

  Alternate ROM special character set

  Tests, Adjustments and Reports:

  All DECTST (device diagnostic tests)

  ECLL (load LED)

  DSR (printer)

3.3 CORRECTED PROBLEMS

The following problems in previous versions of RT-11 have been corrected in RT-11 V5. Change bars indicate problems corrected after publication of the RT-11 V5.3 Release Notes (AA-5286H-TC) and for RT-11 V5.4.

3.3.1 Automatic Installation

The following problem in automatic installation has been corrected:

- If you requested no SYSGEN when running CONFIG.COM, SJFB.ANS remained on the configured disk. CONFIG.COM now deletes SJFB.ANS from the configured disk if you request no SYSGEN.

3.3.2 Monitors

The following monitor problems have been corrected.

- Previously, after initial start-up (cold boot), a monitor built for memory parity support running a program that read locations not initialized by the bootstrap could cause a memory parity error trap. Now, the bootstrap performs a read/write operation on all locations, including those below the 28K-word boundary, during the memory sizing operation. That corrects parity for all memory locations.

- The system no longer halts when a monitor without memory parity support is booted on a KDJ11-A and a cache parity error occurs.

3.3.2.1 **KMON** - The following problems with the keyboard monitor have been corrected:

- When KMON processed an indirect command file containing multiple indirect command files, it saved the command file depth status before determining if all nested indirect command files existed. If one of the nested indirect command files did not exist, and error was set NONE, KMON reported the nonexistent file and then processed the next file twice. Now, KMON correctly saves the command file depth status while reporting the nonexistent file and then properly restores it. The next indirect command file is correctly processed only once.

- Previously, a virtual job could not pass information when chaining to another job. Also, you could not chain to a virtual job. Now, a virtual job can pass information when chaining to another job, and you can chain to a virtual job.

- The keyboard monitor RUN command no longer automatically loads locations 500-777 of the save image. The Chain Bit (bit 8 of offset 44 in block 0) must now be set to do that. Programs such as MUBASIC that do not set that bit but require that the save image be loaded, must be run using the R command. The R command loads those programs properly.

- When the EXECUTE command was issued and the device was not resident, KMON returned the error message ?KMON-F-File not found FORTRA.SAV. KMON now returns the error message ?KMON-F-Handler must be loaded <DEV:>.

- The EXECUTE command when issued for a device and file name, but no file type, causes KMON to first search for .FOR, then .MAC, then .DBL file extensions. If no filespec is found with those extensions, KMON previously returned the error message ?KMON-F-File not found FORTRA.SAV. KMON under those circumstances now returns the error message ?KMON-F-File not found DEV:FILENAM.TYP.

- Previously, nested indirect control files and nested indirect command files that contained a CTRL/C could skip the command following the CTRL/C. Control could revert to the calling file before the nested file finished processing. That could happen to indirect control files nested within indirect command files and indirect command files nested within indirect control files. Now, nested files correctly finish processing under those circumstances.

- Specifying a unit number for SL with the UNLOAD command (UNLOAD SLn) incorrectly let you unload SL while it was running. That could crash the system. Now, KMON checks if SL is running before executing the UNLOAD SLn command, and correctly prohibits execution of that command if SL is running.

- Lowercase characters within a command file are no longer converted to uppercase if the program receives the command line by issuing a .GTLIN request with the lowercase bit set.

- Virtual jobs now correctly initialize synchronous system traps.

- The user stack in KMON containing context switching information was being written-over by a background job overlaying KMON, or a background job in the process of overlaying KMON was being written-over by the context information for a system or foreground job. In either case, the background job's context information or the background job itself was corrupted. Now, the user stack points to the transitional stack in RMON before doing the read operation that overlays KMON. The background job's context information and the job itself are not corrupted.

- When KMON parsed a file name longer than six characters, KMON printed the error message ?KMON-F-Command file not at end of line. In RT-11 V5, KMON prints the more appropriate message ?KMON-F-Error in file spec.

- With previous versions of RT-11, the device had to be loaded prior to chaining to another file. In RT-11 V5, devices need not be loaded.

- The command DELETE * now expands to the command DELETE *.* rather than the command *.NULL.

- When you omitted an argument from an option that required one, a trap to 4 occurred. In RT-11 V5, when you omit a required argument, the trap to 4 does not occur and KMON prints the error message ?KMON-F-Invalid value specified with option.

- If you performed a chain exit when running under a monitor with system job support, vector areas 472-476 were destroyed. Those vector areas are now preserved.

- Typing digits in a command name caused KMON to print the error message ?KMON-F-Ambiguous command. That message is appropriate for commands that contain alphabetic characters only. In RT-11 V5, using digits rather than characters in the command generates the error message ?KMON-F-Invalid command.

- In RT-11 V5, KMON prints the error message ?KMON-F-Invalid set parameter, rather than ?KMON-F-Illegal command, if the handler specified in the command rejects parameters.

3.3.2.2 **MTTINT** - The following problems with the multiterminal interrupt routine module have been corrected:

- RT-11 rejected (did not answer) incoming calls to the shared console that was SET to a remote DL line unless a program was running that had attached to that DL line. RT-11 now accepts (answers) incoming calls under those circumstances.

- Generating a multiterminal XM monitor with BATCH support but without system job support no longer causes a branch out-of-range condition. Previously, this error occurred during the phase of SYSGEN that builds the multiterminal interrupt routine module (MTTINT).

3.3.2.3 **RMON** - The following problems with the resident monitor have been corrected:

- The SJ monitor now correctly uses the .EXIT request to reset the .SFPA request address. That correctly restores the default FPU (floating point unit) trap handling mechanism.

- An SJ monitor that included SYSGEN-option timer support sometimes hung in a completion routine during heavy I/O use. Priorities were not correct during execution of the clock routines, and therefore, the accumulation of clock ticks began at the wrong value. Now, priorities are correctly set, and the accumulation of clock ticks begin at the correct value. The SJ monitor no longer hangs during heavy I/O use.

- The .ABTIO request now aborts only I/O queue elements belonging to the specified channel of the issuing job. Also, .ABTIO no longer loses any of the queue elements that it aborts.

- Programmed requests .FPROT and .SFDAT now work when issued from virtual jobs. The resident monitor no longer fails to relocate the address of the argument block.

3.3.2.4 **USR** - The following problems with the user service routines module have been corrected:

- The .CSISPC programmed request incorrectly closed channels 0, 1, and 2 when it found an error in a command received from the terminal or an indirect file. For RT-11 V5.4, the .CSISPC programmed request correctly does not modify any channels.

- When running the IND control processor under all monitors, any .GTLIN prompt from a program to the terminal was suppressed. Now, a .GTLIN prompt to the terminal from a program running under IND is not suppressed.

- Under the SJ monitor, the .GTLIN programmed request did not echo user input when TTSCP$ was set. The .GTLIN request now echoes user input when TTSCP$ is set.

- When an indirect command file contained blank lines for utility version numbers, the version numbers were not always displayed. Version numbers are now displayed.

- Files created on a device were not always allocated the smallest empty space for the file length requested. The smallest empty space for the file length requested is now allocated.

3.3.3 Utilities

The following problems have been corrected for utility programs.

3.3.3.1 **BINCOM** - The following problems with the binary comparison utility have been corrected:

- BINCOM could corrupt data in its internal buffers before comparing it when the /END:n option was used and the following conditions applied:

  The shorter of two unequal length files being compared was of size n.

  The BINCOM internal buffers were a particular multiple of the shorter file segment being compared.

  That problem is corrected in RT-11 V5.4.

- When a wildcard binary comparison between files on two devices was directed to an output file using a command such as DIFFERENCES/BINARY/OUTPUT:TEST.DIF dd:*.* dd:*.*, BINCOM wrote the output file only when the last two files compared contained differences. No output file was written even if files (other than the last two) contained differences. Now, BINCOM correctly writes the output file when any files on the devices are different.

- When using BINCOM to create a SIPP command file, the flag to strip the leading zeros was cleared at an incorrect time. That caused significant zeros in addresses higher than 7776 to be lost in the created SIPP command file. BINCOM now correctly clears the flag and the SIPP file is correctly created.

- When comparing two diskettes, a bad block on one diskette was treated as end-of-device by BINCOM. That resulted in the other diskette being reported as longer. Now, BINCOM distinguishes the EOF mark from a bad block. The problem is corrected.

- When a comparison was being performed and one or both of the files or volumes being compared resided on a diskette, BINCOM generated a hard error message if it reached the end of the diskette. In RT-11 V5, BINCOM returns to the keyboard monitor when it reaches the end of the diskette.

3.3.3.2 **BUP** - The following problems with the backup utility program have been corrected:

- Previously, when a diskette was incorrectly inserted in a drive or a magtape was off-line during a backup operation, BUP issued a fatal-level error message and aborted the operation. Now, under those circumstances, BUP issues an error-level error message and prompts you to mount another volume in the drive.

- When performing file backup operations, V5.2 BUP incorrectly wrote a 1280.(decimal) block output file regardless of the input file size. That was true only for V5.2 file backup operations. Device backup operations under V5.2 performed as documented. For RT-11 V5.3, BUP correctly backs up files and devices as documented. See the V5.3 file V5NOTE.TXT for complete information.

- A copy operation could corrupt the output file when the number of blocks of the file buffer in memory was not a multiple of the output tape transfer block size. Now, the file buffer in memory need not be a multiple of the tape transfer size for BUP copy operations.

- BUP (or the BACKUP command) could not be used to back up a disk to magtape if the disk contained bad blocks covered with FILE.BAD files. Now, that restriction is removed. However, performance decreases if BUP encounters bad blocks during a backup from disk to magtape.

- When running under the XM monitor, BUP was restricted to a single output volume. Now, backup is no longer restricted to a single volume.

- When backing up an RL02 to a file-structured TSV05, BUP gave an incorrect BUP-F-Read error message, although the transfer was completed correctly. Now, BUP no longer gives that incorrect error message.

- When backing up to a second RL02 disk, BUP gave an incorrect error message ?BUP-W-Incompatible output volume size. Now, BUP no longer gives that incorrect error message.

- When running under the XM monitor, BUP gave an incorrect EOT message when backing up to multiple tape volumes. Now, BUP no longer gives that incorrect EOT message.

3.3.3.3 **DIR** - The following problems with the directory utility have been corrected:

- When you omitted a portion of the date in the DIRECTORY/SINCE[:date] command, DIR did not correctly append values from the current system date. DIR now correctly appends current system date values for any you omit in that command.

- DIR no longer prints the incorrect ?MON-F- output error message prefix, but instead prints the correct ?DIR-F- prefix when the output device is write-locked.

3.3.3.4 **DUMP** - The following problems with the DUMP utility have been corrected:

- The DUMP/ONLY:n DM0: command previously caused the error message, ?DUMP-F-Invalid option /E, to be displayed when the desired block number was less than 32768(decimal). The DUMP/ONLY:n DM0: command now works properly regardless of the block number.

- When a diskette was being dumped, DUMP generated a hard error message when it reached the end of the diskette. In RT-11 V5, DUMP returns to the keyboard monitor when it reaches the end of the diskette.

3.3.3.5 **DUP** - The following problems with the device utility program have been corrected:

- For RT-11 V5.3, DUP would only soft boot processors that contained the KT11 memory management unit. For RT-11 V5.4, DUP does not require the KT11 memory management unit to perform a soft boot.

- DUP did not append a wildcard file specification (*.*) to a CSI command if you omitted it in a CSI command line that contained the /F and /I options (dd:file.ext/F=dd:/I). Therefore, DUP returned the error message ?DUP-F-Invalid command. Now, DUP appends the wildcard file specification if you omit it in a CSI command line that contains the /F/I options.

- If you wanted to create a file using the CREATE/ALLO:nnn command and there was insufficient space to create the file, the DUP utility displayed the error message ?DUP-F-Conflicting SYSGEN options. DUP now displays the error message ?DUP-F-No room for file.

- DUP no longer traps to 4 (odd address/bus timeout trap vector) when you initialize a volume with bad data in the extra bytes field of the directory header.

- Using the /DEVICE/FILE/WAIT options combination (DUP/I/F/W) in a COPY command is now supported, and no longer returns a error message ?DUP-F-Invalid command.

- The BOOT/FOREIGN (DUP/Q) command occasionally did not work because DUP did not properly simulate a hardware bootstrap. DUP now properly simulates hardware bootstraps for all supported devices.

- The CREATE/ALLOCATE:-1 (filespec[-1]=/C) command incorrectly entered a file size of 65535 blocks in a device directory, regardless of the actual amount of available space. CREATE/ALLOCATE:-1 now correctly enters the maximum space available in the device directory.

- When several DUP options were specified with the /WAIT (DUP /W) option, DUP prompted to mount the input volume once for each operation requested. In RT-11 V5, the prompt to mount a volume is printed only once.

- When using the command INIT/BAD/WAIT ddn: (DUP/B/W), where ddn is the device in which the system disk resides, DUP did not prompt to mount the input volume a second time for the initialization operation. Therefore, DUP initialized the system volume instead, destroying the boot blocks. DUP now correctly prompts you to mount the input volume.

- When using the BOOT/WAIT command, DUP printed the prompt ?DUP-W-Foreground loaded. Are you sure? twice. In RT-11 V5.1, the prompt is printed only once.

3.3.3.6 **Error Logger** - The following problem with the error logging utility has been corrected:

- The error logger did not report the correct time of error. The time of error is now reported correctly.

3.3.3.7 **HELP** - The following problems with the on-line HELP utility have been corrected:

- The HELP/PRINT topic command no longer causes the system to hang.

- Previously, the command HELP ? returned an incorrect error message. Now, that command returns the error message ?HELP-F-HELP not available for ?.

3.3.3.8 **IND** - The following problems with the indirect control file processor have been corrected:

- Previously, when the .TESTDEVICE or .VOL directive encountered some device read-related errors, IND would exit with a fatal error condition. That happened even if you included the .ONERR directive in your code, because the .ONERR directive attempted to access the <FILERR> special symbol and .TESTDEVICE and .VOL did not support <FILERR>. Now, the .TESTDEVICE and .VOL directives support the <FILERR> special symbol. You can perform error recovery in your code that prevents IND from exiting due to read-related error conditions.

- The .TESTDEVICE directive incorrectly reported the status of the SPOOL device handler SP as OFL (off line). It now correctly reports the status as UNK (unknown).

- The .TESTDEVICE directive incorrectly reported the status of all logical disks as OFL (off line), whether a logical disk was mounted or not. That was caused by a change made to the LD assignment table format for RT-11 V5.3 that was not recognized by .TESTDEVICE. For RT-11 V5.4, .TESTDEVICE correctly reports the status of logical disks.

- IND trapped to 4 when the .TEST directive was used on a valid string symbol. That occurred because of invalid pointers during the .TEST directive processing. Now, the .TEST directive no longer traps to 4 when it is used with a valid string symbol.

- IND aborted with a fatal error when a read error occurred while the .STRUCTURE directive was determining a device file structure. Now, the .STRUCTURE directive is modified to return an 'UNKNOWN' (code 0) structure if a read error occurs. DIGITAL continues to recommend that you issue the .TESTDEVICE directive before the .STRUCTURE directive to make sure a device is on line and mounted.

- Executing a control file using the DCL keyboard command syntax RUN IND ctrl-filespec[/options] or IND ctrl-filespec[/options] did not let you pass parameters. Those commands now let you pass parameters.

- IND now correctly handles symbol names of more than six characters.

- IND contained a restriction that blank and tab characters could not be placed between operators and numeric symbols. IND no longer contains that restriction, so numeric expressions can be formatted for better readability.

- The .TESTDEVICE directive no longer returns an incorrect device size in <EXSTRI> when reporting the status of a device having one size.

- Previously, the .TESTDEVICE directive returned the error messages, ?IND-F-Device error, and ?IND-F-Device read error, if you used the KMON command LOAD ddx:=F or LOAD ddx:=job to attach to a foreground or system job. Now the .TESTDEVICE directive includes ATT or NAT (attached or not attached) in field nine of special symbol <EXSTRI>, and the two error messages are not displayed.

- Previously, there was a problem with using the .VOL directive if it was issued for a device assigned to the foreground or a system job using the LOAD ddn:=F or LOAD ddn:=job commands. That operation returned the error message ?IND-F-Date file error, Invalid device or unit. Now, the .VOL directive returns the error message ?IND-F-Invalid device or unit, device is attached. That error can be intercepted using the .ONERR directive, and you should use the .TESTDEVICE directive to determine if the device is indeed attached and to what job.

