RT–11
System Release Notes
AA–5286G–TC

digital
software

RT-11
System Release Notes
AA-5286G-TC

April 1985

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

This manual supersedes *RT-11 System Release Notes*, AA-5286F-TC.

Operating System: RT-11 Version 5.2

The following are trademarks of Digital Equipment Corporation:

digital™

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

CONTENTS

iii

CHAPTER 3   CURRENT RESTRICTIONS AND CORRECTED PROBLEMS

CHAPTER 4   INSTALLATION, BOOTSTRAP, AND HARDWARE SETUP PROCEDURES

PREFACE

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

Changes to RT-11 made after the publication of the V5.1 documentation update and those for V5.2 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 V5.2 as well, but references to V5.2 apply only to V5.2.

References to RT-11 documentation other than the RT-11 System Release Notes 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 document 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 – RUNNING PRO/GIDIS UNDER RT-11

  This chapter describes the specifics necessary to run PRO/GIDIS under RT-11. It is to be used with the PRO/GIDIS Manual, AD-Y660A-T1.

ix

- Chapter 6 - GLOBAL REGION SUPPORT IN EXTENDED MEMORY

  This chapter describes the new support for creating global regions in extended memory.

- Chapter 7 - ETHERNET HANDLERS

  This chapter describes the two new Ethernet class handlers, NC and NQ, that support the DECNA and DEQNA Ethernet controllers.

                                    NOTE

  A machine-readable addendum to the RT-11 System Release Notes, V5NOTE.TXT, is provided on the distribution kit. V5NOTE.TXT describes changes that have occurred since this document was printed.

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 Board 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. In RT-11 V5.0, only the FB monitor supported the SBC-11/21, and there was no support for the SBC-11/21 PLUS.

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 byte) 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 V5.2. Although the foreground/background (FB) monitor runs on Professional computers with some restrictions (see Chapter 3 of this manual), 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 now 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 Q-BUS processor system, and the PDP-11/84 UNIBUS processor system.

1.2 NEW MASS STORAGE DEVICES

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

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

1.2.1 RA80 Disk (DU)

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

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

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

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 total storage of 52M bytes per pair. RC25 disks can be used as the system volume or for data storage.

1.2.3 RD5Ø/RD51/RD52 Disks (DU or DW)

The RD5Ø/RD51/RD52 disks are fixed hard disks available on the MicroPDP-11 and Professional 3ØØ series processors.

The MicroPDP-11 can include a 1ØM-byte RD51 disk or a 33M-byte RD52 disk. Each disk is an MSCP device supported by the MSCP disk class handler, DU.

Professional 35Ø and Professional 38Ø systems can include a 5M-byte RD5Ø, a 1ØM-byte RD51, or a 33M-byte RD52. All are supported by the DW handler.

RD5Ø/RD51/RD52 disks can be used as system or data volumes.

1.2.4 RX5Ø Diskette (DU or DZ)

The RX5Ø is a diskette subsystem available on Q-BUS processors such as the MicroPDP-11, UNIBUS processors, and the Professional 3ØØ series computers. The subsystem consists of two drives, each of which holds one 5-1/4-inch diskette. Each diskette provides a storage capacity of 4ØØK bytes (8ØØ blocks).

RX5Ø diskettes on the Q-BUS processors are MSCP (mass storage communication protocol) devices supported by the MSCP disk class handler, DU. On the Professional 325 and 35Ø, RX5Ø 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.

RX5Ø diskettes can be used as system or data storage volumes. The RT-11 V5 distribution kit is available on RX5Ø diskettes.

RX5Ø diskettes are not interchangeable with RXØ1 and RXØ2 diskettes. You can insert an RX5Ø diskette only in an RX5Ø drive.

1.2.5 TK25 Magtape (MS)

The TK25 is a streaming TS11-compatible cartridge magtape drive, operating on the Q-BUS. Data is recorded on a 6ØØ-foot, 1Ø-track (1/4-inch format) magnetic tape. The TK25 stores up to 6ØM 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.

1.2.6 TK5Ø Magtape (MU)

RT-11 V5.2 does not support the TK5Ø magtape at this time. Information on the TK5Ø magtape is supplied in anticipation of future support. Therefore, no information in this document concerning the TK5Ø magtape applies to RT-11 V5.2.

The TK5Ø is a cartridge tape drive that uses the tape class mass storage communication protocol (TMSCP). The TK5Ø stores approximately 1ØØM bytes data on 6ØØ feet of 1/2-inch magtape at a density of 6667 bits/inch. The tape contains 22 tracks.

1.2.7 TS05 (TSU05/TSV05) Magtapes (MS)

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

Data is recorded on nine-track 1/2-inch format magnetic tape. 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 NEW PRINTER SUPPORT

RT-11 V5.2 supports the following new printers.

1.3.1 LA210

RT-11 V5.2 supports the new desk-top LA210 printer. The LA210 is a multifont serial-line dot matrix printer that operates in text and graphics mode. The print matrix and speed are variable in text mode. Graphics mode resolution is up to 330 X 72 dots/inch. Certain hardware characteristics can be set using SETUP commands (see Section 4.4.1).

1.3.2 LN03

RT-11 V5.2 now supports the desk-top LN03 laser printer. The LN03 is a serial-line printer with 300 X 300 dots/inch resolution that prints 8 pages per minute. Certain hardware characteristics of the LN03 can be set using SETUP commands (see Section 4.4.1). No support is provided for graphics (pixel) printing.

The LN03 requires a form feed (<FF>) character at the end of each file printed. To insert a form feed character at the end of each file, SET LS ENDPAG = 1, or if running the SPOOL utility, SET SP ENDPAG = 1.

1.4 OTHER NEW HARDWARE SUPPORT

RT-11 V5.2 supports the following new hardware.

1.4.1 512K-Byte Memory Expansion Card

The 512K-byte memory expansion card is supported for Professional 300 series processors. It adds 512K bytes to the Professional's on-board memory, giving the processor 1024K-byte memory.

1.4.2 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 V5.2 does not provide support for printers on the Mini-Exchange.

See Section 2.35 for using the Mini-Exchange with the virtual terminal communication utility (VTCOM).

1.5 NEW DISTRIBUTION KITS

The format of RT-11 distribution kits has changed because of the new automatic installation and verification procedure (see Section 1.6.1). Refer to Chapter 4 of this manual for procedures for RT-11 V5 distribution kits that will not be installed automatically.