3.3.3.9 **KED** - The following problems with the keypad editor have been corrected:

- The FIND command now works for most control characters, except ^B, ^C, ESCAPE, ^F, and ^X.

- The LOCAL command now works when applied to a range containing a form-feed (<FF>) character.

- KED displayed an I/O error message when it reached the end of an inspected disk. KED now displays the end-of-file symbol.

- The KED chain interface now works properly.

3.3.3.10 **LIBR** - The following problems with the librarian utility have been corrected:

- When you incorrectly coupled the /UPDATE option with the output file name, the error message returned by LIBR incorrectly included the input file name in the error message. Now, LIBR correctly includes the output file name in the error message.

- Under certain circumstances, LIBR generated macro libraries where the last entry would not appear in the library directory. That normally occurred when adding to SYSMAC or HELP, and no LIBR error message was displayed. Now, all entries into macro libraries appear in the library directory.

- The LIBR utility failed if a forms library directory exceeded one block. LIBR now allows forms library directories to exceed one block.

3.3.3.11 **LINK** - The following problems with the linker have been corrected:

- RT-11 V5.3 removes the restriction stating that the total combined size of all virtual overlay segments created by the LINK /V option cannot exceed 96K words. Now, the total combined size is the amount of available memory.

- Previously, when you linked programs using the /DUPLICATE option, LINK occasionally returned the error message, ?LINK-F-Invalid RLD symbol in DEV:FILNAM.OBJ, and prematurely aborted. For RT-11 V5.4, the problems that caused that are corrected.

- Previously, when you linked overlaid programs using the /INCLUDE option, LINK occasionally returned the error message, ?LINK-F-Internal error, and prematurely aborted. For RT-11 V5.4, the problems that caused that are corrected.

- The /DUPLICATE option incorrectly repeated absolute symbols in the link map for each duplication of a library module containing the absolute symbol. Now, only the defined global symbols are entered in the link map.

- In some cases when you used the /U option, LINK displayed the error message ?LINK-F-Address space exceeded, even though the program's high limit was less than 32K words. That problem was caused by the improper initialization of a variable, and has been corrected.

- In some cases when you linked valid object modules, LINK displayed the error message ?LINK-F-Invalid GSD in DEV:FILNAM.TYP. That problem was caused by code that processes transfer address entries in the global symbol directory (GSD), and has been corrected.

- When absolute binary (.LDA) files were linked with ODT or VDT, the load address of the .LDA file was placed such that it was written over and corrupted by the ODT or VDT module declaration macro. That produced an incorrect link map. Absolute binary files now link with ODT or VDT at the correct load address (above ODT or VDT), and produce an accurate link map. When preparing Software Performance Reports (SPRs), include the version information from ODT/VDT that is found in the link map as .AUDIT, and .ODT or .VDT symbol values.

3.3.3.12 **PIP** - The following problems with the peripheral interchange program have been corrected:

- When a hard error occurred on a magtape, such as an attempt to copy to a write-protected magtape, PIP incorrectly trapped to 4 or 10. Now, PIP correctly returns an error message indicating a hard error condition.

- The COPY/SETDATE command, when the input and output filespecs were the same, incorrectly moved the specified input file while changing the file date. That is, the command actually performed a COPY operation. For RT-11 V5.4, the COPY/SETDATE command only changes the file date when the input and output filespecs are the same, and does not perform a COPY operation.

- PIP incorrectly treated printers (LP) as sequential access devices for copy operations from the terminal (TT). Therefore, a COPY command from TT to LP produced an invalid device error. Now, PIP correctly copies from the terminal to the printer.

- When the system device (SY) and the data device (DK) were assigned to the same device, a copy operation from SY to DK incorrectly performed the copy operation twice. Now, PIP renames the file to the output file name rather than copying it.

- The COPY/ALLOCATE:size command created a file with the current date rather than the input file date. Now, that command correctly uses the input file date as the creation date for the output file.

- When the COPY/MULTIVOLUME/NOREPLACE (PIP /E and DUP /R) command was issued, PIP processed the first output volume correctly. However, when the second output volume was mounted, PIP copied the next file from the input volume to the second output volume regardless of whether the file already existed on that output volume. PIP was not reading the directory of the second output volume before beginning the copy operation. PIP now reads the directory of each volume immediately after it is mounted, and the /MULTIVOLUME/NOREPLACE options work correctly.

- When COPY/PREDELETE (PIP /O) was performed and the specified input and output volumes were the same, PIP deleted the file. In RT-11 V5, COPY/PREDELETE performs a RENAME operation when the input and output volumes are the same.

- When wildcards were used in a RENAME/NOREPLACE (PIP /N) operation, the first file was found but not renamed, and an error message appeared. RENAME/NOREPLACE now works as documented in the RT-11 System User's Guide and RT-11 System Utilities Manual.

- When the /WAIT option (PIP /E) was specified, PIP attempted to read the output volume's directory before it prompted to mount the output volume. Therefore, an error occurred (if no volume was present in the output device) or the wrong volume could be read. PIP now prompts you to mount the output volume before attempting to read its directory.

3.3.3.13 **QUEUE** - The following problems with the QUEUE utility package have been corrected:

- QUEUE no longer appends a carriage return to the end of files that are queued to VM.

- QUEMAN displayed an error message when you issued the 2-slash (//) line terminator option on a line by itself. QUEMAN now accepts the 2-slash line terminator option on a line by itself.

- QUEUE did not recognize and correctly print asterisk (*) and percent (%) characters used in job names on banner pages. QUEUE now correctly handles asterisk and percent characters for banner pages.

3.3.3.14 **RESORC** - The following problems with the RESORC utility have been corrected:

- Previously, when global SCCA was enabled, a portion of the SHOW CONFIG data was displayed twice. That has been corrected.
- The SHOW DEV:DU command now displays the DU UNIT, PARTITION, and PORT translation table as illustrated in the RT-11 System Utilities Manual. See also Section 2.27.
- When more than 14 logical names were assigned to a device, the SHOW command (RESORC) printed a corrupted listing. More than 17 logical name assignments to a single device caused RESORC to hang the system. RESORC now correctly prints listings, and more than 17 logical device name assignments to a single device no longer hang the system.
- RESORC now correctly displays the PRO ROM ID.

3.3.3.15 **SETUP** - The following problems with the hardware characteristics SETUP utility have been corrected:

- SETUP NOCOMPOSE now works as documented in the RT-11 System User's Guide.
- The SETUP CAPS and SETUP SHIFT commands now set the appropriate bits in the internal table kept in PI. After these commands are issued, SETUP SHOW correctly indicates the commands.

3.3.3.16 **SPOOL** - The following problem with the spooler utility package has been corrected:

- If a SETUP PRINTER command was issued while SPOOL was running, the system would hang or trap to 4. Now, issuing the SETUP PRINTER command while SPOOL is running does not hang or trap the system.

3.3.3.17 **SRCCOM** - The following problem with the ASCII file comparison utility has been corrected:

- When the DIFFERENCES /CHANGEBAR (SRCCOM /D) option was specified with the console as the output device, the message ?SRCCOM-I-No differences found, would overwrite the end of the console output. That message is now printed on a new line.

3.3.3.18 **TRANSF** - The following problems with the binary transfer utility (TRANSF.SAV) have been corrected:

- TRANSF now forces the terminal to NOCRLF when you run RT-11 under a nonmultiterminal monitor.
- If you type a CTRL/O during a file transfer, the CSI prompt character is now displayed when the transfer is completed.

3.3.3.19 **UCL** - The following problem with the user command linkage utility has been corrected:

- The UCL data file (UCL.DAT) was occasionally corrupted before the maximum number of commands were entered. The corruption occurred because the data file held only 256(decimal) characters for the command definition field, and that number was exceeded before the maximum number of commands was entered. Now, a customization patch lets you change the size of the UCL data file to any maximum size you specify.

3.3.3.20 **VTCOM** - The following problems with the virtual terminal communications utility have been corrected:

- Previously, VTCOM would hang when it received an XOFF from the host, and you then sent an XON (CTRL/Q) to the host from the VTCOM command prompt. You could not abort VTCOM and would need to reboot your system. Now, VTCOM does not hang under those circumstances.

- The EXIT and PAUSE commands now close any open logging file before exiting to the monitor.

3.3.4 Device Handlers

The following device handler problems have been corrected.

3.3.4.1 **DL** - The following problems with the DL handler have been corrected:

- The DL handler no longer occasionally causes a system crash when used under a monitor with error logging support on a system with an RLV12 controller.

- When a disk reported certain hard errors to the DL handler, the handler would reverse seek one track and then reseek to the desired track. The DL handler now homes to track 0 and then reseeks to the desired track. That improves the chance for recovery from the hard error.

3.3.4.2 **DU** - The following problems with the DU handler have been corrected:

- Previously, SPFUN 372 could only read the handler translation table. Now, SPFUN 372 writes and reads the table.

- The DU handler installation procedure did not check for port validity when installing a DU device unit that was SYSGENed for multiple ports. Attempting to access a DU device unit that was mapped to an invalid port crashed the system. Now, that returns an error message and does not crash the system. Use the SHOW DEV:DUn command to determine if the port mapped to a DU device unit is valid. An invalid port is indicated by an asterisk (*) displayed before the port number.

- DU did not correctly check and truncate non-file-structured .WRITE requests that crossed partition boundaries. That could cause data that was written across a partition to corrupt data already written in the next partition. DU now correctly checks and truncates requests that cross a partition boundary. The amount of data actually transferred is not reported to the program requesting I/O, but a hard error is indicated.

- After you boot an SBC-11/21 PLUS processor, the SHOW DEVICE command displays the standard DU CSR (172150), not the DU CSR (176150) used by the SBC-11/21 PLUS. If your CSR address is 176150, DIGITAL recommends you use the SET DU CSR=176150 command after booting your system for the first time. That command enables RESORC to display the correct CSR when you issue the SHOW DEVICE command.

- Volume sizing code reported incorrect information when sizing a nonexistent MSCP unit number. Incorrect information was also returned on some subsequent size requests. The DU handler now returns correct sizing information under those circumstances.

- Errors produced an excessive number of retries because the retry count was incorrectly initialized. The retry count now initializes correctly.

- SPFUN 360 and 371 (special function bypass) were restricted in use only to privileged jobs because the response and command buffer was required to be in the low 28K words of memory. The response and command buffer is no longer required to be in the low 28K words of memory; that restriction is removed.

3.3.4.3 **DW** - The following problems with the DW handler have been corrected:

- Previously, you could specify unit (n) numbers 0 and 1 in the SET DWn [NO]WRITE command. Now, you can only specify unit number 0. An omitted unit number defaults to unit 0.

- Previously, the DW handler applied the wrong mapping of logical block numbers to physical block numbers on RD5x hard disks. The correct mapping is used in RT-11 V5.3. See the DW section in Chapter 2 for information. See the ISPFN and .SPFUN sections in Chapter 2 for changes to the blk argument caused by the DW remapping.

3.3.4.4 **DY** - The following problem with the DY handler has been corrected:

- An overflow of the SET table area of the RX02 handler occurred if the handler was built to include second controller and error logger support. The overflow no longer occurs.

3.3.4.5 **LD** - The following problems with the LD handler have been corrected:

- Previously, start-up command files aborted if they attempted to access a logical disk that was mounted on a noninstalled device. LD returned the fatal error severity message, ?LD-F-Device not installed DEV:. Now, the severity of the error message returned by LD under those circumstances is reduced to a warning, ?LD-W-Device not installed DEV:. You receive the warning message and the start-up command file no longer aborts.

- Previously, you could not customize the LD handler name suffix; you had to use a suffix of X or M. Now, you can customize the LD handler suffix character. You must perform customization D.3, Changing the Handler File-Name Suffix, located in the RT-11 System Generation Guide. You must also perform the customization described in Section 2.5.9.2 of this manual to make that suffix character known to the monitor bootstrap.

- Previously, a fetch error occurring during an operation requiring LD to fetch a handler could corrupt LD's internal table. Now, a fetch error occurring under those circumstances correctly returns the error message ?LD-F-Fetch error.

- Previously, the error message ?LD-F-No file specified <DEV:>, printed extraneous characters. It no longer does that.

- Monitors built with nonstandard suffixes caused the SET LD CLEAN and SHOW SUBSET commands to not work properly. Those commands now work properly with monitors built with nonstandard suffixes.

- Error messages that contain an option or an option with a value are no longer corrupted.

- LD no longer does an odd address reference (trap to 4) when it encounters an unknown option letter.

- LDX and LDM no longer print a corrupted version number.

3.3.4.6 **LS/LP** - The following problems with the LS and LP handlers have been corrected:

- CTS-300 SYSGEN now builds the LS handler for Professional 300 series processors.

- A coding error in the handler sometimes caused loss of characters when you ran RT-11 on a Professional series machine. That problem has been corrected.

3.3.4.7 **NC/NQ** - The following problem with the Ethernet handlers NC and NQ has been corrected:

- A problem occurred with processing queue elements on the internal receive queue. Depending on the number of receive requests outstanding and the order in which they were satisfied, queue elements could become lost, causing the system to hang. For RT-11 V5.4, queue elements are correctly processed on the internal receive queue.

3.3.4.8 **PI** - The following problems with the Professional Interface (PI) handler have been corrected:

- An error in the secondary device attribute escape sequence processing has been corrected.
- The combination of SETUP WRAP and SETUP SMOOTH no longer causes partial display of the first character of a wrapped line.
- Horizontal tabs now work properly on a double-wide line.
- On Professional 300 series processors, the escape sequence <ESC>[0;0r (set top and bottom margins of scrolling region) was treated as invalid and ignored. That escape sequence is now processed as <ESC>[1;24r in accordance with the VT100 standard.

3.3.4.9 **RK** - The following problem with the RK handler has been corrected:

- Previously, the RK handler did not account for the hardware differences between RKV11 and RK11 controllers regarding DMA transfer limits. Now, the RK handler returns a hard error if a transfer would have accessed memory beyond the limits addressable by the controller in use, that is, 64K bytes on the Q-BUS and 256K bytes on the UNIBUS.

3.3.4.10 **TT** - The following problem with the TT handler has been corrected:

- When running under the SJ monitor, the TT handler prompted with an up arrow (^) for every .READ request rather than correctly prompting only when the .READ request specified block 0 as the starting block. Now, the TT handler prompts only for a block 0 .READ request.

3.3.4.11 **VM** - The following problems with the virtual memory (VM) handler have been corrected:

- Previously, the system crashed after you installed VM if the BASE was SET beyond the end of available memory. Now, the error message ?KMON-F-Invalid device installation xx:VM.SYS, is displayed.

- VM failed to turn on 22-bit addressing when the boot code base address in extended memory was set at or slightly above the 18-bit/22-bit boundary. VM now correctly turns on 22-bit addressing when the boot code base address is set at or above the 18-bit/22-bit boundary. A further refinement was made to the VM code for RT-11 V5.4 that completely resolved this problem.

3.3.5 System Generation

The following problems in the system generation program SYSGEN have been corrected.

- If an answer file was used as input to SYSGEN to generate a system for CTS-300, SYSGEN would not generate new printer source files. Now SYSGEN.COM generates new source files, if necessary, when an answer file is used as input. The .CTS answer file created by SYSGEN shows the number of printers (LPNUM) requested and the CSR and vector addresses for each printer.

- SYSGEN included the conditional LS$PC in the .CND file only when both LS PC300 printer port and LP device support were specified. LS$PC is now included when only LS PC300 printer port support is specified. Selecting LP support is no longer required to set SYSGEN conditional LS$PC = 1.

- A special dialog for Professional/CTS V01.0 support has been added to SYSGEN. To initiate the dialog, the file PROCTS.ANS must be on the default volume.

3.3.6 System Macro Library (SYSMAC)

The following problems with programmed requests in the system macro library (SYSMAC) have been corrected.

3.3.6.1 .ABTIO

- The .ABTIO request incorrectly aborted all of a job's outstanding I/O operations; that is, it aborted all outstanding I/O on all channels. Also, all aborted I/O queue elements were lost. Now, .ABTIO correctly aborts a job's outstanding I/O operations only on the specified channel. The aborted I/O queue elements are correctly returned to the list of available queue elements. See the System Macro Library section in Chapter 2 for details.

3.3.6.2 .QSET

- When running under the XM monitor, .QSET incorrectly returned the number of queue elements requested in R0. Now, .QSET correctly returns the address of the first word beyond the allocated queue elements in R0.

3.3.7 System Subroutine Library (SYSLIB)

The following problems with subroutines in the system subroutine library (SYSLIB) have been corrected.

3.3.7.1 CONCAT

- A problem in the CONCAT function caused incorrect error reporting. An error was not correctly reported when both the length specified for the output string was shorter than the first input string and the address of the first input string fell on a block boundary. CONCAT now correctly reports errors.

3.3.7.2 IASIGN

- IASIGN could suppress console (TT) carriage control if it was called before any output was sent to the console. Now, IASIGN no longer suppresses console carriage control if it is called before output is sent to the console.

- If a program did not specify carriage control, IASIGN set the default carriage control to expanded. Now, if carriage control is not specified, IASIGN causes the FORTRAN OTS to send expanded carriage control information to the terminal and line printer, and unexpanded carriage control information to all other devices and files.

3.3.7.3 ILUN

- The ILUN function called the FORTRAN OTS routine, $FCHNL, and returned the RT-11 channel number associated with a specific FORTRAN LUN. The $FCHNL routine assigned a channel number to a LUN if it was not already assigned one. When a user program looped to determine which channels the LUNs were assigned to, the program would eventually fill the channel address table and cause FORTRAN Error 21 (out of logical units). The ILUN function now calls a local copy of the $FCHNL routine. $FCHNL no longer assigns an available channel to an unassigned LUN.

3.3.7.4 INTSET

- The linkage established by INTSET between the processor and device priorities was performed improperly. That resulted in the processor executing the interrupt at a processor priority level that was lower than the device priority level. Now, the processor executes the interrupt when the processor and device priority levels are the same.

3.3.7.5 MTIN, MTOUT

- The optional argument ocnt in the MTIN and MTOUT subroutines returned the value zero, regardless of the setting of bit 6 in M.TSTS. The optional argument ocnt now returns a value equal to the number of characters transferred on return from the subroutines when bit 6 of M.TSTS is set.

3.3.7.6 SCOPY

- SCOPY did not set the err argument (to .TRUE.) when the input and output arguments specified the same array name, and the input array was correctly truncated to the value specified in the len argument. Now, the err argument is correctly set (to .TRUE.) under those circumstances.

3.3.8 MACRO-11 Assembler

See Appendix J of the PDP-11 MACRO-11 Language Reference Manual, Update 2, for corrections to the MACRO-11 assembler.

3.3.9 Error Severity Levels

Error level descriptions were inconsistent throughout the RT-11 documentation. They have been corrected and made consistent.

3.3.10 Miscellaneous Corrections

The following miscellaneous problems have been corrected.

STARTF.COM	The /BUFFER:256. option has been added to the FRUN SPOOL and SRUN SPOOL commands. Also, the SET USR NOSWAP command has been added before the spooler related commands, and the SET USR SWAP command has been added after the spooler related commands.
V5USER.TXT	The system introductory text now includes a reference to the file V5NOTE.TXT.

3.4 DOCUMENTATION CORRECTIONS AND ADDITIONS

This section contains information that was incorrect in or inadvertently left out of the RT-11 documentation set.

3.4.1 RT-11 System Utilities Manual

This section contains corrections or additions to the RT-11 System Utilities Manual.

- **LD (logical disk subsetting program)**

  Chapter 9 does not include specific information on creating logical disks. Use the following method to create logical disks.

  Creating Logical Disks

  Creating a logical disk in this manner eliminates the possibility of directory conflicts with previous logical disks that might have resided in the same area on your physical disk.

  1. Create a file of the required size using the CREATE command with the /ALLOCATE:size option.
  2. Mount the file as a logical disk using the MOUNT command.
  3. Initialize the logical disk.

  You can create logical disks using the following indirect control file. In the file, P1 is the file name, P2 is the size you allocate for the file, and P3 is the logical disk device and unit number. The disk device and unit number is followed by a colon (:). Call the file (for example) MAKDSK.COM.

```
CREATE 'P1'/ALLOCATE:'P2'
MOUNT 'P3' 'P1'
INITIALIZE 'P3'/NOQUERY
```

  You can invoke MAKDSK.COM and create (for example) an 800(decimal) block logical disk named FOO.DSK on logical disk LD0 by executing the following command:

```
.IND MAKDSK FOO.DSK 800. LD0: <RET>
```

  or, if KMON is SET IND:

```
.@MAKDSK FOO.DSK 800. LD0: <RET>
```

- **LINK Cross-Reference Option (/N)**

  The last two paragraphs in Section 11.6.13, Cross-Reference Option (/N), contain inaccurate information. LINK does not first attempt to generate the temporary file CREF.TMP on the DK device. In fact, LINK first determines if a device has been ASSIGNed the logical name CF, and if so, LINK uses that device to generate CREF.TMP. LINK only uses the DK device to generate CREF.TMP if no device has been assigned the logical name CF.

3.4.2 RT-11 Installation Guide

This section contains corrections or additions to the RT-11 Installation Guide.

- **Section 2.3.8, Line Printer Handlers**

  The information in this section, starting at the second paragraph, is incorrect. Do not rename your serial (line) printer handler LS.SYS or LSX.SYS to LP.SYS or LPX.SYS. (The X suffix indicates a handler for the XM monitor.)

  Substitute the following information for the entire Section 2.3.8:

  > The software distribution kit includes the line printer handlers, LP.SYS or LPX.SYS, and the serial printer handlers LS.SYS or LSX.SYS. If your hardware configuration includes a serial printer instead of a line printer, you should include only LS.SYS or LSX.SYS in your working system.
  >
  > You use the serial printer handler the same way you use the line printer handler. However, you should include the following command line in your start-up command file:
  >
  > ```
  > ASSIGN LS LP
  > ```
  >
  > If your serial printer is installed at nonstandard vector and control status register addresses, use the SET command to change the addresses (see Section 2.7.11.1). See the RT-11 System User's Guide for all LS handler SET commands.

- **Appendix Section B.8, Loading the TSV05 Bootstrap Using MICRO-ODT**

  Step 6 does not contain complete information. It should read:

  6. Finally, enable the processor (remove the halt condition) and type:

     ```
     10000G
     ```

     (The rest of the information is correct.)

- **Creating a Bootable Magtape**

  Chapter 8 does not include specific information on creating a bootable magtape. See Section 1.5.9 of this manual for information on creating a bootable magtape.

3.4.3 RT-11 System User's Guide

This section contains changes and corrections to the RT-11 System User's Guide.

- **DIFFERENCES Command**

  The /SLP[:filespec] option for the DIFFERENCES command is not correctly documented. You must specify an output filespec; it is not optional. The correct syntax is DIFFERENCES/SLP:filespec.

- **LIBRARY Command**

  The /MACRO[:n] option is incorrectly described. The n is incorrectly described as being the size in blocks, when in fact n is the number of macro names that can be stored in the directory. The default value for n is 128 (which uses 2 blocks for the macro name table). The rest of the description is correct.

- **SET DU RETRY=n Command**

  The DU handler is incorrectly listed under the dd: RETRY=n command section. The description of the dd: RETRY=n command states that the Error Logger utility must be running to enable that command. That is not true in the case of DU. When you SET DU RETRY=n, you modify without restriction the number of times the handler attempts error recovery.

- **SET LS WIDTH=n Command**

  The maximum value for n in this command description is incorrectly listed as 255. In fact, the maximum decimal integer line width you can specify for the SET LS WIDTH=n command is 32767. The rest of the command description is correct.

3.4.4 RT-11 Software Support Manual

This section contains changes and additions to the RT-11 Software Support Manual.

- Chapter 7, Device Handlers

  7.2.1.3 Device Status Word

  The comment for the symbol SPECL$ (bit 12), ;NO DIRECTORY, is incorrect. The line for the symbol SPECL$ should read:

```
SPECL$  = =  10000  ;SPECIAL DIRECTORY STRUCTURED DEVICE
```

  Also, below Table 7-2, the stat argument for the LP handler should read:

```
For LP:     WONLY$!SPECL$
```

  7.9.8 Extended Memory Subroutines

  The description of the XALLOC subroutine for obtaining free memory does not make clear that the size argument passed in R2 is in units of 32(decimal) words. For example, to allocate 32000. words, specify 1000. as the size argument passed in R2.

  Also, if the required amount of memory is available, the carry bit will be clear, and R1 contains the region address divided by 32(decimal).

- Chapter 8, File Formats

  Figure 8-35: Object Library Header Format

  The descriptions for Offset 10 and Offset 20 are reversed. Offset 20 in the figure is incorrectly described as '1 if library created with /X option'. That description applies to Offset 10. Offset 20 should be described as 'Reserved'.

  The description for Offset 30 is incorrect. Offset 30 should be described as 'Size (bytes) of directory and header'.

- Chapter 10, Programming for Specific Devices

  EOF (end-of-file) detection for TT, XC, and XL:

  The following information concerning end-of-file detection for the TT, XC, and XL handlers is not included in Chapter 10.

  A CTRL/Z within data being read from the TT, XC, or XL device is treated as end-of-file (EOF) by the .READ request. At least two .READ requests are necessary to return the EOF error (carry bit set and byte 52 containing error code 0). The first .READ request transfers into your buffer all data up to (but not including) the CTRL/Z. The rest of the buffer is padded with nulls. A second .READ request is required to get the EOF error. Subsequent .READ requests can return additional characters.

  Section 10.14.3, Example of Allocating an Ethernet Unit

  The following example program contains code supporting the NU Ethernet handler, along with support for NC and NQ. The example program in Section 10.14.3 is accurate, but does not include support for NU. Support for the NU Ethernet handler is new for RT-11 V5.4.

  The program works for Professional 300 series processors, Q-BUS processors, and UNIBUS processors, and maintains job separation.

```
CONFG2  = 370           ;Config word 2
                        ; (RMON fixed offset)
PROS$   = 020000        ;RT is running on a PRO-3xx
BUS$    = 000100        ;Q-BUS/UNIBUS processor
          .
          .
          .
.GVAL   #AREA,#CONFG2   ;Get contents of Config word 2
MOV     #<^RNC >,DBLK   ;Assume PRO
BIT     #PROS$,R0       ;Correct assumption?
BNE     10$             ;yes...
MOV     #<^RNQ >,DBLK   ;No, so assume Q-BUS
BIT     #BUS$,R0        ;Correct assumption?
BNE     10$             ;yes...
MOV     #<^RNU >,R0     ;Nope, must be
                        ; UNIBUS after all
```

```
10$:    .GTJB   #AREA,#JOBDAT   ;Get info on this job
        MOV     JOBDAT,R0       ;R0 = job number (*2)
        ASR     R0              ;Convert to job number 0-7
        ADD     #<^R  0>,R0     ;Make it final RAD50 digit
        ADD     R0,DBLK         ; and add it to
                                ; the device name
        .LOOKUP #AREA,#0,#DBLK  ;Open a channel to Ethernet
                .
                .               ;.LOOKUP error processing
                .
        .SPFUN  #AREA,#0,#200,#BUFFER,#0,#1
                                ;Allocate the unit to this job
                .
                .               ;.SPFUN error processing
                .
AREA:   .BLKW   3
JOBDAT: .BLKW   12.
DBLK:   .WORD   0,0,0,0
BUFFER: .BLKW   4
                .
                .
                .               ;END OF EXAMPLE
```

Section 10.15.1, (MU) Use of Special Functions (SPFUN)

This section does not include the definitions for the request arguments when you use .SPFUN 360 with MU. The following is the correct request syntax and definitions for the request arguments:

Macro Call: .SPFUN area,chan,func,buf,wcnt,blk

where:

area is the address of a 6-word EMT argument block

chan is a channel number in the range 0 to 376(octal)

func is 360, the MU/DU BYPASS function code

buf is 0 (unused)

wcnt when nonzero, is the virtual address of a data buffer to send to the handler. That virtual address is converted to a physical address and placed in the command message

when zero, the physical address of a data buffer is sent to the handler. That physical address is placed in the command message

blk indicates whether the handler should perform retries:

1 = specifies retries

0 = specifies no retries

Section 10.15.2, (MU) Unit Support, CSR, and Vectors

This section contains inaccurate information. In fact, MU supports up to four units. Each unit does require a separate controller. You can only boot RT-11 from MU0, and MU0 must be installed at CSR address 774500 and vector 260. The addresses for MU1 through MU3 float; they depend on what other devices are on the bus. The default CSR and vector assignments are as follows:

CSR	Vector
774500	260
774504	264
774510	270
774514	274

3.4.5 RT-11 Programmer's Reference Manual

This section contains changes and additions to the RT-11 Programmer's Reference Manual.

- Chapter 2, Programmed Request Description and Examples

  .CSIGEN Request

  The description of the .CSIGEN request incorrectly states that it closes channels 0 through 10(octal). In fact, .CSIGEN under the FB and XM monitors performed a purge of those channels. For RT-11 V5.4, .CSIGEN performs a purge of those channels under all monitors. Also, if CSI errors are detected, .CSIGEN purges channels 0 through 10(octal) under all monitors.

  .DRBEG Macro

  The first sentence of the first paragraph is no longer accurate. In fact, .DRBEG now sets up the first six (not five) words of the handler. See Section 2.36 of this manual for information on the sixth word.

  .SCCA Request

  The description of the .SCCA request should mention that it is only available under the FB and XM monitors. The .SCCA request is also incorrectly listed in Table 1-4: Programmed Requests for All RT-11 Environments. The .SCCA request should be listed in Table 1-5: Foreground/Background and Extended Memory Programmed Requests. Given those corrections, the description of the .SCCA request is otherwise correct.

.SPFUN Request

The functions Forward one block and Backspace one block (function codes 376 and 375) for devices MM, MS, MT, MU are incorrectly described. Those functions do not necessarily act on only one block, but instead on n blocks, where the value for n is supplied in the wcnt argument. The description for those functions should read:

Function	MM,MS,MT,MU
Forward n blocks*	376
Backspace n blocks*	375

* The value for n is supplied by the wcnt argument.

Therefore, the first two sentences in the paragraph above the function table should be replaced with the following:

> The chan argument is the same as defined for the .READ/.WRITE requests. The contents and meaning for the blk and wcnt arguments are specific to the .SPFUN request in which they are used. See Chapter 10 of the RT-11 Software Support Manual for information. If the crtn argument is left blank,...(the rest of the paragraph is correct).

- Chapter 3, System Subroutine Description and Examples

CONCAT Subroutine

The description of the len argument implies that the output string is always truncated to the specified value. That is not correct. The output string is truncated to the value specified in the len argument only when the size of the concatenated string (a concatenated with b) is larger than the value specified in the len argument.

ISPFN Subroutines

The functions Forward one block and Backspace one block (function codes 376 and 375) for devices MM, MS, MT, MU are incorrectly described. Those functions do not necessarily act on only one block, but instead on n blocks, where the value for n is supplied in the wcnt argument. The description for those functions should read:

Function	MM,MS,MT,MU
Forward n blocks*	376
Backspace n blocks*	375

* The value for n is supplied by the wcnt argument.

SCCA System Subroutine

The description of the SCCA subroutine should mention that it is only available under the FB and XM monitors.

SCOPY Subroutine

The description of the len argument implies that the output string is always truncated to the specified value. That is not correct. The output string is truncated to the value specified in the len argument only when the size of the input string (in argument) is larger than the value specified in the len argument.

- Appendix C, Running PRO/GIDIS Under RT-11

  In the first paragraph of Appendix C, the order number reference for the PRO/GIDIS Manual should read AD-Y660A-TK. The order number AD-Y660A-T1 is for the first Update to that manual.

3.4.6 RT-11 Mini-Reference Manual

The following section contains additions and corrections to the RT-11 Mini-Reference Manual.

- KMON Commands

  Library

  The /MACRO[:n] option is incorrectly described. The n is incorrectly described as being the size in blocks, when in fact n is the number of macro names that can be stored in the directory. The default value for n is 128 (which uses 2 blocks for the macro name table).

- Utilities

  Libr

  The /M:n option is incorrectly described. See the correction in the KMON LIBRARY command section above.

- Programmed Requests

  .DRBEG Macro

  The first sentence of the first paragraph is no longer correct. In fact, .DRBEG now sets up the first six (not five) words of a device handler. The sixth word is described in Section 2.36 of this manual.

- Data Structures

Configuration Word 2 (RMON fixed offset 370)

The meaning for the BUS$ and PROS$ bits is not clear. It should be:

Name	Bits	Meaning
BUS$	100	Set = Q-BUS
		Clear = UNIBUS or CTI-BUS (check PROS$)
PROS$	20000	Set = RT-11 running on a Professional series computer
		Clear = UNIBUS or Q-BUS processor (check BUS$)

CHAPTER 4

INSTALLATION, BOOTSTRAP, AND HARDWARE SETUP PROCEDURES

This chapter describes procedures you may need to follow, depending on your distribution kit and your hardware configuration. Additional information for this chapter appears in two files on the RT-11 distribution kit:

V5NOTE.TXT contains release note information formalized too late for inclusion in the printed release notes.