1.6 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)

- Graphics utility support (PRO/GIDIS) (V5.2)

- Indirect control file processor (IND)

- Virtual KED (KEX)

- Logical disk subsetting handler (LD)

- Hardware setup utility (SETUP)

- Single-line editor (SL)

- Transparent spooling utility (SPOOL)

- User commands first utility support (UCF) (V5.2)

- 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 provided in the documentation set.

1.6.1 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 or LA100 series console terminal, or a Professional 300 series system

- One of the following mass storage configurations:

    MicroPDP-11 (RX50/RD51/RD52) - distribution kit resides on RX50 diskettes

    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.

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 and 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.

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.

For the TK50 magtape distribution, you must have at least 256K bytes of memory. Automatic installation of the TK50 magtape distribution requires a Q-BUS processor, such as a MicroPDP-11 or PDP-11/23 PLUS, with an installed MU boot ROM.

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 of this manual for procedures for RT-11 V5 distribution kits that do not support automatic installation.

1.6.2 New System Procedures and Utilities

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

1.6.2.1 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 utilize the 100 in/s streaming mode of the TS05 magtape drive.

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

Changes have been made to the BUP utility for RT-11 V5.2. See Chapter 2 of this manual for details.

1.6.2.2 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. In the following example, the second command shows the CCL equivalent of the first command.

Running a Utility Program

```
.R PIP
*DL1:MYPROG.OLD=DL0:MYPROG.MAC
*
```

CCL Equivalent

.PIP DL1:MYPROG.OLD=DL0:MYPROG.MAC
.

or

.PIP DL0:MYPROG.MAC DL1:MYPROG.OLD
.

See Section 4.6 in the <u>RT-11</u> <u>System</u> <u>User's</u> <u>Guide</u> for more information on CCL.

1.6.2.3 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. It 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 V5.2 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 Chapter 5 of this manual for complete details on running PRO/GIDIS under RT-11.

1.6.2.4 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.

A new directive, .STRUCTURE, has been added to IND for RT-11 V5.2. Some current directives are also changed. Section 2.15 in this manual for information.

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 <u>RT-11</u> <u>Installation</u> <u>Guide</u>.

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

1.6.2.5 **Virtual KED (KEX)** – The virtual KED program (KEX) is available for use under the XM monitor only. 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. In addition, KEX may be run as a foreground or system job, allowing editing to continue while the background is performing some function. On systems that include multiterminal support, multiple copies of KEX may be run (by using the SRUN and FRUN commands), each from its own terminal.

1.6.2.6 **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.

1.6.2.7 **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, most 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.

SETUP modes have been added to RT-11 V5.2. See Chapter 2 of this manual for details.

1.6.2.8 **Single-Line Editor (SL)** – The single-line editor (SL) lets you edit the current keyboard command line or CSI command string 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.

SLMIN.SYS, a smaller SL handler for use under the SJ/FB monitors, is available with RT-11 V5.1.

Two functions, save and get saved, have been added to SL for RT-11 V5.2. See Section 2.27 of this manual.

See Section 4.3 in the RT-11 System User's Guide for instructions on using the single-line editor.

1.6.2.9 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.

1.6.2.10 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, as explained at the end of this subsection.

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 it 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.

The following DCL commands do not function when UCF is enabled:

    B
    CLOSE
    D
    E
    GET
    REENTER
    SAVE
    START

Once you have written a UCF utility, you make it available to KMON by performing the following customization patch. The patch can also be used to selectively inhibit command line processors (DCL, CCL, and UCL) from processing KMON commands. Great care must be taken when inhibiting command line processors; if all are inhibited, no commands are recognized.

In the customization, the octal value of the lower four bits of the value represented in xxxxxx specify the command line processor: 1 for UCF, 2 for DCL, 4 for CCL, and 10 for UCL. The distributed monitors support DCL, CCL, and UCL command line processors; the value represented by xxxxxx is 177416.

To enable the UCF command line processor, set bit 0 (octal value 001) by specifying 177417 for nnnnnn. To selectively inhibit distributed command line processors DCL, CCL, or UCL, clear respective bits 1, 2, or 3 in xxxxxx, by entering the corrected octal value for nnnnnn.

```
.RUN SIPP<RET>
*monitor.SYS/A<RET>
Base?    0<RET>
Offset?  CLIFLG<RET>
    Base        Offset      Old       New
    000000      CLIFLG      xxxxxx    nnnnnn<RET>
    000000      CLIFLG+2    ??????    <CTRL/Y><RET>
*<CTRL/C>
.
```

Creating a UCF.SAV from UCL.SAV

After the customization patch is performed, a functioning UCF.SAV can be created from UCL.SAV. After checking for valid IND construction, the keyboard monitor (KMON) then examines the UCL data file first, rather than last, when looking for valid keyboard commands. That greatly increases the execution speed of UCL commands.

Use the COPY or RENAME commands to create a UCF.SAV from UCL.SAV. The COPY command creates a UCF.SAV and keeps (does not delete) UCL.SAV. The RENAME command creates a UCF.SAV and eliminates (deletes) UCL.SAV.

Execute the following command in response to the monitor prompt (.) to create UCF.SAV by copying UCL.SAV:

    .COPY UCL.SAV UCF.SAV

Or execute the following command in response to the monitor prompt (.) to create UCF.SAV by renaming UCL.SAV:

    .RENAME UCL.SAV UCF.SAV

The created UCF.SAV maintains the same characteristics as the UCL.SAV from which it is created. Modifications to UCL.SAV made by the customization patches and optimization described in Section 2.34 are kept by the created UCF.SAV.

The customization patches and optimization described in Section 2.34 can be applied transparently to the created UCF.SAV.

The only difference between UCL.SAV and the created UCF.SAV is the command used to display the data file. Execute the following command in response to the monitor prompt (.) to display the UCF data file:

```
.UCF<RET>
*TT:=<RET>
```

1.6.2.11 **User Command Linkage Feature** (UCL) - User command linkage (UCL) lets you write your own command parser, so you can create your own commands. UCL support is included in the distributed monitors. The data structure of UCL.DAT (the default data file), and some UCL commands have been redesigned for RT-11 V5.2. See Section 2.34 in this manual for information.

Refer to Section 2.2.4.2 in the RT-11 Software Support Manual for more information on UCL.

A usable example of a UCL command parser is provided on the distribution kit as an executable image (UCL.SAV). See the RT-11 System User's Guide for instructions on how to define commands using the distributed UCL.SAV.