CUSTOM.TXT contains a table that helps you determine the correct values and addresses to use when installing software customizations.

Chapter 2 of the RT-11 Installation Guide provides and describes software customizations. In the customizations, symbols are used in place of values and addresses. When you install software customizations, use the values and addresses provided in CUSTOM.TXT in place of each symbol shown in the customization.

4.1 FORMATTING DISKS ON PROFESSIONAL 300 SERIES PROCESSORS

DIGITAL recommends formatting RD50, RD51, and RD52 hard disks before they are used. If you boot a Professional 300 series processor from a diskette, and the processor contains an unformatted hard disk, a hardware diagnostic picture is displayed on the screen. The diagnostic picture displays the number 010030 followed underneath by 000401, indicating an unformatted hard disk. Software cannot be installed on the hard disk.

The RT-11 system boots from the diskette after displaying the diagnostic picture. Execute the following sequence of commands to format and install the hard disk:

```
.FORMAT DW:
.INSTALL DW:
.LOAD DW:
.FORMAT/VERIFY:ONLY DW:
```

The command FORMAT DW: generates the informational message ?FORMAT-I-Install DW. Attempting to verify a hard disk with FORMAT/VERIFY or FORMAT/VERIFY:ONLY before formatting an unformatted hard disk generates the error message ?FORMAT-F-Invalid device for /VERIFY, or not formatted.

4.2 PROCEDURES FOR DISTRIBUTION KITS

All RT-11 distribution kits contain system software, and automatic installation and verification software. However, automatic installation and verification is supported only with certain distribution kits: RL02, RX02, RX50, and TK50. If you attempt to bootstrap the automatic installation monitor (RT11AI.SYS for PDP-11 systems, RT11PI.SYS for Professional 300 series systems) on a distribution kit that does not support automatic installation, the installation procedure informs you that automatic installation is not supported, and you must install your system manually.

You may need to follow one of the procedures described below, if your distribution kit does not support automatic installation or if your hardware configuration does not meet automatic installation requirements.

- When you bootstrap an RL02, RX02, RX50, or TK50 distribution kit, the automatic installation monitor (RT11AI.SYS for PDP-11 systems, RT11PI.SYS for Professional 300 series systems) is bootstrapped. If your hardware configuration meets automatic installation requirements, the installation procedure tells you how to proceed. (See the appropriate RT-11 Automatic Installation Booklet for more details on automatic installation.) If, however, your hardware configuration does not meet automatic installation requirements, the automatic installation procedure informs you that automatic installation is not supported, and you must install your system by following the procedures in the RT-11 Installation Guide. If your configuration does not include a line clock or 24K words of memory (FB monitor requirements), you must first copy the bootstrap for the SJ monitor from RT11SJ.SYS to your system volume while still operating under the automatic installation monitor, RT11AI.SYS or RT11PI.SYS.

- RX01 distribution kits contain two copies of the RT-11 software, so you do not need to back up the distribution kit. Store one copy as your master distribution kit, and bootstrap Volume 1 of the second copy to initiate system installation. Follow the instructions in the RT-11 Installation Guide. If you are following Chapter 3, you need not perform any of the operations described in Section 3.2 for preserving the distribution volumes, except for the instructions for removing protection from files beginning near the middle of page 3-6.

- Volume 10 of RX01 distribution kits includes the same software as the AUTO volume on kits that support automatic installation. That volume is also PDT-11/150 bootable. However, if you bootstrap that volume on a PDT-11/150, the installation procedure informs you that automatic installation is not supported, and you must follow the procedures in the RT-11 Installation Guide to install your system. To create a PDT-11/150-bootable RT-11 distribution kit, you should respond to the keyboard monitor prompt (.) by using the COPY/BOOT command to copy the PD bootstrap to volume 1 of the RX01 distribution kit.

4.3 RX33 DISKETTE DRIVE INSTALLATION AND FORMATTING SUPPORT

This section describes how to manually install RT-11 on processors that contain the RX33 diskette drive and how to format RX33 diskettes on that drive.

4.3.1 Manual Installation Using the RX33 Diskette Drive

Automatic installation is not supported on the RX33 diskette drive. If your distribution media is RX50 diskettes and your processor contains the RX33 diskette drive, you must install RT-11 manually.

CAUTION

The following installation procedure assumes you have not previously used your system. The following procedure initializes your hard disk. That erases any files located on the disk. If you have files on your hard disk you want to preserve, back up those files before performing the following installation procedure.

Use the following procedure to install RT-11:

1. If your processor is turned on, turn it off by pressing the power switch to the 0 position. Remove any diskette located in the RX33 diskette drive.

2. From your RX50 distribution diskettes, select the diskette with the following text on the diskette label:

   RT-11 V5.4 BIN RX50-1

   Put aside the diskette labeled, RT-11 V5.4 BIN RX50-AUTO. You will use it later in this procedure.

3. Insert the RX50-1 diskette in the RX33 diskette drive.

4. Turn on your processor.

5. Your processor will go through a system start-up self-test and then boot that diskette. Some introductory text appears on your terminal screen and you receive the monitor prompt (.). You are now running your system from your RX33 diskette drive.

6. Issue the following command:

   ```
   .INITIALIZE/NOQUERY DU0:
   ```

   Wait while RT-11 initializes your hard disk. You can proceed to the next step when you receive the monitor prompt on your terminal screen.

7. Issue the following command:

   ```
   .COPY/SYSTEM/VERIFY SY:  DU0:
   ```

8. When you receive the monitor prompt, issue the following command:

   ```
   .COPY/BOOT DU0:RT11XM.SYS DU0:
   ```

9. When you receive the monitor prompt, issue the following command:

   ```
   .BOOT DU0:
   ```

   The same introductory text appears on your terminal screen. You are now running your system from your hard disk.

10. Remove the diskette from your RX33 diskette drive.

11. From your RX50 distribution diskettes, select the diskette with the following text on the label:

    RT-11 V5.4 BIN RX50-2

12. Insert that diskette in the RX33 diskette drive and issue the following command:

    ```
    .COPY/SYSTEM/VERIFY DU1:  DU0:
    ```

    That command copies the files from the diskette located in the RX33 diskette drive to your hard disk. Ignore the message, ?PIP-W-Reboot, if it appears on your terminal screen.

13. When you receive the monitor prompt, remove the diskette from the RX33 diskette drive.

14. Select successive diskettes with ascending numbers (RX50-3, RX50-4, and so forth). Insert each diskette in the RX33 diskette drive and issue the following command:

    ```
    .COPY/SYSTEM/VERIFY DU1:  DU0:
    ```

    That command copies the files on each diskette to your hard disk. Ignore the message, ?PIP-W-Reboot, if it appears on your terminal screen.

15. Select the diskette labeled RT-11 V5.4 BIN RX50-AUTO. Insert that diskette in the RX33 diskette drive. Issue the following command:

    ```
    .COPY/SYSTEM/NOREPLACE/VERIFY DU1:  DU0:
    ```

    Ignore the message, ?PIP-W-Reboot, if it appears on your terminal screen. Leave that diskette in the drive.

16. Run the installation verification procedure (IVP) by issuing the following command:

    ```
    .IND VERIFY
    ```

    Follow the instructions that are displayed on your terminal screen. You have correctly installed RT-11 on your system when IVP successfully completes.

17. Place the RX50 distribution diskettes back in their box and store them in a safe place.

18. You can optionally run the system configuration command file CONFIG.COM. That command file selectively deletes files that are not appropriate to your system.

    Issue the following command to run the system configuration program:

    ```
    .IND CONFIG
    ```

    Follow the instructions that are displayed on your terminal screen.

19. DIGITAL recommends you edit the start-up command file, STARTX.COM, to set the functions you want when you boot your

system. See the RT-11 System User's Guide for information about start-up command files. See also Appendix C of this manual for information on running an RT-11 working system from your virtual memory (VM) device.

Now, when you turn on your processor, it boots the RT-11 operating system located on your hard disk. You have manually installed RT-11.

4.3.2 Formatting RX33 diskettes

RX33 diskettes must be formatted before you use them. Two versions of a formatting diskette are shipped with the RX33 diskette drive. If your RX33 diskette drive was shipped with formatting diskette BL-FM5AA-MC, see Section 4.3.2.1. If it was shipped with diskette BL-FM5AB-MC, see Section 4.3.2.2.

4.3.2.1 **Using Formatting Diskette BL-FM5AA-MC** - Use the following procedure to format RX33 diskettes with diskette BL-FM5AA-MC:

1. If your processor is turned on, close any files you have open. Turn off your processor by setting the power switch to 0.

2. Remove any diskette you have in the drive.

3. Select the formatting diskette that was included with your RX33 diskette drive (part number BL-FM5AA-MC). Be sure an adhesive write protection tab is covering the write protection notch on that diskette. If one is not installed there, put one there.

4. Load that diskette in the RX33 drive.

5. Turn on your processor by setting the power switch to 1.

6. Your processor goes through a self-test sequence and boots the formatting diskette. Some lines appear on your terminal screen indicating that, and the formatting diskette monitor prompt (.) is displayed. Enter the following command in response to that monitor prompt:

   ```
   .R ZRQFA0 <RET>
   ```

7. The formatting program starts, prints some information on your screen, and returns the formatting program prompt (DR>). Enter the following command in response to that prompt:

   ```
   DR>START <RET>
   ```

   That command starts the formatting procedure. The formatting procedure displays a dialog on your terminal screen. Enter the following responses (indicated by red print) to the dialog questions:

   ```
   Change HW (L)  ?  Y

   # UNITS (D)  ?  1
   ```

```
unit 0
IP Address (0) 172150  ?  <RET>
Vector Address (0)  154  ?  <RET>
Logical Drive (0-255)  (D) 1  ?  <RET>

MSCP Controller model #: 19
    Microcode version #:  2

WARNING - Remove boot diskette if in drive.
```

Remove the formatting diskette from the RX33 diskette drive. Insert an RX33 diskette you want to format in the RX33 diskette drive, and press the RETURN key in response to the following question:

```
Insert a diskette to be formatted & press <RETURN>.  (L) N ?  <RET>
```

The following three lines are displayed as the formatting procedure formats the RX33 diskette in the drive. That can take a few minutes.

```
Format Begun
FCT was not used
Format Completed
```

When you see the last line above displayed on your terminal screen, the formatting of the RX33 diskette in the drive is complete. In response to the following question, press the RETURN key if you do not want to format more RX33 diskettes, or type Y if you do want to format more diskettes. Assuming you do not, this example dialog shows the negative response:

```
Do you want to format another diskette (L) N ?  <RET>
```

Now, remove the newly formatted RX33 diskette from the drive and reinsert the formatting diskette. Once you have done that, press the RETURN key:

```
If boot drive, reinsert boot diskette & press <RETURN>. (L) N ?  <RET>
```

Disregard any error messages you may see displayed on your terminal screen.

Now that the formatting procedure is finished, you are returned to the formatting program prompt. Type EXIT in response to that prompt:

```
DR>EXIT
```

You are then returned to the formatting diskette monitor prompt (.).

8. Remove the formatting diskette from the RX33 drive and reboot your RT-11 system.

9. Once your system has booted RT-11, insert the RX33 diskette(s) you have formatted in the RX33 drive and issue the following command for each:

```
.INITIALIZE/BADBLOCKS/NOQUERY DU1:
```

10. You have now formatted and initialized your RX33 diskette(s), making them usable as RT-11 system and data media.

4.3.2.2 **Using Formatting Diskette BL-FM5AB-MC** - Use the following procedure to format RX33 diskettes with diskette BL-FM5AB-MC:

1. If your processor is turned on, close any files you have open. Turn off your processor by setting the power switch to 0.
2. Remove any diskette you have in the drive.
3. Select the formatting diskette that was included with your RX33 diskette drive (part number BL-FM5AB-MC). Be sure an adhesive write protection tab is covering the write protection notch on that diskette. If one is not installed there, put one there.
4. Load that diskette in the RX33 drive.
5. Turn on your processor by setting the power switch to 1.
6. Your processor goes through a self-test sequence and boots the formatting diskette. Follow the instructions on your terminal screen to format RX33 diskette.
7. Once you have completed the formatting procedure, boot your RT-11 system, insert the RX33 diskette(s) you have formatted in the RX33 drive and issue the following command for each:

```
.INITIALIZE/BADBLOCKS/NOQUERY DU1:
```

8. You have now formatted and initialized your RX33 diskette(s), making them usable as RT-11 system and data media.

4.4 MANUAL INSTALLATION OF THE TK50 SOFTWARE DISTRIBUTION

The procedure to manually install the software distribution from a TK50 magtape is essentially the same as that described in Chapter 8 of the RT-11 Installation Guide. However, the TK50 magtape uses the new MU device handler, so you must specify the new magtape build program MDUP.MU in response to the MSBOOT prompt. The rest of the procedure is the same.

If your processor does not contain the correct MU boot ROM and you want to boot your processor from a bootable TK50 magtape, you must manually load the TMSCP software bootstrap. You can load the TMSCP software bootstrap using MICRO-ODT, or the console switch register if your processor has a console switch register.

4.4.1 Loading the TMSCP Bootstrap Using MICRO-ODT

Deposit the TMSCP bootstrap loader in memory as follows:

1. Place the bootable TK50 magtape in drive unit 0 and power-up the drive.
2. Turn on your processor; if it is already on, halt it.
3. At the console, type the first address from the table in Section 4.4.3 (76026) followed by a slash (/):

```
76026/
```

The system responds by printing the contents of address 76026 (represented below by xxxxxx) on the console:

```
76026/ xxxxxx
```

4. On the same line, type the first contents value from the table (5000) followed by a line feed:

```
76026/ xxxxxx 5000 <LF>
```

The system deposits the value you just typed (5000) and displays the next memory location:

```
76030/
```

5. Type the contents of the next memory location followed by a line feed.

6. Repeat step 5 until you have deposited all the contents. Then press RETURN.

7. Finally, enable the processor (remove the halt condition) and type:

```
76026G
```

The processor reads the software bootstrap from the magtape into memory. The system prints the following prompt when it is finished:

```
MSBOOT V05.0x
*
```

If the system does not print the MSBOOT prompt after you type 76026G, you probably made an error in entering the bootstrap loader. Open and examine each location using MICRO-ODT and correct the error.

8. Respond to the asterisk (*) by typing MDUP.MU:

```
* MDUP.MU <RET>
```

9. Follow the instructions in Chapter 8 of the RT-11 Installation Guide.

4.4.2 Loading the TMSCP Bootstrap Using the Switch Register

Deposit the TMSCP bootstrap loader in memory as follows:

1. Place the bootable TK50 magtape in drive unit 0 and power-up the drive.

2. Set the ENABLE/HALT switch on the processor to HALT.

3. Set the first address from the table in Section 4.4.3 (76026) in the switch register.

4. Press the LOAD ADDR switch.

5. Set the contents for the first address from the table (5000) in the switch register.

6. Lift the DEP switch.

   The processor automatically advances to the next address.

7. Set the contents for the next address from the table in the switch register.

8. Lift the DEP switch.

9. Repeat steps 7 and 8 until you have deposited all the instructions.

Now verify that you deposited the bootstrap loader properly.

1. Set the starting address, 76026, 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 Section 4.4.3.

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, correct the contents value and lift the DEP switch. Press EXAM again to display the contents. Verify the contents of all addresses in this way.

Once you have correctly deposited the bootstrap in memory, start the computer as follows:

1. Set the starting address, 76026, in the switch register.

2. Press the LOAD ADDR switch.

3. Set the ENABLE/HALT switch to ENABLE.

4. Press the START switch.

   The processor reads the software bootstrap from the magtape into memory. The system prints the following prompt when it is finished:

```
MSBOOT V05.0x
*
```

   If the system does not print the MSBOOT prompt after you press the START switch, you probably made an error in entering the bootstrap loader. Repeat the procedure.

5. Respond to the asterisk (*) by typing MDUP.MU:

   *MDUP.MU <RET>

6. Follow the instructions in Chapter 8 of the RT-11 Installation Guide.

4.4.3 Addresses and Contents of the TMSCP Bootstrap Loader

The following are the addresses and contents of the TMSCP bootstrap loader:

Address	Contents
076026	5000
076030	12701
076032	174500
076034	12704
076036	76160
076040	12705
076042	4000
076044	10102
076046	12422
076050	5712
076052	100770
076054	31205
076056	1774
076060	12412
076062	6305
076064	100371
076066	10502
076070	5022
076072	105702
076074	100375
076076	105237
076100	100077
076102	105237
076104	100075
076106	10037
076110	100104
076112	112437
076114	100115
076116	112437
076120	100110
076122	10522
076124	10522
076126	12722
076130	100100
076132	10512
076134	24242
076136	5711
076140	5712
076142	100776
076144	5737
076146	100012
076150	1331
076152	105704
076154	100344
076156	5007
076160	46525
076162	100000
076164	100200
076166	0
076170	1
076172	4400
076174	20476
076176	20402

4.5 MANUAL LOADING OF THE MSCP SOFTWARE BOOTSTRAP

If your processor does not contain the correct DU boot ROM and you want to boot your processor from an MSCP device, you must manually load the MSCP software bootstrap. You can load the MSCP software bootstrap using MICRO-ODT or the console switch register if your processor has a console switch register.

You can then manually install the software distribution as described in Chapter 9 of the RT-11 Installation Guide.

4.5.1 Loading the MSCP Bootstrap Using MICRO-ODT

Deposit the MSCP bootstrap loader in memory as follows:

1. Place the bootable MSCP volume in physical unit DU0: (the unit installed at DU CSR0) and power-up the drive.

2. Turn on your processor; if it is already on, halt it.

3. At the console, type the first address from the table in Section 4.5.3 (76000) followed by a slash (/):

   ```
   76000/
   ```

   The system responds by printing the contents of address 76000 (represented below by xxxxxx) on the console:

   ```
   76000/ xxxxxx
   ```

4. On the same line, type the first contents value from the table (5000) followed by a line feed:

   ```
   76000/ xxxxxx 5000 <LF>
   ```

   The system deposits the value you just typed (5000) and displays the next memory location:

   ```
   76002/
   ```

5. Type the contents of the next memory location followed by a line feed.

6. Repeat step 5 until you have deposited all the contents. Then press RETURN.

7. Finally, enable the processor (remove the halt condition) and type:

   ```
   76000G
   ```

   The processor reads the software bootstrap from the MSCP volume into memory. When it is finished, the system prints an RT-11 monitor identification message such as the following:

   ```
   RT11FB V05.0x
   ```

   If the system does not print an RT-11 monitor identification after you type 76000G, you probably made an error in entering the bootstrap loader. Open and examine each location using MICRO-ODT and correct the error.

4.5.2 Loading the MSCP Bootstrap Using the Switch Register

Deposit the MSCP bootstrap loader in memory as follows:

1. Place the bootable MSCP volume in physical unit DU0: (the unit installed at DU CSR0), and power-up the drive.
2. Set the ENABLE/HALT switch on the processor to HALT.
3. Set the first address from the table in Section 4.5.3 (76000) in the switch register.
4. Press the LOAD ADDR switch.
5. Set the contents for the first address from the table (5000) in the switch register.
6. Lift the DEP switch.

   The processor automatically advances to the next address.
7. Set the contents for the next address from the table in the switch register.
8. Lift the DEP switch.
9. Repeat steps 7 and 8 until you have deposited all the instructions.

Now verify that you deposited the bootstrap loader properly.

1. Set the starting address, 76000, 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 Section 4.5.3.
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, correct the contents value and lift the DEP switch. Press EXAM again to display the contents. Verify the contents of all addresses in this way.

Once you have correctly deposited the bootstrap in memory, start the computer as follows:

1. Set the starting address, 76000, in the switch register.
2. Press the LOAD ADDR switch.
3. Set the ENABLE/HALT switch to ENABLE.
4. Press the START switch.

   The processor reads the software bootstrap from the MSCP volume into memory. When it is finished, the system prints an RT-11 monitor identification message such as the following:

```
RT11FB V05.0x
```

If the system does not print an RT-11 monitor identification after you press the START switch, you probably made an error in entering the bootstrap loader. Repeat the procedure.

4.5.3 Addresses and Contents of the MSCP Bootstrap Loader

The following are the addresses and contents of the MSCP bootstrap loader:

Address	Contents
76000	5000
76002	12701
76004	172150
76006	12704
76010	76156
76012	12705
76014	4000
76016	10102
76020	10022
76022	5712
76024	100770
76026	31205
76030	401
76032	42125
76034	1772
76036	14412
76040	6305
76042	100367
76044	105744
76046	1002
76050	10704
76052	5007
76054	5002
76056	5022
76060	20227
76062	17204
76064	103774
76066	105237
76070	17101
76072	10037
76074	17110
76076	111437
76100	17114
76102	114437
76104	17121
76106	12722
76110	17004
76112	10522
76114	12722
76116	17104
76120	10512
76122	24242
76124	5711
76126	5712
76130	100776
76132	5737
76134	17016
76136	1742
76140	0
76142	20402
76144	4400

```
76146          1
76150          0
76152      17204
76154     100000
```

4.6 PROCEDURES FOR LINE PRINTERS AND TERMINALS

This section describes procedures you may need to follow to use your line printer or terminal.

4.6.1 SETUP Modes and Controls for Line Printers

The following sections list SETUP modes and controls for line printers supported by the SETUP utility.

4.6.1.1 Uniform SETUP Mode and Control Support for Line Printers -

The following SETUP modes and controls act in a uniform manner. That is, the action of the mode or control is consistent for all indicated line printers. A Yes in the table indicates support for the indicated mode or control. Dashes (---) indicate no support.

Modes and Controls	LA50 Printer	LA100* Printer	LA210* Printer	LN03 Printer	LQP02 Printer
[NO]BOLD	Yes	---	---	Yes	Yes
CLEAR	Yes	Yes	Yes	Yes	Yes
DOWN	Yes	Yes	Yes	Yes	Yes
GRAPHIC	Yes	Yes	Yes	---	---
HELP	Yes	Yes	Yes	Yes	Yes
HORIZONTAL	Yes	Yes	---	Yes	---
LANDSCAPE	---	---	---	Yes	---
LANG	Yes	Yes	Yes	Yes	---
PAGELENGTH	Yes	Yes	Yes	Yes	Yes
PORTRAIT	---	---	---	Yes	---
RESET	Yes	Yes	Yes	Yes	Yes
[NO]TABS:n	Yes	Yes	Yes	Yes	Yes
TEXT	Yes	Yes	Yes	Yes	---
[NO]UNDERLINE	Yes	Yes	Yes	Yes	Yes
UP	Yes	Yes	Yes	---	---
VERTICAL	Yes	Yes	Yes	---	---
[NO]WRAP	Yes	Yes	Yes	Yes	Yes

* Set LA100 and LA210 printers to AUTO mode before issuing them SETUP commands.

4.6.1.2 Nonuniform SETUP Mode and Control Support for Line Printers -

The following SETUP modes and controls act in a manner specific to particular line printers. That is, the action of the mode or control depends on the particular line printer you are using. Dashes (---) indicate no support.

Modes and Controls	LA50 Printer	LA100* Printer	LA210* Printer	LN03 Printer	LQP02 Printer
DENSE	---	Letter Mode, High Density	---	---	---
DRAFT	Normal Density	Draft	Draft	---	---
LETTER	Hor:10	Letter Mode, Medium Density, Hor:10	Hor:10	Hor:10	Hor:10
LISTING	Normal Density, Hor:16.5	Draft, Hor:16.5	Draft, Hor:16.5	Hor:16.5	Hor:12
MEMO	Enhanced Density	Letter Mode, Medium Density	Letter Mode, Medium Density	---	---

* Set LA100 and LA210 printers to AUTO mode before issuing them SETUP commands.

4.6.2 Line Printer Handlers

This section describes restrictions and workaround solutions for using line printers with RT-11.

- If output to some line printers is terminated prematurely, the print head may not be at the left margin. When output to the printer is resumed, printing may begin in the middle of the line. To prevent that, reset the printer manually.

- Some conditions cause data sent to a printer using the LS handler to be lost without warning. Those conditions are:

  For all terminals: the printer is powered down or placed off line.

  For all terminals except LA34, LA38, LA120, LS120, and serial LA180: The terminal encounters an error condition while in operation.

- The LS handler cannot be used to communicate with the console terminal. Instead, you must use the TT handler.

- The SET LS NOHANG option is valid only if device timeout support is included in a monitor created through system generation. However, DIGITAL recommends that you do not use the command SET LS NOHANG. That command can cause printers with very large buffers to abort before they are through printing.

- If the LP or LS handler NOFORM or SKIP option is used, load the handler by including a LOAD command in your STARTx.COM files. Then, manually set the printer paper to top of page each time the system is bootstrapped (top of page is normally set so printing begins on the fourth print line down from the page perforation). Afterward, you should not move the paper manually. Instead, the handler should perform all movement of the paper.

  For example, to remove the last printed file from the printer, you must send two form feeds to the printer. Create the file FF.LST with an editor, and include in the file only two form-feed characters (CTRL/L). When you issue the command PRINT FF, the printer will perform two form feeds, so you can remove the last file printed. The top of page will still be set correctly for printing the next file.

- For the LS handler, the default CSR address is 176500 and the default vector address is 300. If your line printer is connected to a DL11/DLV11 interface with different addresses, use the SET LS CSR and SET LS VECTOR commands to modify the default values. For example, to modify the LS handler to use the MINC-11 printer port, SLU2, issue the following command:

```
.SET LS CSR=176520,VECTOR=320
```

4.6.3 Recommended Terminal/Line Printer SET Command Options

This section lists the nondefault SET options recommended for use with the terminals and line printers shown.

LA50, LQP02

```
SET LP/LS TAB,CR,CTRL,FORM,NOFORM0,LC,WIDTH=n
```

LA100

```
SET TT TAB,FORM,WIDTH=n
SET LP/LS TAB,CR,CTRL,FORM,NOFORM0,LC,WIDTH=n
```

LA210

```
SET LP/LS TAB,CR,CTRL,FORM,NOFORM0,LC,WIDTH=n
```

LA30, LA35, LA36, and LS120

```
SET TT WIDTH=n
SET LP/LS CR,CTRL,NOFORM,NOFORM0,LC,WIDTH=n
```

LA34 and LA38

```
SET TT TAB,WIDTH=n
SET LP/LS CR,CTRL,NOFORM,NOFORM0,LC,TAB,WIDTH=n
```

LA120

```
SET TT FORM,TAB,WIDTH=n
SET LP/LS CR,CTRL,NOFORM0,LC,TAB,WIDTH=n
```

LA180

```
SET LP/LS CR,CTRL,NOFORM0,LC,WIDTH=n
```

VT05

```
SET TT SCOPE,TAB,WIDTH=72
```

VT50, VT52, VT55, and VT100 Series

```
SET TT NOCRLF,SCOPE,TAB
```

SET TT commands are not permanent and must be issued every time the monitor is bootstrapped. Therefore, DIGITAL recommends that you include the command in the appropriate STARTx.COM file(s). However, SET LP and SET LS commands modify the permanent copy of the handler, so you need to issue them only once.

When running under a multiterminal monitor, you can set the characteristics of local terminals other than the boot-time console: Include the command SET TT CONSOLE=n in an appropriate STARTx.COM file followed by the SET TT commands you want for that terminal. After all the terminal characteristics have been set, include the SET TT CONSOLE=0 command in the STARTx.COM file to return control to the boot-time console.

Of the SET TT commands listed, only the SET TT SCOPE command is valid for a single-terminal SJ monitor.

4.6.4 Recommended VT100 Series, VT105, and LA120 Series Setup Mode Options

This section lists recommended setup modes for VT100 series, VT105, and LA120 series terminals. Terminal bauds should be set to correspond to the computer interface. Setup options other than those discussed below should be set for operator preference or form requirements. To permanently save the setup options you select, type <SHIFT/S> on a VT100 series or VT105, or <SHIFT/9> on an LA120 series.

4.6.4.1 **VT100 Series and VT105 Terminals** - The following setup mode options are recommended for normal use on VT100 series and VT105 terminals:

On line

Autorepeat on

Margin bell off (when preparing FORTRAN programs, you may want to have the margin bell set on, to notify you when column 72 is reached)

Auto XON/XOFF on

Wrap around on

New line off

Interlace off

Parity off

Bits per character 8

Tabs set every 8 columns

4.6.4.2 **LA120 Series Terminals** - LA120 series terminals should be initialized to factory settings. Then, buffer control should be changed to small. The following factory setup options should not normally be changed:

Line/local status

Local echo

Auto newline

Parity and data bits

Auto repeat

Auto linefeed

XON/XOFF

Printer new line character

Use the LA120 local form-feed key to make sure that the paper is positioned correctly at top of form before printing a listing.

CHAPTER 5

NATIVE TRANSFER UTILITY (TRANSFER)

The native transfer utility (TRANSFER) is an unsupported program that runs on one of five host operating systems and copies files from an RT-11 stand-alone processor to the host processor and from the host to the standalone. TRANSFER runs on the following host operating systems:

- VAX/VMS
- MicroVMS
- RSX-11M
- RSX-11M-PLUS
- Micro/RSX

(In the rest of this chapter, VAX/VMS refers to VAX/VMS and MicroVMS, and RSX refers to RSX-11M, RSX-11M-PLUS, and Micro/RSX.)

TRANSFER is run only from the host processor.

TRANSFER can copy files from Files-11 volumes (on VAX/VMS or RSX) to RT-11 volumes. It can also perform transfers in the other direction; that is, from files on RT-11 volumes to files on Files-11 volumes. Files-11 is the disk volume format maintained by the VAX/VMS and RSX operating systems.

You can use command qualifiers to specify the format of the transferred output file. The supported formats are:

- ASCII
- Binary
- FORTRAN
- Image

In addition, you can use command qualifiers that cause TRANSFER to:

- Supply additional information during the transfer operation
- Display HELP information
- Operate in PROMPT mode, an interactive mode with questions and defaults
- Queue the output file to the host system default printer (VAX/VMS host only)

Before you can use TRANSFER, you must install the appropriate version on the host system. Two versions of TRANSFER are included on the RT-11 distribution kit. The program named TRANSF.EXE is the version that runs on VAX/VMS. The program named TRANSF.TSK is the version that runs on RSX.

Once you have installed the appropriate utility program on the host processor (running VAX/VMS or RSX), use VTCOM to establish a connection between the RT-11 stand-alone processor and the host. You can use the TRANSFER utility only after you have established that connection.

This chapter:

- Describes how to copy and install TRANSFER on the host processor
- Explains how to use TRANSFER
- Lists TRANSFER error messages

5.1 INSTALLING TRANSFER ON THE HOST

There are two ways that TRANSFER can be installed on the host processor. It can be installed as a system utility that is available to all system users or in an individual's account.

The host system manager must install TRANSFER if it is to be a system-wide utility.

You can install TRANSFER in your own account on the host. If you want to do so, perform the procedures in Section 5.1.1 (VAX/VMS) or 5.1.2 (RSX). You must have a privileged account to install TRANSFER on an RSX host. If you do not have a privileged account on an RSX host, you must have the system manager install TRANSFER in your account. You do not need a privileged account to install TRANSFER on a VAX/VMS host.

5.1.1 Installing TRANSFER On VAX/VMS

Install TRANSFER on Version 4.0 or subsequent versions of VAX/VMS.

Step 1: Copy TRANSF.EXE

If you know that the VAX/VMS system is running RTEM, you can use TRANSF.SAV (a different transfer utility) under RTEM to copy TRANSF.EXE from the RT-11 stand-alone processor to the host. Do not attempt that unless you are familiar with RTEM.

If the VAX/VMS system is not running RTEM, mount the RT-11 distribution volume on the host and issue the following VAX/VMS command to transfer TRANSF.EXE to the host.

When you type the command, replace the variable ddcu: with the VAX/VMS physical device name for the device on which TRANSF.EXE resides. Type the command exactly as illustrated.

```
$ EXCHANGE COPY/CONTIGUOUS ddcu:TRANSF.EXE/VOLUME_FORMAT=RT11 TRANSFER.EXE
```

NOTE

Because TRANSF.EXE is not an ASCII file, you cannot use the VTCOM SEND command to copy TRANSF.EXE from the RT-11 stand-alone processor to the host.

Step 2: Define TRANSFER As a Foreign Command

The following command completes the installation of TRANSFER by defining it as a foreign command. (A foreign command is a symbol name, in this case TRANSFER, that you use to invoke the utility.) Placement of the asterisk (*) lets you abbreviate TRANSFER to TRA.

When you type the following command, replace the variable disk with the name of the disk on which your VAX/VMS directory resides. Be sure to include the dollar sign ($) before the variable disk. Replace the variable directory with the name of your VAX/VMS directory.

```
$ TRA*NSFER :== $disk:[directory]TRANSFER.EXE
```

5.1.2 Installing TRANSFER On RSX

Install TRANSFER on the indicated version or subsequent versions of the operating systems in the following list. The RMSRES resident library must be installed on any RSX operating system running TRANSFER. The DAPRES resident library must be installed, along with system DECnet support, to access files across DECnet.