1.6.2.12 **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. Read the file UNSUP.TXT on your distribution kit for details and restrictions.

1.6.2.13 **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 RTEM-11 and the program TRANSF.SAV 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 in the background while you maintain a connection to the host computer.

The RT-11 System Utilities Manual lists requirements for running VTCOM and explains how to use VTCOM and TRANSF.SAV.

Commands have been added to VTCOM for RT-11 V5.2. See Chapter 2 of this manual for details. |

1.6.3 New Programmed Requests

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

1.6.3.1 .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.

1.6.3.2 .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.

1.6.3.3 .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.

1.6.3.4 .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.

1.6.3.5 **.DRINS** – 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, the CSRs checked by the INSTALL keyboard command code, and defines the system and data device installation entry points.

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

1.6.3.6 **.FPROT** – The .FPROT programmed request sets or resets the protection status of files.

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

1.6.3.7 **.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 <u>RT-11 Programmer's Reference Manual</u> for details and examples of the .PEEK programmed request.

1.6.3.8 **.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 <u>RT-11 Programmer's Reference Manual</u> for details and examples of the .POKE programmed request.

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

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

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

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

1.6.3.11 **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 <u>RT-11 Programmer's Reference Manual</u> for more information on the SOB macro.

1.6.4 New System Subroutine Library (SYSLIB) Routines

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

1.6.4.1 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.

1.6.4.2 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.

1.6.4.3 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.

Form: i = IFREER (work[,<type>])

where:

work   is a 7-word work area block. The first five words of the
        work area block contain information from the region
        definition block (RDB):

        A unique region identification (R.GID)

        The size of the region (R.GSIZ)

        The region status word (R.GSTS)

        The region name in two RAD50 words (R.NAME,
            R.NAME+2)

        The last two words in the work block area are reserved by
        RT-11.

        The work area specified in IFREER must be the same as the
        IGETR work area.

type   is eliminate. Eliminate can be specified using the first
        letter "e". If you do not specify the type argument, you
        detach but do not eliminate the global region.

Errors:

    i =   0   Normal return (success)

      =   -3   An invalid region identifier was specified

      =   -15   Global region in use

      =   -19   First character of type argument invalid (not "e")

      =   -20   Required work argument is missing

The following errors can be returned by IFREER 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 RMON

      = -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

1.6.4.4 **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 it or by calling a specified subroutine.

IGETR does not fetch handlers. Any handler required by I/O in IGETR must be loaded or fetched by the program.

Form: i = IGETR (work,char,name,addr[,csize] [,offset[,msize]] {[,chan[,blk]] })
                                                                    {[,file[,blk]] }
                                                                    {[,sbrtn,-1] }

where:

work   is a 7-word work area block. The first five words of the work area block contain information from the region definition block (RDB):

A unique region identification (R.GID)

The size of the region (R.GSIZ)

The region status word (R.GSTS)

The region name in two RAD50 words (R.NAME, R.NAME+2)

The last two words in the work block area are reserved by RT-11.

The work area specified in IGETR must also be the work argument specified in IFREER.

char   is a character constant specifying the type of ownership of the global region. Only the first letter of the character constant need be specified, and it must be enclosed in single quotes ('). Specify one of the following:

'private' Program solely owns global region

'shared'   Global region available to other programs

'age'   Enables automatic global elimination

name   is the 2-word name of the global region in six RAD50 characters.

addr   is a variable specifying the global region's base address. The base address must begin on a PAR boundary (4K-word multiples beginning at 000000).

csize   is the size of the global region you want to create, expressed in words. If you specify csize as zero or omit it, the actual global region size is used. Specifying zero for csize is invalid unless the glob 1 region already exits.

msize   is the number of words you want mapped to the global region. If you specify msize as zero or omit it, you map the whole global region.

offset   is the offset from the beginning of the global region, expressed in units. A unit is 64(decimal) bytes. The offset is the number of units you skip before mapping begins. If you specify offset as zero or omit it, you begin mapping at the beginning of the global region.

chan   is a channel opened on a file from which to read initialization data. When specifying chan, the argument value must be from 0 through 255(decimal), and the blk argument cannot be -1.

file   is a pointer to a 4-word data block. The last three words contain a device and file specification for a file containing initialization data to open and read. If the device specification is valid, the first word contains a value greater than 255(decimal).

The value in the blk argument cannot be -1.

sbrtn   is the name of a subroutine that initializes the global region. The addr and msize arguments are passed to that subroutine.

The value in the blk argument must be -1.

blk   is the number of the first block to use in the file that initializes the global region. Specify a zero value in this argument to load from the beginning of the file.

The value must be -1 if blk is coupled with the sbrtn argument.

The work, name, addr, csize, wsize, offset, chan, file, and blk arguments are INTEGER*2 values. The sbrtn argument is the EXTERNAL type.

Errors:

i =	0	Normal return (success)
=	-1	Invalid addr alignment (detected by IGETR)
=	-2	No window definition block for .CRAW
=	-3	Any .CRAW error except no window definition block (-2)
=	-4	End-of-file on .READW
=	-5	I/O error on .READW
=	-6	Channel closed when .READW attempted; channel not available from IGETR
=	-7	No region control block for .CRRG
=	-8	Insufficient memory for .CRRG
=	-9	Reserved
=	-10	Memory too fragmented to return at .ELRG

= -11   Global region not found (and no nonzero size specified)

= -12   No global region control block for .CRRG

= -13   Reserved

= -14   Reserved

= -15   .LOOKUP found channel already open

= -16   .LOOKUP could not find requested file

= -17   .LOOKUP found device in use and not shareable

= -18   Any .ELRG error except memory too fragmented (-10)

= -19   First character of char argument invalid (not 'p', 's', or 'a')

= -20   Required argument missing (work, char, name, or addr)

The following errors can be returned by IGETR 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 RMON

= -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

1.6.4.5 **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.

1.6.4.6 **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.

1.6.4.7 **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.

1.6.5 **New Handlers**

RT-11 V5 includes the following new handlers.

1.6.5.1 **DU (MSCP Handler)** - The DU handler supports disk systems that implement the mass storage communication protocol (MSCP). Presently, RT-11 supports five MSCP storage systems: RC25 disk, RA80 disk, and on the MicroPDP-11, RX50 diskette, RD51 and RD52 disks.

The DU handler can be modified, by using SET commands, to address MSCP disks by unit number, partition number, and port number. (A port is the same as an individual controller.)