- RSX-11M-PLUS 3.0
- RSX-11M 4.2
- Micro/RSX 3.0

Step 1: Copy TRANSF.TSK

If you know that the RSX system is running RTEM, you can use TRANSF.SAV (a different transfer utility) under RTEM to copy TRANSF.TSK from the RT-11 stand-alone processor to the host. Do not attempt that unless you are familiar with RTEM.

If the RSX system is not running RTEM, mount the RT-11 distribution volume on the host and use the following RSX commands to transfer TRANSF.TSK to the host.

When you type the commands, replace the variable ddn: with the RSX physical device name for the device on which TRANSF.TSK resides.

```
>MOUNT/NOSHARE/FOREIGN ddn:

>FLX SY:=ddn:TRANSF.TSK/CO/RT/IM

>RENAME TRANSF.TSK TRANSFER.TSK
```

NOTE

Because TRANSF.TSK is not an ASCII file, you cannot use the VTCOM SEND command to transfer TRANSF.TSK from the RT-11 stand-alone processor to the host.

Step 2: Install the RSX Task Image

Use the following command to install TRANSFER. You must have a privileged account to issue this command. If you do not have a privileged account, have your system manager install TRANSFER.

```
>INSTALL TRANSFER.TSK
```

5.2 USING TRANSFER

TRANSFER can copy files between a PDP-11 processor running RT-11 and either a PDP-11 processor running RSX or a VAX processor running VAX/VMS. No intermediary (such as the RT-11 emulator, RTEM) is required.

To use TRANSFER, make sure the appropriate version of the TRANSFER utility has been installed on the host processor (running VAX/VMS or RSX). Then use VTCOM to establish a connection between the RT-11 stand-alone processor and the host. You can use the TRANSFER utility only after you have established that connection. If you run VTCOM as a system job, you must LOAD all handlers for devices you are using with VTCOM. The RT-11 SHOW command displays information about devices. See the RT-11 System Utilities Manual for complete information on using VTCOM.

Run the TRANSFER utility by issuing the TRANSFER command in response to the host system's prompt.

You can abort a file transfer at any time and return to the host system's prompt by typing CTRL/C. CTRL/C is the preferred method of aborting file transfers on RSX and VAX/VMS systems. You can also abort file transfers on VAX/VMS systems by typing CTRL/Y and then issuing the EXIT command in response to the monitor prompt.

You can exit from TRANSFER and return to the host system's prompt by typing CTRL/Z in response to a TRANSFER prompt.

5.2.1 TRANSFER Command Syntax

The TRANSFER command has the following syntax:

```
TRANSFER<SP>input-filespec¦ {/ASCII[:n]}[<SP>output-filespec¦ {/ASCII[:n]}]
                          ¦ {/BINARY[:n]}                  ¦ {/BINARY[:n]}
                          ¦ {/FORTRAN[:n]}                 ¦ {/FORTRAN[:n}
                          ¦ {/IMAGE[:n]}                   ¦ {/IMAGE[:n]}
                          ¦  /LOG                          ¦  /LOG
                          ¦  /PROGRESS[:n]                 ¦ /PROGRESS[:n]
                          ¦ {/REMOTE  }                    ¦ {/REMOTE  }
                          ¦ {/TERMINAL}                    ¦ {/TERMINAL}
                          ¦  /STATISTICS                   ¦  /SPOOL
                                                           ¦  /STATISTICS
```

or:

```
TRANSFER {/HELP}
         {/PROMPT}
         {/VERSION}
```

or:

```
TRANSFER<RET>
From: input-filespec[/qualifier[/qualifier...]]
To:   output-filespec[/qualifier[/qualifier...]]
```

where:

input-filespec

is the input file specification of the file you want to copy. The file specification is operating system dependent. Consult your operating system documentation for the correct construction of the file specification. However, observe the following:

- Wildcards are not allowed in the file specification.
- If you do not specify an output-filespec, the first six characters of the input file name and the first three characters of the input file type (extension) must be alphanumeric.
- TRANSFER does not access files contained within a virtual disk on the host.
- You cannot specify any host system record oriented device, such as a line printer, terminal, or magtape device.

output-filespec

is the file specification of the file to which you want the file transferred. The file specification is operating system dependent. Consult your operating system documentation for the correct construction of the file specification. However, observe the following:

- Wildcards are not allowed in the file specification.
- RT-11 truncates input file names to six characters and input file types (extensions) to three characters.
- You cannot specify any host system record oriented device, such as a line printer, terminal, or magtape device.

/qualifier

is a TRANSFER qualifier. There are two types of TRANSFER qualifiers. Mode qualifiers determine the format of the transferred file. Control qualifiers affect transfer file processing by providing information, by invoking interactive mode, or by queuing the output file to a printer.

{ }

indicates mutually exclusive qualifiers, which are mutually exclusive on the input or output-filespecs and must appear only once in the command.

5.2.2 TRANSFER Default File Types

TRANSFER recognizes certain file types. If you do not specify a mode qualifier, TRANSFER performs the transfer in the default mode for that file type. TRANSFER recognizes the following file types as being ASCII or Binary:

ASCII (variable-length records)

.ANS	.BAK	.BAS	.BAT	.BLI	.B16	.B2S	.B32	.C
.CBL	.CMD	.COM	.COR	.CTL	.DAT	.DBL	.DDF	.DIF
.DIR	.DMP	.DOC	.FOR	.FTN	.H	.LIS	.LOG	.LST
.MAC	.MAP	.MAR	.MEM	.ODL	.PAS	.REQ	.RNO	.R16
.R32	.S	.SLP	.SRC	.TEC	.TES	.TXT		

Binary (variable-length records)

.BIN	.LDA	.OBJ	.STB

IMAGE mode is TRANSFER's default transfer mode. If you do not specify the /ASCII, /BINARY, /FORTRAN, or /IMAGE qualifier, and if the file you specify does not have a recognized file type, TRANSFER performs the transfer operation in IMAGE mode with 512-byte fixed-length records.

5.2.3 TRANSFER Mode Qualifiers

TRANSFER has four mode qualifiers that specify each type of supported file format. Format conversions can be in either direction between Files-11 volumes and RT-11 volumes. Only one transfer mode qualifier can be specified in a command.

/ASCII[:n]

This qualifier formats the output file as ASCII. On an RT-11 output volume, the file contains ASCII data records, each terminated by a carriage return/line feed, escape, form feed, or vertical tab. TRANSFER removes rubouts, nulls, and vertical tabs from input records and adds carriage return/line-feed pairs to the end of records that do not end with escape, form feed, or line feed. When the host input file record attributes do not specify carriage control, TRANSFER assumes embedded carriage control. Embedded carriage control means that each record contains the control characters necessary for proper formatting. In that case, carriage return/line feed pairs are not appended to records.

In transfers from RT-11 to Files-11 volumes, TRANSFER removes carriage return/line-feed pairs from the end of records.

If you specify a size value (:n), TRANSFER generates fixed-length records of that size and pads them with nulls if required. If you omit the size value (or specify a size of zero), TRANSFER generates variable-length records.

/BINARY[:n]

Use the /BINARY qualifier to format the output file as binary. TRANSFER adds formatted binary headers and checksums to records that

it copies to RT-11 files and removes binary headers and checksums from records that it copies to Files-11 files.

When you transfer files to Files-11 volumes, TRANSFER generates fixed-length records of the size (:n) you specify. If you omit the size value (or specify a size of zero), TRANSFER generates variable-length records.

/FORTRAN[:n]

The /FORTRAN qualifier transfers files that contain FORTRAN carriage control characters. Use the /FORTRAN qualifier when the first character of each record is to be interpreted as the carriage control specifier. The /FORTRAN qualifier does not alter any record data. Use this qualifier only when the output volume is a Files-11 volume.

TRANSFER generates fixed-length records of the size (:n) you specify. If you omit the size value (or specify a size of zero), TRANSFER generates variable-length records.

/IMAGE[:n]

Use the /IMAGE qualifier to transfer files without performing any record translations on them. In other words, TRANSFER copies the files exactly as they are.

When you transfer files to Files-11 volumes, TRANSFER generates fixed-length records of the size (:n) you specify. If you omit the size value (or specify a size of zero), TRANSFER generates fixed-length records of 512 bytes.

5.2.4 TRANSFER Control Qualifiers

TRANSFER provides the following qualifiers to control transfer processing:

/HELP

The /HELP qualifier displays limited information about TRANSFER. The /HELP qualifier lists default TRANSFER formats for various file types.

/LOG

The /LOG qualifier creates a log of the names of all files transferred. The log displays a success message, the complete input and output filespecs, and the number of blocks or records transferred. For example:

```
$ TRANSFER FOO.TXT/LOG FOO.TXT
%TRANSFER-S-COPIED, USER:[RTINDEX]FOO.TXT;2 copied to __TTB0::DK:FOO.TXT
(6 blocks)
```

/PROGRESS[:n]

The /PROGRESS qualifier causes TRANSFER to display the progress of the transfer at specific intervals while the operation is taking place. Progress is displayed in record or block intervals, depending on the

format of the file and the direction of the transfer. You can specify the interval (:n). The default PROGRESS report interval is 10(decimal) records or blocks.

The /PROGRESS qualifier displays an informational message, the time-of-day, the blocks or records transferred during that interval, the number of retries, and the packet size. An increase in the number of retries and a decrease in the packet size generally indicate interference on the transmission line. If the packet size decreases to an unacceptable level, retry the transfer operation when your transmission line might be more clear.

An example of the /PROGRESS qualifier:

```
$ TRANSFER FOO.TXT/PROGRESS:1 FOO.TXT
%TRANSFER-I-PROGRESS, 14:33:25 blocks transferred=1 retries=0 packet_size=512
%TRANSFER-I-PROGRESS, 14:33:27 blocks transferred=2 retries=0 packet_size=512
```

/PROMPT

The /PROMPT qualifier causes TRANSFER to operate in an interactive mode that displays questions, indicates defaults, and accepts input. The questions and defaults change dynamically, based on the responses to earlier questions. PROMPT mode shows a list of available responses in parentheses. The default response appears in brackets. To choose the default response, press the return key <RET>. Choose the default unless you are sure that another qualifier is correct.

The following are sample PROMPT mode sessions.

The following example copies the ASCII file V5NOTE.TXT from the RT-11 stand-alone system to the host, keeping the same file name. Defaults are taken for the file format (ASCII) and host file record length (variable).

```
$ TRANSFER/PROMPT
Original file is on (HOST,REMOTE) [HOST]: remote<RET>
Name of original REMOTE file to copy: v5note.txt<RET>
Name of file to create on HOST [V5NOTE.TXT]: <RET>
Create HOST file with (ASCII,BINARY,FORTRAN,IMAGE) records [ASCII]: <RET>
Create HOST file with fixed-length records (YES,NO) [NO]: <RET>
%TRANSFER-S-COPIED, __TTA2::DK:V5NOTE.TXT copied to USER:[RT11]V5NOTE.TXT;19
(714 records)
There were 0 retries, with 3362 characters saved through compression encoding.
```

The following example copies the file VDT.OBJ from the host to an RT-11 stand-alone system, keeping the same file name. Because VDT.OBJ is a binary file, the default BINARY qualifier is chosen by pressing RETURN.

```
$ TRANSFER/PROMPT
Original file is on (HOST,REMOTE) [HOST]: <RET>
Name of original HOST file to copy: vdt.obj<RET>
Name of file to create on REMOTE [VDT.OBJ]: <RET>
Create REMOTE file with (ASCII,BINARY,FORTRAN,IMAGE) records [BINARY]: <RET>
%TRANSFER-S-COPIED, USER:[WINNING]VDT.OBJ;1 copied to __TTA2::DK:VDT.OBJ
(8 blocks)
There were 0 retries, with 276 characters saved through compression encoding.
```

The following example copies the image file (executable program) VTCOM.SAV from the RT-11 system to the host, keeping the same file name. All defaults are taken.

```
$ TRANSFER/PROMPT
Original file is on (HOST,REMOTE) [HOST]: remote<RET>
Name of original REMOTE file to copy: vtcom.sav<RET>
Name of file to create on HOST [VTCOM.SAV]: <RET>
Create HOST file with (ASCII,BINARY,FORTRAN,IMAGE) records [IMAGE]: <RET>
Create HOST file with fixed-length records of size (0-512) <512>: <RET>
%TRANSFER-S-COPIED, __TTC2:DK:VTCOM.SAV copied to USER:[RT11]VTCOM.SAV;2
(24 blocks)
There were 0 retries, with 5274 characters saved through compression encoding.
```

/REMOTE
/TERMINAL

The /REMOTE and /TERMINAL qualifiers indicate to TRANSFER which file is on the RT-11 system. /REMOTE and /TERMINAL are interchangeable; they have the same meaning. Do not specify both qualifiers in the same command.

To transfer a file from the host system to the stand-alone RT-11 system, use /REMOTE or /TERMINAL on the output-filespec. That operation is the default, so you can omit those qualifiers altogether from the command for this type of transfer.

To transfer a file from the stand-alone RT-11 system to the host system, use /REMOTE or /TERMINAL on the input-filespec.

The following example transfers the file FOO.TXT from the host to the stand-alone RT-11 system:

```
$ TRANSFER FOO.TXT FOO.TXT
```

The following example transfers the file FOO.TXT from the stand-alone RT-11 system to the host:

```
$ TRANSFER FOO.TXT/TERMINAL FOO.TXT
```

/SPOOL

Use the /SPOOL qualifier to queue the TRANSFER output file to the default VAX/VMS system printer queue. Specify the /SPOOL qualifier on the host output-filespec.

This qualifier is available only for VAX/VMS system transfers.

/STATISTICS

The /STATISTICS qualifier displays the number of retries and the number of characters saved through compression encoding. (Compression encoding is a transfer speed enhancement used by TRANSFER whether or not you specify the /STATISTICS qualifier.) For example:

```
$ TRANSFER FOO.TXT/STATISTICS FOO.TXT
There were 0 retries, with 400 characters saved through compression encoding.
```

/VERSION

The /VERSION qualifier displays the TRANSFER utility's version number.

5.3 TRANSFER UTILITY MESSAGES

This section describes the messages that can be returned by the TRANSFER utility.

5.3.1 Utility Message Format

The messages displayed by the TRANSFER utility have the following format:

```
%FACILITY-L-IDENT, TEXT
[-FACILITY-L-IDENT, TEXT]
```

where:

FACILITY — is the TRANSFER utility or operating system facility or component name. A percent sign (%) prefixes the first message issued, and a hyphen (-) prefixes each subsequent message.

L — is the severity level indicator, which has one of the following values:

Code	Meaning
S	Success
I	Information
W	Warning
E	Error
F	Fatal or severe error

Consult your host operating system documentation for the meaning of each severity level.

IDENT — is the abbreviation of the message text; the messages below are alphabetized by this abbreviation.

TEXT — is the explanation of the message.

[-FACILITY-L-IDENT, TEXT] — is the next message.

5.3.2 Messages Returned By TRANSFER

The following is an alphabetical listing of the abbreviation (IDENT) and text for messages that can be returned by TRANSFER. Included is an explanation of each message and the recommended action to be taken. Consult your host operating system documentation for messages not listed here.

AMBIGQUAL, ambiguous qualifier in command

Explanation — The command qualifier abbreviation contains too few characters to make it unique. Another qualifier begins with the same characters.

User action — Retype the command, using at least four characters of the qualifier name.

CHKSUMERR, binary record checksum error

Explanation — The checksum for a BINARY record was in error.

User action — Make sure the input file is not an object library. Object libraries must be copied in IMAGE mode rather than BINARY mode.

Make sure the input file is a valid binary file. Retry the transfer.

CLOSEIN, error closing 'input-file' as input

Explanation — RMS encountered an error while closing the indicated input file. This message is usually accompanied by an RMS message indicating the reason for the failure.

User action — Take corrective action based on the associated message.

CLOSEOUT, error closing 'output-file' as output

Explanation — RMS encountered an error while closing the indicated output file. This message is usually accompanied by an RMS message indicating the reason for the failure.

User action — Take corrective action based on the associated message.

CONFQUAL, conflicting qualifier in command

Explanation — You specified qualifiers that are mutually exclusive. For example, you can specify only one TRANSFER mode qualifier (/ASCII, /BINARY, /FORTRAN, or /IMAGE) in a command. Also, you can specify /REMOTE or /TERMINAL on only one side of a command.