MSCP supports unit numbers 0-255. However, since RT-11 supports only unit numbers 0-7, you can map a given RT-11 unit number to any specific MSCP unit number using the SET DUn UNIT=x command. RT-11 unit n is translated to MSCP unit x.

MSCP supports disks larger than 64K blocks. Under RT-11, large disks are divided into 64K-block partitions. The DU handler supports up to 255 partitions on a single disk, with each partition referenced as though it were an RT-11 disk. The SET DUn PART=x command translates a DU unit number (n) to a particular partition (x) on a disk.

The DU handler also supports up to four separate controllers or ports, numbered 0-3. The command SET DUn PORT=x defines which port (x) is accessed when DU device unit n is referenced. An alternative method of supporting multiple ports, with some potential performance advantages, is to create a duplicate copy of the DU handler (DU.SYS) under another file name for each additional controller.

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

1.6.5.2 DW (Professional 350 and 380 Hard Disk Handler) - The DW handler supports 5M-byte RD50, 10M-byte RD51, and 33M-byte RD52 fixed hard disks for the Professional 350 and 380 computers. RT-11 supports only drive unit DW0.

1.6.5.3 DZ (Professional 325 and 350 Diskette Handler) - The DZ handler supports RX50 diskette drives for the Professional 300 series computers. RT-11 supports drive units DZ0 and DZ1.

1.6.5.4 MU (TMSCP handler) - The MU handler supports magtape systems that use the tape mass storage communication protocol (TMSCP).

Device Identification (.DSTATUS)

The MU handler causes the .DSTATUS programmed request to return a device identification of 60 "TMSCP class magnetic tape handler".

File Structure

The file structure magtape module (FSM) is included in the distributed MU handler, so MU supports the RT-11 file structure. You can remove the FSM portion of the handler during system generation if you require only hardware support. That option lets you write your own file structure support.

SET Commands

The MU handler supports the following commands:

```
SET MUx:   CSR=n
SET MUx:   VECTOR=n
```

See Section 2.3.3 of this manual for a description of these commands.

Use of Special Functions (SPFUN)

The MU handler uses the same special functions (SPFUN) and special function codes as other file-structured magtape handlers, except it supports a new SPFUN 360. SPFUN 360, when specified, bypasses all unit number translation and allows direct access to the TMSCP port. SPFUN 360 (direct TMSCP access) serves the same purpose as SPFUN 371 (direct MSCP access) used by the DU handler.

The buffer address in SPFUN 360 must point to a 52-word area in the user's job. The first 26 words are used to hold:

- A response packet length in bytes

- A virtual circuit identifier

- An end packet when the command is complete

The second 26 words are set up by the caller and contain:

- A length word (length of command)

- A virtual circuit identifier (must have octal 1 (001) in high byte)

- A valid TMSCP command (60-byte command buffer)

Except for port initialization, the user program must do all command packet sequencing, error handling, and reinitialization when the bypass operations are complete.

The MU handler supports the following special functions:

Function	Code
Direct TMSCP access	360
Read	370
Write	371
Off-line rewind	372
Rewind to load point	373
Write with extended file gap	374
Backspace one block	375
Forward one block	376
Write EOF	377

For compatibility, SPFUN 360 is supported by the DU handler.

Unit Support, CSR and Vectors

The MU handler supports up to eight units. Each unit requires a separate controller. The first controller CSR address should be 774500. The actual addresses occupied depend on what other devices are on the bus. The default CSR and vector are as follows:

CSR	Vector
774500	260
774504	264
774510	270
774514	274
774520	300
774524	304
774530	310
774534	314

See Section 2.31 of this manual for new SYSGEN support for the MU handler.

1.6.5.5 **NC and NQ (Ethernet Handlers)** - The NC and NQ 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. Each Ethernet handler supports one Ethernet controller. The NC and NQ handlers run only under the XM monitor.

Ethernet communications between two RT-11 systems using the NC or NQ handlers can be demonstrated using the unsupported source file NITEST.MAC. The on-line file UNSUP.TXT describes using NITEST.MAC.

Default CSR and Vectors

There is no default CSR and vector for the NC handler; they are determined at initialization time by the NC handler.

The default CSR and vector for the NQ handler are:

CSR	Vector
174440	120

Device Identification (.DSTATUS)

The NC and NQ Ethernet handlers cause the .DSTATUS programmed request to return a device identification of 61 for "Ethernet class handler".

System Generation Support

See Section 2.31 of this manual for information on system generation support for the NC and NQ Ethernet handlers.

Use of Special Functions (.SPFUN)

The NC and NQ handlers operate the Ethernet controllers using the .SPFUN programmed request with the following function codes.

Function	Code
Allocate Unit	200
Deallocate Unit	200
Reserved	201
Enable protocol type	202
Disable protocol type	202
Enable multicast address	203
Disable multicast address	203
Transmit Ethernet frame	204
Receive Ethernet frame	205

See Chapter 7 of this manual for complete information on using the NC and NQ Ethernet handlers.

1.6.5.6 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.

1.6.5.7 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.

1.6.5.8 XC and XL (Communication Port Handler) - 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 the virtual terminal communication package, VTCOM.

1.6.5.9 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.

The command SET VM BASE=nnnnnn lets you select the memory location that corresponds to logical block 0 of the VM device. For the distributed SJ/FB version, the base is set to 1600 which corresponds to memory location 160000 (the 28K-word boundary). For the distributed XM version, the base is set to 10000 (memory location 1000000), the 18-bit/22-bit addressing boundary. Consequently, when you first boot an XM monitor, only the low 256K bytes of memory will be available for use with the memory management programmed requests, the same amount of memory supported under RT-11 V4. To obtain more memory, remove the VM handler or set its base to a higher value.

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.

Changes have been made to the VM handler for RT-11 V5.2. See Chapter 2 of this manual for details.

1.6.6 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.

1.6.6.1 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.)

NEW FEATURES

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 can define the conditional LIGH$T = 1 to enable support for the idle-loop light pattern. These 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 <u>RT-11 System Generation Guide</u> 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 asks you
in one question 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.

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

● **High-speed ring buffer.** The high-speed ring buffer allows
characters that are received at a very rapid rate to be
processed and transmitted. This option is particularly useful
for systems with intelligent terminals, such as VT1ØØs, and
for Q-BUS systems. This option has no effect for Professional
series computers.

● **Up to 96ØØ baud for DZ11 and DZV11 lines.** The system
generation process lets you set the baud rate for DZ11 and
DZV11 lines to 11Ø, 15Ø, 3ØØ, 12ØØ, 24ØØ, 48ØØ, or 96ØØ. The
baud rates for all DZ11 and DZV11 lines are the same.

● **User command linkage.** UCL support lets you use your own
command parsing program or the distributed UCL.SAV command
parsing program to define your own commands.

● **Fetchable handlers under XM.** You no longer need to load
device handlers when running under the XM monitor. Instead,
RT-11 V5 lets you use the .FETCH programmed request in
background jobs, as has been possible under the SJ and FB
monitors in previous versions of RT-11. This feature is
enabled in the distributed XM monitor, but you can disable
this feature through system generation.

NOTE: Magtape handlers (MM, MT, MS, and MU) are not always
fetchable under the XM monitor. DIGITAL recommends you always
LOAD magtape handlers prior to use.

If you use any device handlers that are not distributed by
DIGITAL, you may need to make some modifications to use the
.FETCH programmed request under XM. See the RT-11
Programmer's Reference Manual for more information on the
.FETCH programmed request.

- **Global .SCCA.** Global .SCCA support lets you inhibit double CTRL/C aborts. When a global .SCCA request is issued, all CTRL/C characters are disabled until another global .SCCA is issued. Inclusion of global .SCCA support adds approximately 100 (decimal) words to a generated monitor.

  Only background jobs can issue global .SCCA requests, and these do not affect foreground or system job operation. Global .SCCA requests issued by foreground and system jobs act as local .SCCA requests.

- **Exclude Professional printer port support from LS handler.** The LS handler has been modified to include support for the Professional series computer printer port. You can choose to exclude the overhead of support for the Professional printer port during system generation.

- **New devices.** You can generate support for the following new devices:

      Ethernet handlers (NC and NQ)
      TK50 magtape (MU)
      TSV05/ TSU05 magtape (MS)
      MSCP disks (DU) - RA80, RC25, RD51, RD52, RX50
      Professional 325, 350, and 380 devices: RD50, 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)

- The idle-loop light pattern has been removed from system generation. If you wish to include idle-loop light pattern support, you must set the conditional LIGH$T to 1 by defining the system conditional during system generation.

- The BASIC keyboard command (conditional BASI$$) has been removed. See Section 2.3.1 of this manual for more details.

- The clock for the Professional 325 and 350 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.

1.6.7 New Error Messages

Error messages have been added for the following new software components:

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

New error messages for RT-11 V5.2 are located in Appendix A of this manual.

1.6.8 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.

1.6.9 New Unsupported Utilities

Several unsupported utilities have been added to the RT-11 distribution kit. Read the file UNSUP.TXT on your RT-11 distribution kit for a description of the unsupported utilities and instructions on how to use them.

1.7 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.

1.7.1 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 medium, if your medium 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)

1.7.2 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.

1.7.3 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 1A of your RT-11 binder set.

1.7.4 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.

1.7.5 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).

1.7.6 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).

1.7.7 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.

1.7.8 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.

1.7.9 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 V5.2.

Sources for all components (supported and unsupported) are provided on the RT-11 source distribution kit, except the following: MACRO, CREF, KED, KEX, and K52.

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 V5.2 software distribution kit:

ALPH00.FNT	PRO GIDIS font table
GIDIS.SAV	Graphics utility for Professional 300 series processors
NC.MAC	Ethernet port handler source code file for system generation for Professional 300 series processors (DECNA controller)
NCX.SYS	Ethernet handler for Professional 300 series processors
NI.MAC	Ethernet class handler source code file for system generation
NITEST.MAC	Ethernet test source file (unsupported)
NQ.MAC	Ethernet port handler source code file for system generation for Q-BUS processors (DEQNA controller)
NQX.SYS	Ethernet handler for Q-BUS processors

2.1.2 New File Type

The following new file type has been added for RT-11 V5.2:

.FNT   Font table file

2.2 MONITORS

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

- **KMON passes unaltered commands** - V5.2 enables KMON to pass 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. KMON stores the unchanged command (all data following the file name and separator) in the chain area beginning at location 512.

  An octal byte count of the contents in the chain area is stored at location 510. The byte count is of chain area contents beginning at location 512, including all null bytes. A value of 1 at location 510 informs a program that it was executed with no characters following the file name.

  If you plan to take data from the chain area, first perform a .GTLIN to remove the converted line. Then get the data from the chain area. .GTLIN returns the CSI form (second group of arguments, the equal (=) sign, first group of arguments). Parentheses in the original command line are expanded, and multiple tabs and spaces are compressed to a single space in the chain area.

  A new optional third argument for .GTLIN supports a program getting the unaltered command line. See Section 2.32.

- **New KMON preprocessor interface (UCF)** - V5.2 lets you write a KMON preprocessor (UCF) that can intercept all command line input after KMON has tried IND command syntax. This 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.

  The following is a description of the KMON/UCF interface:

  CLIFLG indicates which command processors can be run, and is the low byte, located one word below INDDEV (address of INDDEV minus 2). INDDEV (ASCII device name and unit number running IND) is pointed to by RMON fixed offset $INDDV (word offset 426).

  CLITYP is the high byte of the word located below INDDEV. The value stored in CLITYP indicates which command parser the program is operating:

        Bit 0 set - indicates the current program is running
                    as UCF.

Bit 1 set - indicates the current program is running
       from DCL.

Bit 2 set - indicates the current program is running
       as CCL.

Bit 3 set - indicates the current program is running
       as UCL.

KMON collects a command from the command line and checks for IND command syntax. If IND command syntax is not found, KMON checks bit 7 (UCFKMN) of byte CLIFLG.

If bit 7 (UCFKMN) of CLIFLG is set, KMON clears bit 7 and goes through the normal parsing sequence (DCL, CCL, UCL). If bit 7 is clear, KMON checks bit 0 (UCFON). If bit 0 is set, KMON deposits the command line beginning at chain area location 512 and deposits the byte count of command in the word location 510. KMON then clears bit 0 and runs UCF.SAV. UCF.SAV sets bit 0, notifying KMON that UCF is running.

If UCF processes a command line and returns it to KMON, UCF sets bit 7 (UCFKMN) in CLIFLG, and performs a special chain exit. If UCF executes the command line, it does not set bit 7, and does not perform a special chain on exit.

The distributed file UCL.SAV (user command linkage utility) can be copied and renamed to UCF.SAV, creating a functioning UCF preprocessor. See Section 1.6.2.10.

● **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 1.6.2.10 of this manual 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.