User action — Correct the qualifiers and retry the operation.

FILSYNTAXERR, error in file specification

Explanation — You specified a file containing syntax that was invalid for the RT-11 system. For example, the file name might contain more than six characters.

User action — Correct the syntax of the file specification for the RT-11 system.

ILLBINFORMAT, illegal binary record format

Explanation	You attempted to transfer a file from the RT-11 system to the host system using the /BINARY format qualifier. The file either was not a binary file or was a binary file with a bad format.
User action	Make sure the specified record format qualifier agrees with the actual file record format. Retry the operation.

INVCMDSYNTAX, invalid command syntax

Explanation	The TRANSFER utility command contained invalid syntax.
User action	Correct the syntax and reenter the command.

INVINPUTQUAL, invalid input qualifier /'qualifier'

Explanation	The indicated input qualifier is invalid in the command.
User action	Correct the qualifier and reenter the command.

INVQUAL, invalid qualifier /'qualifier'

Explanation	The indicated qualifier is invalid in the command.
User action	Correct the qualifier and reenter the command.

INVQUALVAL, invalid value for /'qualifier' qualifier

Explanation	You specified an invalid value for the indicated qualifier.
User action	Check the range of valid values for that qualifier. Correct the value for the qualifier and reenter the command.

NOVTCOM, VTCOM not running on remote

Explanation	VTCOM is not running on your RT-11 stand-alone system, or the host system response time is slow due to heavy usage. TRANSFER timed out before receiving a response from VTCOM.
User action	Make sure VTCOM is running on the RT-11 system. Decrease the load on the host system or lower the baud rate between the RT-11 stand-alone system and the host. Retry the operation.

OPENIN, error opening 'input-file' as input

Explanation	The indicated input file cannot be opened. This message is usually accompanied by an RMS message indicating the reason for the failure.
User action	Take corrective action based on the associated message.

OPENOUT, error opening 'output-file' as output

Explanation	The indicated output file cannot be opened. This message is usually accompanied by an RMS message indicating the reason for the failure.
User action	Take corrective action based on the associated message.

QUOTNOOUTPUT, quoted string specification as input needs output specification

Explanation	You did not specify the output (RT-11 only) file, when the input (RMS) file was a quoted string specification.
User action	Explicitly specify the output file when the input file is a quoted string specification.

READERR, error reading 'input-file'

Explanation	The indicated input file cannot be read. This message is usually accompanied by an RMS message indicating the reason for the failure.
User action	Take corrective action based on the associated message.

REMABORT, file transfer aborted by remote

Explanation	The file transfer was aborted due to an I/O error on the RT-11 system, or the VTCOM RESET command was issued.
User action	Check the procedures listed in Section 2 of the RT-11 System Message Manual for recovery from hard error conditions.

REMACCESS, error accessing remote

Explanation	A hard error occurred on the host system during a file transfer.
User action	Retry the operation after performing hard error recovery procedures on the host system.

REMCLOSEIN, error closing 'input-file' as input on remote

Explanation	TRANSFER encountered an error while closing the indicated input file on the RT-11 system. This message is usually accompanied by another message indicating the reason for the failure.
User action	Take corrective action based on the associated message.

REMCLOSEOUT, error closing 'output-file' as output on remote

Explanation	TRANSFER encountered an error while closing the indicated output file on the RT-11 system. This message is usually accompanied by another message indicating the reason for the failure.
User action	Take corrective action based on the associated message.

REMOPENIN, error opening 'input-file' as input on remote

Explanation	TRANSFER cannot open the indicated file on the RT-11 system. This message is usually accompanied by another message indicating the reason for the failure.
User action	Take corrective action based on the associated message.

REMOPENOUT, error opening 'output-file' as output on remote

Explanation	TRANSFER cannot open the indicated output file on the RT-11 system. This message is usually accompanied by another message indicating the reason for the failure.
User action	Take corrective action based on the the associated message.

REMREADERR, error reading 'input-file' on remote

Explanation	TRANSFER cannot read the indicated input file on the RT-11 system. This message is usually accompanied by another message indicating the reason for the failure.
User action	Take corrective action based on the associated message.

REMTIMEOUT, remote timed out during file transfer

Explanation	TRANSFER timed out during a file transfer.
User action	Make sure VTCOM is running. Retry the operation.

REMWRITEERR, error writing 'output-file' on remote

Explanation	TRANSFER encountered an error while writing the indicated file on the RT-11 system. This message is usually accompanied by another message indicating the reason for the failure.
User action	Take corrective action based on the associated message.

SENSEMODE, couldn't read terminal characteristics

Explanation	TRANSFER could not read the terminal hardware characteristics required for a file transfer operation.
User action	Retry the operation after performing hard error recovery procedures on the host system.

SETMODE, couldn't write terminal characteristics

Explanation	TRANSFER could not set the terminal hardware characteristics required for a file transfer operation.
User action	Retry the operation after performing hard error recovery procedures on the host system.

TERMINIT, error initializing terminal for I/O

Explanation	TRANSFER could not initialize the terminal hardware on the host system for a file transfer operation.
User action	Retry the operation after performing hard error recovery procedures on the host system.

TOOBIG, record too large for file's I/O buffer

Explanation	You attempted to transfer a Files-11 record that was too large for TRANSFER's file buffer.
User action	Make sure you are transferring a file in the correct mode or transfer the file in IMAGE mode.

TOOLONG, 'nn' byte record exceeds output file's maximum record length

Explanation	You specified a record length that was too small to hold the largest record from the RT-11 file.
User action	Specify a record length large enough to hold the largest record from the RT-11 file or transfer the file using variable-length records.

USERABORT, file transfer aborted by user

Explanation	You aborted the file transfer by typing CTRL/C, or an I/O error occurred on the host system.
User action	If an I/O error occurred, perform hard error recovery procedures on the host system. Retry the operation.

WRITEERR, error writing 'output-file'

Explanation	RMS encountered an error while writing the indicated file. This message is usually accompanied by another message indicating the reason for the failure.
User action	Take corrective action based on the associated message.

APPENDIX A

NEW ERROR MESSAGES

A.1 V5.4 ERROR MESSAGES

The following error messages are new for RT-11 V5.4.

?BUP-E-Directory input error <dev:>

BUP detected a hardware problem with the input device. For example, you inserted a floppy diskette incorrectly or your magtape device is off line.

Correct your hardware problem and respond YES to the BUP prompt.

?BUP-F-Enough space on one volume - use COPY/DEVICE

The backup operation input fits on one output volume.

Use the COPY/DEVICE command to back up the file(s). Refer to the RT-11 System User's Guide for information on how to use the COPY/DEVICE command.

?BUP-W-Bad block read at <nnnnnn>

A bad block was detected at block number <nnnnnn> during a backup or restore operation. The bad block was not copied.

During restoration, check the validity of the date in block n.

?BUP-W-Directory output error DEV:

On a backup operation to magtape, BUP could not successfully write the volume header or volume ID label to the magtape.

Mount another magtape and retry the operation.

?DUP-E-Output device is smaller than input device
?DUP-E-Output device is truncated

You issued a COPY/DEVICE/NOQUERY command, and the output device is smaller than the input device. The operation completes, but input data is truncated (lost) on the output device.

Mount an output device large enough to receive all the data from the input device.

?DUP-E-Output device is smaller than input file
?DUP-E-Output device is truncated

You issued the command COPY/DEVICE/FILES/NOQUERY, and the output device is smaller than the input file. The operation completes, but input data is truncated (lost) on the output device.	Mount an output device large enough to receive all the data from the input file.

?DUP-E-Output file is smaller than input device
?DUP-E-Output file is truncated

You issued the command COPY/DEVICE/FILES/NOQUERY, and the output file is not large enough to receive the entire input device. The operation completes, but input data is truncated (lost) on the output file.	Perform a SQUEEZE operation to consolidate all available free space on the output volume. If the SQUEEZE operation does not produce enough free space, mount another output volume with enough free space to contain the entire input device.

?DUP-W-Output device is larger than input device

You issued a COPY/DEVICE/NOQUERY command, and the output device is larger than the input device. The operation completed.	The number of blocks you can address on the output device is reduced, because the /DEVICE option copies the home and directory blocks from the input device to the output device.

?DUP-W-Output device is larger than input file

You issued the command COPY/DEVICE/FILES/NOQUERY, and the device containing the output file is larger than the input file. The operation completed.	The number of blocks you can address on the output device is reduced, because the /DEVICE option copies the home and directory blocks from the input device to the output device containing the output file.

?LD-W-Device not installed <DEV:>

You attempted to access a logical disk mounted on a device that is not installed.	This is a warning message. Install the device on which the logical disk is mounted.

?MDUP-F-Insufficient memory, bypassing automatic installation

You attempted to perform the automatic installation (AI) procedure on a processor that does not contain enough memory to support AI with this media.

The automatic installation procedure stopped, and the procedure to manually install your media has begun. The asterisk prompt (*) you see on your terminal screen is from MDUP.MU; that is, MDUP.MU is now loaded in memory. See Chapter 8 of the RT-11 Installation Guide for complete information on manually installing your media.

?PIP-F-Directory I/O error<dev:filnam.typ>

The PIP utility experienced an I/O error when accessing the directory for the indicated volume.

The I/O error is caused by a write-protected output volume or a corrupted directory structure on the input or output volume. Check if the output volume is write protected. If it is, remove the write protection. If the output volume is not write protected, attempt to recover data on the volume with the corrupted directory, using the methods described in Chapter 9 of the RT-11 Software Support Manual.

?SYSGEN-E-Ethernet handler support available for XM only

You requested support for an Ethernet handler (NC, NQ, or NU) while performing a SYSGEN for a monitor other than XM.

You can request Ethernet handler support only when performing a SYSGEN to create an XM monitor. If you require Ethernet support then SYSGEN for an XM monitor or use the distributed XM monitor.

?VTCOM-F-Clock not running

You attempted to start VTCOM on a processor that does not have a running system clock.

Enable the system clock on your processor and restart VTCOM.

A.2 V5.3 ERROR MESSAGES

The following error messages are new for RT-11 V5.3.

?BA-F-Must type 'R BATCH',type '/U'

An UNLOAD BA command was entered before the handler was unlinked.

Run BATCH and specify the /U option to unlink the handler. BATCH automatically performs the UNLOAD BA command.

?BUP-F-Too many mismatches during verify operation

The BUP verification procedure encountered an unacceptably high number of mismatches between the input and output data.

In most cases, the mismatches are caused by bad blocks on the input device. If this error occurs, check the input device for bad blocks. By default, BUP returns this error if it finds more than 25 decimal mismatches. To change this number, perform the customization patch procedure described in Section 2.8.

?BUP-W-Bad buffer read

BUP encountered a bad block on the tape during a restoration from tape to disk.

When BUP returns this error, it copies the rest of the data in its buffer to the disk. Check the validity of the data on the disk. It may not be what you want.

?BUP-W-Bad block read at n

A read error occurred at block n during a backup operation from disk to tape.

During restoration, check the validity of the data in block n.

?BUP-W-Verification error at n

The BUP verification procedure encountered a mismatch between the input and output data at block n of the input device.

This error is usually caused by a bad block located near block n. Check the output data. It may not be what you expect.

?SL-F-SL cannot be unloaded while running

You attempted to UNLOAD the SL handler while SL was running.

Enter the command SET SL OFF, and then UNLOAD the SL handler.

?SYSGEN-F-Next default CSR address out of range

SYSGEN calculated that the next default CSR address is out of range.

SYSGEN determines the default CSR address from the previous CSR address you entered. Because the next default is out of range, SYSGEN asks the questions again (restarts the query sequence) for that device.

?SYSGEN-F-Next default VECTOR out of range

SYSGEN calculated that the next default VECTOR address is out of range.

SYSGEN determines the default VECTOR address from the previous VECTOR address you entered. Because the next default is out of range, SYSGEN asks you the questions again (restarts the query sequence) for that device.

?SYSGEN-W-Address not even

SYSGEN encountered an odd number for a CSR or VECTOR address.

SYSGEN will ask for the CSR or VECTOR address again. Enter an even number.

APPENDIX B

SOFTWARE PERFORMANCE REPORT (SPR) GUIDELINES

Each new installation is provided with Software Performance Report (SPR) forms. The SPR form lets you report problems with, or suggest enhancements to, DIGITAL software and documentation. If you encounter a problem, complete an SPR and mail it to the local SPR center (see the inside back cover of the SPR form).

DIGITAL responds to the name and address that you indicate on the form. You can obtain additional SPR forms by writing to the local SPR center. SPR response is provided at no charge for 90 days after installation and can be continued by subscription thereafter.

DIGITAL requests that you include the following information in all SPRs:

- The CPU type.
- The system device type and the unit on which it was mounted.
- The amount and type of memory.
- The configuration of the system at the time of the problem. Determine the configuration by using the SHOW ALL monitor command. If your configuration includes a hard-copy terminal, send a listing created by the command SHOW ALL. If your configuration does not include a hard-copy terminal, issue the SHOW ALL command, then write down and submit any information from the report that seems relevant to the problem.
- The release and version numbers of all RT-11 image programs that you think are involved with the problem. Use the RESORC utility, with the new /V option, to determine the release and version numbers in the following manner:

  ```
  .RESORC filespec/V <RET>
  ```

  where filespec is an image program from Version 5.1 or later.

  The new RESORC /V option does not report release and version numbers for ASCII or .OBJ programs, but only for image programs such as those with file types .SAV, .REL, and .SYS.

  RESORC prints a single-line report on your terminal. Repeat the command for each RT-11 image program. Write down the report for each program, and submit the reports with your SPR.

- If the RT-11 monitor was created using the SYSGEN process, submit the link map and answer (.ANS) file produced by SYSGEN.

SOFTWARE PERFORMANCE REPORT (SPR) GUIDELINES

- If possible, submit a machine-readable copy of your monitor, your utilities, test files, programs, or command streams that can be used to reproduce the problem. Preferred media are diskettes. Request that the media be returned if you want it back.

  If the problem requiring an SPR involves a program that is not part of the RT-11 distribution, send media containing the program. The program should contain comments. MACRO-11 and FORTRAN programs should use the standard RT-11 programmed requests and SYSLIB routines, as appropriate.

- If the problem resulted in a system halt, include with the SPR:

  The display given by the SHOW MEM command

  The contents of the general registers at the time of halt

  The top 16 words of the stack

  The contents of memory locations around the halt or the error location

  The contents of low memory locations 44 through 56

  The contents of the processor status word

  Any other system information that may relate to the problem

- If your configuration includes hardware from a vendor other than DIGITAL, include a description of the hardware and where it is located in your configuration.

- Any additional information you think may be relevant. A clear and precise SPR can expedite the response.

APPENDIX C

INSTALLING A WORKING SYSTEM ON THE VIRTUAL DEVICE (VM)

This appendix describes a procedure you can follow to run a working RT-11 system from virtual memory. This procedure assumes you have 512K bytes of extended memory. You may be able to use this procedure with some success on systems having less virtual memory, but your working system will have less functionality. Further, this procedure assumes you intend to run the XM monitor from your VM device.

You will notice an obvious performance increase in your processor when you use VM as your system device.

NOTE

The VM device uses extended virtual memory as a disk; that is what gives VM its speed. VM uses volatile memory for data storage; it is not permanent. Any files residing on your VM device can be destroyed when you hard boot your system or if power is disrupted to your system. Therefore, a good practice is to copy only RT-11 files or your programs to VM. Create and edit files and data on permanent media, such as your hard disk.

The device handlers shown in this procedure are appropriate for Professional series processors. You can use this procedure on other processors by replacing device handlers appropriate to Professional series processors with device handlers appropriate to your processor.

This appendix is divided into three main sections:

- The first section describes in some detail the steps necessary to create a model working system. That model contains the utilities KEX, RESORC, SETUP, SL, SPOOL, VBGEXE, and VTCOM.
- The second section describes how to run RT-11 from your VM device.
- The third section describes how to create and run other working systems from your VM device. It uses working systems appropriate for MACRO program development, performing the SYSGEN procedure, and a minimal general purpose working system as models.

The files shown in this appendix work as documented. DIGITAL recommends that you create and run the following VM working system as practice before you modify it (or create another) for your own requirements. Run KEX to demonstrate the increased performance available with the VM system device.

C.1 CREATING A WORKING SYSTEM FOR THE VM DEVICE

Perform the following steps to create a working system for VM (each step is described further in this section):

1. Create a start-up command file that creates a minimal working system on the device your processor first boots. Do that by editing the file STARTX.COM.