Although the USR continues to convert lowercase option characters to uppercase, DIGITAL recommends that all programs test and convert option characters before processing them. The recommended method for testing and converting lowercase characters to uppercase is:

```
        CMPB    @SP,#'A+40
        BLT     x
        CMPB    @SP,#'Z+40
        BGT     x
        BICB    #40,@SP
x:
```

● **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.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

- **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.

- **No SJ monitor for Professional** - RT-11 does not support running the SJ monitor on Professional 300 series systems.

- **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

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

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

● **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.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

	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 128K word program logical address space (PLAS) using virtual overlays and/or virtual .SETTOP. The job PLAS may be up to 4M bytes 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.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

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 /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. (See Section 2.3.3 for a description of these two new options.)

                    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.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

DIFFERENCES

The DIFFERENCES command now accepts wildcards; this allows you to 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, it was necessary to use /OUTPUT:filespec with /SLP to specify a command file; you could not produce a differences listing and a command file simultaneously.

DIRECTORY

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.

The DIRECTORY/BADBLOCKS/VERIFY command is no longer valid. This option combination caused data to be written to the suspected bad block. Therefore, when soft errors occurred, invalid data was written to the block, destroying it.

EXECUTE

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.

The INITIALIZE/BADBLOCKS/VERIFY command is no longer valid. This option combination caused data to be written to the suspected bad block. Therefore, when soft errors occurred, invalid data was written to the block, destroying it.

LINK

You can now link privileged foreground jobs with virtual overlays. Therefore, the /FOREGROUND and /XM options are no longer mutually exclusive.

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

The following new functionality has been added for the REMOVE command under the XM monitor only; it does not apply to the SJ and FB monitors.

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.

When RT-11 is running under the XM monitor, use the following syntax for the REMOVE command:

$$\text{REMOVE} \begin{cases} xx: \\ yy \\ zzzzzz \end{cases}$$

where:

xx: is a device name to be removed from the system tables. Any global region attached to that device is automatically eliminated. If no device name is found, the monitor returns an error message.

yy   is a device name or global region. If a device name, that device name is removed from the system tables. If not a device name, the monitor looks for a global region of that (two character) name, and if found, eliminates it. If device name or global region is not found, the monitor returns an error message.

zzzzzz   is a global region of between one and six characters. If found, that global region is eliminated. If global region is not found, the monitor returns an error message.

Any attempt to eliminate a global region currently attached to a job returns an error message.

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

SHOW   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. For RT-11 V5.2, the SHOW ALL command 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

For RT-11 V5.2, 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. Six types of regions can be indicated.

The region type displayed for regions that have the characteristics of more than one type uses the following order of precedence: private (PVT), permanent (PRM), AGE, hardware (HDW), shared (SHR), and local (LCL). For example, a shared region that also has automatic global elimination enabled, is displayed as type AGE. AGE takes precedence over SHR.

The following table describes the six types of regions:

Symbol	Meaning	Description
AGE	AGE	Automatic global elimination. Created with the RS.AGE status argument set. Any program can attach to and access the region.
HDW	Hardware	Considered part of the hardware configuration. RT-11 runs only if this region is present.
LCL	Local	Created by a program for sole use of that program. When program exits or issues a .ELRG request, the region is eliminated.
PRM	Permanent	Permanently installed when RT-11 is bootstrapped. RT-11 runs only if this region is present.
PVT	Private	Created by a program that has not detached from the region. The creating program has sole use (possession) of the region. Cannot be attached to or accessed by another program.

Symbol	Meaning	Description
SHR	Shared	Created by a program that has detached (and possibly reattached) from it. Can be attached to and accessed by any other program. The region remains after the creating program has exited. Can be eliminated by REMOVE keyboard command or by specifying the RS.EGR status argument when a program issues the .ELRG request.

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

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

2.3.2 New Keyboard Monitor Commands.

The following keyboard monitor commands are new.   Options   for   these commands are listed in Section 2.3.3.

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 disassociate 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 Keyboard Monitor Command Options

This section describes new options for old and   new   keyboard   monitor commands.

BACKUP

/DEVICE

Backs-up or restores an entire volume.

/FILE

Restores a single file from   a   backed-up   volume. Requires that /DEVICE option be used to initially back up volume.   Option must be used with /RESTORE and   /DEVICE;   for   example, BACKUP/RESTORE/DEVICE/FILE.

/RESTORE

Copies a file or volume stored on   several   backup volumes to one volume or file.

/VERIFY

Provides verification of data   transferred   to   or from   a   device.   If input does not match output, BUP prints an error message.

COMPILE

/BUFFERING

With the /DIBOL option, directs   the   compiler   to use single-buffered I/O.

/LOG

With the /DIBOL option, creates   a   log   file   of error messages generated by the compiler.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

/PAGE:n   With the /DIBOL option, specifies the number of
                     lines in a listing page. The default is 66.

/TABLES   With the /DIBOL option, includes symbol and label
                     tables in the output listing.

COPY

/BEFORE[:date]   Copies files created before the specified date.

/DATE[:date]   Copies files created on the specified date.

/INFORMATION   Causes an informational rather than fatal message
                     to print when a file in the command line is not
                     found. If used in an indirect command file, the
                     remainder of the command file is processed.

/MULTIVOLUME   Copies files from an input volume to one or more
                     output volumes using one command.

/PROTECTION   Assigns a protection status to the output file,
                     which prevents deletion of the output file until
                     the protection status is changed.

/NOPROTECTION   Removes protection from the output file so you can
                     delete it. If you use neither /PROTECTION nor
                     /NOPROTECTION in a command line, the output file
                     retains the protection status of the input file.

/RETAIN   With /DEVICE option, preserves the output volume's
                     bad-block replacement table.

/SINCE[:date]   Copies files created on or after the specified
                     date.

DELETE

/BEFORE[:date]   Deletes files created before the specified date.

/DATE[:date]   Deletes files created on the specified date.

/INFORMATION   Causes an informational rather than fatal message
                     to print when a file in the command line is not
                     found. If used in an indirect command file, the
                     remainder of the command file is processed.

/SINCE[:date]   Deletes files created on or after the specified
                     date.

DIBOL

/BUFFERING   Directs the compiler to use single-buffered I/O.

/LOG   Creates a log file of error messages generated by
                     the compiler.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

/PAGE:n	Specifies the number of lines in a listing page. The default is 66.
/TABLES	Includes symbol and label tables in the output listing.

DIFFERENCES

/DEVICE	Compares two entire devices starting with block 0.

DIRECTORY

/BACKUP	Lists the directory of volumes created with the BACKUP command.
/PROTECTION	Includes in the directory listing only those files on the specified volume that are protected against deletion.
/NOPROTECTION	Includes in the directory listing only those files on the specified volume that are not protected against deletion.

EDIT

/KEX	Selects the virtual KED editor, KEX.

EXECUTE

/BUFFERING	With the /DIBOL option, directs the compiler to use single-buffered I/O.
/DUPLICATE	Places duplicate of a library module in each overlay segment that references the module. This option reduces the size of your program's root segment.
/GLOBAL	With the /MAP option, includes a global symbol cross-reference section in the load map.
/LOG	With the /DIBOL option, creates a log file of error messages generated by the compiler.
/PAGE:n	With the /DIBOL option, specifies the number of lines in a listing page. The default is 66.
/TABLES	With the /DIBOL option, includes symbol and label tables in the output listing.

INITIALIZE

/BACKUP	Initializes a volume to be used as an output volume with the BACKUP command.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

LINK

/DUPLICATE	Places duplicate of a library module in each overlay segment that references the module. This option reduces the size of your program's root segment.
/GLOBAL	With the /MAP option, includes a global symbol cross-reference section in the load map.
/LIMIT:n	Used with /XM to limit the amount of memory allocated by a .SETTOP programmed request to nK words (octal).

MOUNT

/WRITE	Write-enables the logical disk you specify.
/NOWRITE	Write-locks the logical disk you specify.

PRINT

/BEFORE[:date]	Prints files created before the specified date.
/DATE[:date]	Prints files created on the specified date.
/INFORMATION	Causes an informational rather than fatal message to print when a file in the command line is not found. If used in an indirect command file, the remainder of the command file is processed.
/SINCE[:date]	Prints files created on or after the specified date.

PROTECT

/BEFORE[:date]	Protects files created before the specified date.
/DATE[:date]	Protects files created on the specified date.
/EXCLUDE	Protects all files on a volume except the ones you specify.
/INFORMATION	Causes an informational rather than fatal message to print when a file in the command line is not found. If used in an indirect command file, the remainder of the command file is processed.
/LOG	Lists on the terminal the names of files protected by the current command.
/NOLOG	Does not list on the terminal the names of files being protected.
/NEWFILES	Protects files created on the current system date.
/QUERY	Requests confirmation before protecting each file.
/SINCE[:date]	Protects files created on or after the specified date.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

/SYSTEM   Lets you protect system (.SYS) files when you use wildcards in the file specification.

/WAIT   Initiates the PROTECT operation, then pauses and waits while you change volumes. For example, you may need to temporarily replace the system volume with a data volume.

RENAME

/BEFORE[:date]   Renames files created before the specified date.

/DATE[:date]   Renames files created on the specified date.

/INFORMATION   Causes an informational rather than fatal message to print when a file in the command line is not found. If used in an indirect command file, the remainder of the command file is processed.

/SINCE[:date]   Renames files created on or after the specified date.

SET

dd CSR=n   Modifies device handler dd to use n as the CSR address for the first controller. The following handlers are valid with this command: DD, DL, DM, DX, DY, LP, LS, NQ, RK, and XL.

dd RETRY=n   Defines the number of times (n) that a device handler tries to recover from an error while the error logger is running. This command is valid for any device the error logger supports.

dd SUCCES   Causes the error logger to log both successful and unsuccessful I/O transfers. This command is valid for any device the error logger supports.

dd NOSUCCES   Causes the error logger to log only unsuccessful I/O transfers. This command is valid for any device the error logger supports.

dd VECTOR=n   Modifies the device handler dd to use n as the vector address for the first controller. The following handlers are valid with this command: DD, DL, DM, DX, DY, LP, LS, NQ, RK, and XL.

DU CSR2=n   Modifies the DU handler to use n as the CSR address for the second controller.

DU CSR3=n   Modifies the DU handler to use n as the CSR address for the third controller.

DU CSR4=n   Modifies the DU handler to use n as the CSR address for the fourth controller.

DU VEC2=n   Modifies the DU handler to use n as the vector for the second controller.

DU VEC3=n   Modifies the DU handler to use n as the vector for the third controller.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

DU VEC4=n   Modifies the DU handler to use n as the vector for
                       the fourth controller.

DUn PART=x   Defines the disk partition on which device unit n
                       resides.

DUn PORT=x   Defines which port to access when device unit n is
                       specified.

DUn UNIT=x   Defines which unit plug number to access when
                       device unit n is specified.

DUn: WRITE   Write-enables MSCP drive unit n.

DUn: NOWRITE   Write-locks MSCP drive unit n.

DW WCHECK   Verifies output to RD50, RD51, and RD52 disks by
                       reading data after writing it to the disk.

DW NOWCHECK   Does not verify output to RD50, RD51, and RD52
                       disks.

DW WRITE   Write-enables RD50/RD51/RD52 drive unit 0.

DW NOWRITE   Write-locks RD50/RD51/RD52 drive unit 0.

DXn WRITE   Write-enables RX01 drive unit n.

DXn NOWRITE   Write-locks RX01 drive unit n.

DYn WRITE   Write-enables RX02 drive unit n.

DYn NOWRITE   Write-locks RX02 drive unit n.

EDIT KEX   Selects the KEX (virtual KED) editor as the
                       default under XM only.

EL LOG   For error logging under the SJ monitor, turns on
                       the error logger if the EL handler is loaded, and
                       begins logging errors in an EL handler internal
                       buffer.

EL NOLOG   For error logging under the SJ monitor, turns off
                       the error logger.

EL PURGE   For error logging under the SJ monitor, discards
                       the contents of the EL handler internal buffer.

EXIT SWAP   Causes any portion of a program that resides in
                       SWAP.SYS to be written back into memory on program
                       termination.

EXIT NOSWAP   Prevents any portion of a program that resides in
                       SWAP.SYS from being written back into memory on
                       program termination.

KMON IND   Causes IND to execute the file specified with the
                       @filespec syntax as an indirect control file.
                       This may be overridden by using the syntax
                       $@filespec which forces execution of the command
                       file by KMON regardless of the current KMON/IND
                       setting.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

KMON NOIND
Causes KMON to execute the file specified with the @filespec syntax as an indirect command file.

LD EMPTY
Dismounts all logical disk assignments.

LDn CLEAN
For logical disk subsetting, verifies and corrects logical disk assignments.