2. Create a start-up command file that creates your working system on VM. Name that file STARTV.COM.

3. Create an indirect command file that initializes VM and copies your working system to VM. Name that file VM.COM.

C.1.1 Create the Initial Start-up Command File (STARTX.COM)

STARTX.COM is the file your processor uses when you first boot your processor. Assuming you will be running your system from VM, you should edit the start-up file STARTX.COM for minimal functionality. (The following procedure preserves your current STARTX.COM file, so you will not destroy the contents of that file if you have already edited it and want to keep that edited file.) Perform the following:

1. COPY the file STARTX.COM to STARTY.COM. You will use the file STARTY.COM in the following section.

2. Edit STARTX.COM to resemble the following example:

```
ASS LS LP
ASS DW DK
SET LS SPEED=4800  ! Set to your printer's baud rate
SET LS ENDPAG=1,NOFORM0
RUN SETUP
VT100,JUMP,LIGHT,CLEAR
```

That example sets some printer and terminal characteristics. Those characteristics might not be ones you use. You can set any printer and terminal characteristics you want. See the RT-11 System User's Guide for explanation of all SET and SETUP commands. Include those commands you want.

C.1.2 Create the VM Start-up file STARTV.COM

You have already edited the start-up command file for the device your processor initially boots - STARTX.COM. You must now create a start-up command file for the VM device - STARTV.COM.

VM runs the file STARTV.COM when you boot VM. Therefore, STARTV.COM should contain the start-up commands you normally use in your working system. You create STARTV.COM using the following command:

```
.COPY STARTY.COM STARTV.COM
```

Keep in mind the following points when editing STARTV.COM:

- Be sure to delete the command INITIALIZE/NOQUERY VM: from STARTV.COM. Inclusion of that command destroys the contents of VM when you boot VM.

- Your hard disk will no longer be your system device and therefore will no longer be automatically loaded. Any device used by a system job (such as KEX, VTCOM, or SPOOL) must be loaded. Be sure to LOAD your hard disk device handler (DW) if you are going to be using system jobs. If you are going to transfer files using VTCOM between a host and your diskette drive, you should also LOAD DZ.

- You should ASSIGN a default data device (probably your hard disk).

Provision for those points is made in the following example VM start-up command file. This file performs the following functions:

- Enables the SL editor.

- Runs VTCOM as a system job and sets the controller baud rate to 1200 (set the value you use).

- Runs SPOOL as a system job and sets the serial line printer baud rate to 4800 (set the value you use). Flagpage support is disabled and the spooler is set to append a form feed to the final page of each file you spool.

- Sets terminal hardware characteristics.

- Assigns DW as the default data device.

- Loads DW (for system jobs).

- Shows the memory listing and the date.

Using the KEX editor, open the file STARTV.COM and edit it to appear as follows or as you prefer:

```
!
SET SL KMON
!
SRUN SY:VTCOM.SAV/PAUSE
LOAD XC=VTCOM
RESUME VTCOM
SET XC SPEED=1200
!
SRUN SPOOL/PAUSE
LOAD LS=SPOOL
RESUME SPOOL
ASS SP0 LP
ASS SP0 LP0
ASS SP0 LS
ASS SP0 LS0
SET LS SPEED=4800
SET SP FLAG=0,ENDPAG=1
!
SETUP VT100,JUMP,LIGHT,CLEAR
ASS DW DK
LOAD DW
SHOW MEMORY
DATE
```

C.1.3 Create VM.COM to Initialize and Load VM

Create an indirect command file named VM.COM using the KEX editor. VM.COM initializes the VM disk and copies your working system to that disk.

The following files must be included on any working system. They are the minimum files necessary to run RT-11 on the VM device:

```
SWAP.SYS            ! Should be copied first
RT11XM.SYS          ! Should be copied second
PIX.SYS             ! Should be copied third
VMX.SYS             ! Should be copied fourth
DWX.SYS             ! Required to communicate with hard disk
DZX.SYS             ! Required to communicate with diskettes
PIP.SAV             ! These three utilities should follow
DUP.SAV             !  immediately after
DIR.SAV             !  all .SYS files
```

(PIX.SYS,DWX.SYS, and DZX.SYS are specific to Professional series processors. The other files are required by all processors.)

C.1.3.1 **Modifying the Base Address of VM** - You must lower the base address of the VM region in virtual memory to create enough blocks to hold your working system. You should also leave enough virtual memory to run any jobs that require it. Using the following values creates a virtual device (VM) of 686 blocks and leaves approximately 50K words of virtual memory. That is a large enough virtual device to hold a functional working system and enough unused extended memory to run KEX, VTCOM, and SPOOL as system jobs.

The following commands are included in VM.COM to modify the base address of the VM device:

```
SET VM SIZE=0
SET VM BASE=5000
UNLOAD VM
REMOVE VM
INSTALL VM
```

Those commands set the size of the VM device to its default value, that is, from the base address to the top of available memory. Those commands also set the base address of VM at 500000(octal), and VM resides at that address when VM.COM loads VM in virtual memory. You will notice that those commands, with various base address values, are included in all the command files that copy working systems to VM. Once you have changed the base address of VM, you must boot a VM with that base address. Therefore, those commands should be included in all command files that copy working systems to VM. That avoids any possible conflict in VM base addresses.

C.1.3.2 **The Contents of VM.COM** - The following indirect command file copies the necessary system (.SYS) files and utilities to run RT-11 on a Professional series processor. VM.COM also copies the useful utilities KEX, RESORC, SETUP, SL, SPOOL, VBGEXE, and VTCOM. The last line copies the VM device start-up command file, STARTV.COM, to the VM device and renames it STARTX.COM. That lets VM find its start-up command file when you boot VM.

Use the KEX editor to create the file VM.COM. Enter the following commands in that file:

```
SET VM SIZE=0
SET VM BASE=5000
UNLOAD VM
REMOVE VM
INSTALL VM
LOAD VM:
INIT/NOQ VM:
COPY/SYS SWAP.SYS VM:
COPY/SYS RT11XM.SYS VM:
COPY/SYS (PIX,VMX,DWX,DZX,XCX,LSX).SYS VM:
COPY/SYS (SLX,SPX).SYS VM:
COPY (DUP,DIR,PIP,KEX,RESORC,SETUP).SAV VM:
COPY (VBGEXE,VTCOM).SAV VM:
COPY SPOOL.REL VM:
COPY/BOOT VM:RT11XM.SYS VM:
COPY STARTV.COM VM:STARTX.COM
```

C.2 RUNNING RT-11 FROM THE VM DEVICE

This section assumes you have performed the steps described in Sections C.1 through C.1.3.2. Use the following steps to run RT-11 from your VM device:

1. If you are now running your processor, close any files you have open. Boot your processor:

   ```
   .BOOT SY:  <RET>
   ```

   Your processor executes the commands contained in the start-up file STARTX.COM.

2. After your processor has booted, issue the following command:

   ```
   .$@VM <RET>
   ```

   That command runs the indirect command file VM.COM. If VM.COM does not run to completion (execute all commands in the file), repeat steps 1 and 2. You might have previously assigned the size of the VM device (SET VM SIZE=nnnnn command) to some nonzero value, causing VM.COM to abort for any of a number of reasons. You should not have this problem again unless you assign the size of the VM device to some nonzero value at a later time. If you do that, you will simply have to repeat this correction.

3. When you receive the monitor prompt (.), boot the VM device:

   ```
   .BOOT VM:  <RET>
   ```

   After you have booted VM, examine the directory:

   ```
   .DIR SY:  <RET>
   ```

   If the size of the VM directory is not 686 blocks (combined used and free blocks), you probably have some variation of the problem described in Step 2. Repeating steps 1, 2, and 3 should correct the problem.

You are now running the XM monitor from your VM system device. Running your working system from VM is different than running from your hard disk in at least two ways:

- VM is faster than your hard disk. The increase in performance of your processor will be especially apparent with jobs such as KEX.
- The system device is smaller and therefore contains fewer utilities. That is not normally a problem as the device is large enough to contain the set of utilities you use when performing particular work, such as writing and editing or programming. If you perform more than one type of work with your processor, you may want to create more than one working system for your VM device. See Section C.3 for a discussion of multiple VM system devices.

C.3 MULTIPLE WORKING SYSTEMS FOR THE VM DEVICE

A working system containing the utilities KEX, VTCOM, and SPOOL would be appropriate for text editing, maintaining communications with a host processor, and so forth. That working system, however, would not be appropriate for writing and debugging programs.

If you want to take advantage of the increased performance you attain by running RT-11 from the VM device and you perform more than one type of work with your processor, you should create multiple working systems for the VM device in the following manner:

1. Determine the functionality you require; that is, the utilities you want to run.
2. Determine how many blocks those utilities need on the VM device. Remember to include the minimum set of system (.SYS) files and utilities that RT-11 requires to run. Those are listed in Section C.1.3. A useful technique is to copy all the files to an initialized diskette, boot that diskette and examine its size.
3. Determine the base address for the VM device for that working system. Setting the base indicated below produces a VM device with the indicated number of blocks on Professional series processors:

Base	Number of Blocks
7000	558
6000	622
5000	686
4000	750
3000	814

   Remember that you have a finite amount of virtual memory. Therefore, do not create a VM device that is larger than you need. (The larger the VM device, the smaller the amount of remaining virtual memory.) You do not need to specify the exact values given above; the value 4200 is valid for example.
4. Create a start-up command file.
5. Create an indirect command file to set the VM device base, initialize VM, and copy your files to VM.

The following three sections describe creating and running different types of working systems for the VM device.

C.3.1 A MACRO Programmer's Working System

The following indirect command file might be appropriate for creating a working system for writing, assembling, linking, and debugging MACRO programs. The base address of VM is changed to 4000, creating a 750 block VM device. That is done to provide enough free blocks for the MACRO and LINK work files. Perform the following steps to create a MACRO programmer's working system:

1. Create a start-up command file using the KEX editor and name it STARTP.COM. Include in that file all commands you want executed when you boot this working system. The following file might be appropriate; use the proper baud rate for your printer.

```
ASS LS LP
SET LS SPEED=4800
ASS VM0 CF
ASS VM0 WF
ASS DW DK
LOAD DW
RUN SETUP
VT100,JUMP,LIGHT,CLEAR
RUN RESORC
/X
^C
```

2. Create the following indirect command file, using the KEX editor and name it VMPRG.COM:

```
SET VM SIZE=0
SET VM BASE=4000
UNLOAD VM
REMOVE VM
INSTALL VM
LOAD VM
INIT/NOQ VM:
COPY/SYS DW:(SWAP,RT11XM,PIX,VMX,DWX,DZX).SYS VM:
COPY/SYS DW:(LSX,SDSX).SYS VM:
COPY DW:(DUP,PIP,DIR,MACRO,LINK).SAV VM:
COPY DW:(KEX,CREF,DBGSYM).SAV VM:
COPY DW:(VDT.OBJ,SYSLIB.OBJ,SYSMAC.SML) VM:
COPY DW:STARTP.COM VM:STARTX.COM
COPY/BOOT VM:RT11XM.SYS VM:
```

Then, to use the VM device for writing, assembling, linking, and debugging MACRO programs, perform the following procedure:

1. Close any files you now have open.

2. If you are now running on your hard disk (DW), issue the following commands:

```
.$@VMPG <RET>
.BOOT VM: <RET>
```

If you are now running on your VM device, issue the following commands:

```
.BOOT DW: <RET>
.$@VMPG <RET>
.BOOT VM: <RET>
```

C.3.2 A Working System for Performing a SYSGEN

You can decrease the time required to perform a SYSGEN by running RT-11 from the VM device. Use the following procedure:

1. Use the KEX editor to create a start-up command file named STARTG.COM. Include the following command lines in that file:

```
ASS LS LP
SET LS SPEED=4800      !Use your printer's baud rate
SET LS ENDPAG=1,NOFORM0
ASS DW DK
LOAD DW
ASS VM0 CF
ASS VM0 WF
RUN SETUP
VT100,JUMP,LIGHT,CLEAR
RUN RESORC
/X
^C
```

2. Use the KEX editor to create an indirect command file named VMGEN.COM. Include in that file the following command lines:

```
SET VM SIZE=0
SET VM BASE=5000
UNLOAD VM
REMOVE VM
INSTALL VM
LOAD VM
INIT/NOQ VM:
COPY/SYS DW:(SWAP,RT11XM,PIX,VMX,DWX,DZX).SYS VM:
COPY/SYS DW:LSX.SYS VM:
COPY DW:(DUP,PIP,DIR,MACRO,LINK).SAV VM:
COPY DW:IND.SAV VM:
COPY DW:(SYSMAC.SML,SYSLIB.OBJ) VM:
COPY DW:STARTG.COM VM:STARTX.COM
COPY/BOOT VM:RT11XM.SYS VM:
```

Then, to perform the SYSGEN procedure while running RT-11 from your VM device:

1. Close any files you now have open.

2. If you are now running on your hard disk (DW), issue the following commands:

```
.$@VMGEN <RET>
.BOOT VM: <RET>
```

If you are now running on your VM device, issue the following commands:

```
.BOOT DW: <RET>
.$@VMGEN <RET>
.BOOT VM: <RET>
```

3. Once you have booted the VM device, start the SYSGEN procedure using the following command:

```
.RUN SY:IND.SAV SYSGEN.COM <RET>
```

Perform the procedure as documented in the RT-11 System Generation Guide. At the end of the SYSGEN dialog, be sure to specify your hard disk (DW0) as the device that contains the source input files and as the device to which you want to send all output files.

C.3.3 A Minimal Working System

The following working system contains only those files necessary to boot the VM device. The base address of VM is set to 5000, giving a device size of 686 blocks. The procedure uses the minimum functionality STARTX.COM start-up command file you created in Section C.1.1.

You then copy programs to the VM device and thereby increase the performance attainable by those programs. Perform the following steps to create a general purpose working system:

1. You have already created the start-up command file STARTX.COM.

2. Create the indirect command file, VMMIN.COM, using the KEX editor. Include the following commands in that file:

```
SET VM SIZE=0
SET VM BASE=5000
UNLOAD VM
REMOVE VM
INSTALL VM
LOAD VM
INIT/NOQ VM:
COPY/SYS DW:(SWAP,RT11XM,PIX,VMX,DWX,DZX).SYS VM:
COPY/SYS DW:LSX.SYS VM:
COPY DW:(DUP,PIP,DIR).SAV VM:
COPY DW:STARTX.COM VM:
COPY/BOOT VM:RT11XM.SYS VM:
```

Then, to use the VM device as a high-speed general purpose working system, perform the following:

1. Close any files you now have open.

2. If you are now running on your hard disk (DW), issue the following commands:

```
.$@VMMIN <RET>
.BOOT VM: <RET>
```

If you are now running on your VM device, issue the following commands:

```
•BOOT DW: <RET>
•$@VMMIN <RET>
•BOOT VM: <RET>
```

3. Once you have booted the VM device, copy any programs you want to run to it. Then run those programs from the VM device.

HOW TO ORDER ADDITIONAL DOCUMENTATION

DIRECT TELEPHONE ORDERS

In Continental USA and Puerto Rico call 800–258–1710	In Canada call 800–267–6146	In New Hampshire Alaska or Hawaii call 603–884–6660

ELECTRONIC ORDERS (U.S. ONLY)

Dial 800–DEC–DEMO with any VT100 or VT200 compatible terminal and a 1200 baud modem.
If you need assistance, call 800–DEC–INFO.

DIRECT MAIL ORDERS (U.S. and Puerto Rico*)

DIGITAL EQUIPMENT CORPORATION
P.O. Box CS2008
Nashua, New Hampshire 03061

DIRECT MAIL ORDERS (Canada)

DIGITAL EQUIPMENT OF CANADA LTD.
940 Belfast Road
Ottawa, Ontario, Canada K1G 4C2
Attn: A&SG Business Manager

INTERNATIONAL

DIGITAL
EQUIPMENT CORPORATION
A&SG Business Manager
c/o Digital's local subsidiary
or approved distributor

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Northboro, Massachusetts 01532

*Any prepaid order from Puerto Rico must be placed with the Local Digital Subsidiary:
809–754–7575

READER'S COMMENTS

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

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

Did you find errors in this manual? If so, specify the error and the page number.

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

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code or Country _____

— — Do Not Tear — Fold Here and Tape — — — — — — — — — — — — — — — — — —

digital

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
CORPORATE USER PUBLICATIONS
MLO5–5/E45
146 MAIN STREET
MAYNARD, MA 01754–2571

— — Do Not Tear — Fold Here — — — — — — — — — — — — — — — — — —

Cut Along Dotted Line