LDn WRITE
For logical disk subsetting, write-enables logical disk unit n.

LDn NOWRITE
For logical disk subsetting, write-locks logical disk unit n.

LP BIT8
Sets the LP handler to pass the eighth bit unaltered, but does not guarantee the eighth bit will arrive at the output device unaltered. The following can alter or strip the eighth bit:

- Some RT-11 utilities in ASCII mode
- Some RT-11 KMON commands, such as PRINT
- Hardware device initialization program options
- Hardware device straps

LP NOBIT8
Sets the LP handler to force the eighth bit to zero, but does not guarantee the eighth bit will arrive at the output device as zero (see SET LP BIT8).

LP ENDPAG=0
Sets the LP handler to not append a form feed (<FF>) character to the end of each file sent to the printer. The default.

LP ENDPAG=n
Sets the LP handler to append n number of form feed (<FF>) characters to the end of each file sent to the printer.

LS BIT8
Sets the LS handler to pass the eighth bit unaltered, but does not guarantee the eighth bit will arrive at the output device unaltered. The following can alter or strip the eighth bit:

- Some RT-11 utilities in ASCII mode
- Some RT-11 KMON commands, such as PRINT
- Hardware device initialization program options
- Hardware device straps

LS NOBIT8
Sets the LP handler to force the eighth bit to zero, but does not guarantee the eighth bit will arrive at the output device as zero (see SET LS BIT8).

LS ENDPAG=0
Sets the LS handler to not append a form feed (<FF>) character to the end of each file sent to the printer. The default.

LS ENDPAG=n
Sets the LS handler to append n number of form-feed (<FF>) characters to the end of each file sent to the printer.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

LS SPEED=n

On a Professional 300 series system only, sets the printer to run at baud rate n, where n can be any of the following:

50	1200
75	1800
110	2000
134	2400
150	3600
200	4800
300	9600
600	19200

You must specify a value for n. If you do not use this option, the printer runs at 4800 baud.

MUx CSR=n

Modifies the specified (x) MU device handler to use n as the CSR address.

MUx VECTOR=n

Modifies the specified (x) MU device handler to use n as the vector address.

NC SHOW

Displays the station physical address for the DECNA controller.

NQ SHOW

Displays the station physical address for the DEQNA controller.

SL ASK

Determines terminal type; if supported, terminal sets SL appropriately.

SL KMON

Sets SL to edit only a monitor command line; that is, input following the monitor prompt (.) and monitor input requests. Use SET SL KMON as an alternative to SET SL ON.

SL LEARN

Locks SL help text on a VT100 or VT102 screen.

SL NOLEARN

Clears SL help text from VT100 or VT102 screen.

SL LET

Enables string substitution using the unsupported program LET.SAV.

SL NOLET

Disables LET.SAV string substitution.

SL OFF

Turns off the single-line editor. This must be the last option in a multiple option SET command; for example, SET SL VT100,LEARN,OFF.

SL ON

Turns on the single-line editor. SL edits all lines typed at the terminal. This must be the last option in a multiple option SET command; for example, SET SL VT100,LEARN,ON. See SET SL KMON.

SL SYSGEN

Causes SL to adapt to the booted monitor's system generation options. You cannot rename SL.SYS to SLX.SYS and issue this command to run under XM.

SL TTYIN

Causes SL to edit all line mode .TTYIN (.CSIxxx and .GTLIN) input, which lets you edit responses to prompts printed by the system utilities.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

SL NOTTYIN | Causes SL to edit only lines that are requested with .CSIGEN, .CSISPC, and .GTLIN programmed requests.

SL VT52 | Causes SL to support the console as a VT52.

SL VT62 | Causes SL to support the console as a VT52.

SL VT100 | Causes SL to support the console as a VT100.

SL VT101 | Causes SL to support the console as a VT101.

SL VT102 | Causes SL to support the console as a VT102. VT102 support is appropriate for all VT1XX family terminals that support INSERT/OVERSTRIKE mode selection (VT102, VT131, VT132, and VT100s with certain options).

SL WIDTH=n | Sets terminal (and editing) width.

SP ENDPAG=0 | Sets the SP handler to not append a form-feed (<FF>) character to the end of each file sent to the printer. The default.

SP ENDPAG=n | Sets the SP handler to append n number of form-feed (<FF>) characters to the end of each file sent to the printer.

SP FLAG=n | Sets the number of flag pages that SPOOL generates whenever it begins printing a file. PRINT/FLAGPAGE:n, when a value is specified for n, overrides this setting. The default for n is 0. The largest value for n is 4.

SP FORM0 | Issues a form feed on the output device each time SPOOL encounters block 0 of a file to be printed. Useful if the output device is part of a multiterminal system, or if the output device handler does not support its own FORM0 option. The default is NOFORM0.

SP NOFORM0 | Turns off FORM0 mode. The default.

SP KILL | Removes all currently spooled output from SPOOL's work file.

SP NEXT | Stops sending output from the current file, discards the remaining spooled output for that file, and begins sending output from the next listing in SPOOL's work file.

SP WAIT | Suspends sending output from SPOOL's work file to the output device, but does not delete anything from the work file. SPOOL continues to accept input with SET SP WAIT in effect.

SP NOWAIT | Resumes sending spooled output suspended by SET SP WAIT.

SP WIDE | Causes SPOOL to generate 132-column flag pages.

SP NOWIDE | Causes SPOOL to generate 80-column flag pages.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

VM BASE=nnnnnn   Lets you select the memory location where block 0 of the virtual device will begin.

XC SPEED=n   Sets the communication port to run at baud rate n, where n can be any of the following:

50	1200
75	1800
110	2000
.134	2400
150	3600
200	4800
300	9600
600	19200

You must specify a value for n. If you do not use this option, the communication port runs at 1200 baud.

SETUP

480INTERLACE   Applies to Professional 380 series processors only. When you set 480INTERLACE, the Professional 380 terminal screen can display high-resolution graphics.

COLOR
MONO   Applies to Professional 350 and 380 processors only. Only set COLOR if you have a color terminal. When you set COLOR, your color terminal supports color functionality with utilities such as GIDIS.

Set MONO mode when your processor does not have a color terminal. The default.

COMPOSE
NOCOMPOSE   Applies to Professional 300 series processors only. When you set COMPOSE, the COMPOSE CHARACTER key on your keyboard functions. COMPOSE mode lets you combine keys to construct national replacement character (NRC) characters. See Section 4.5.1 for explanation of NRC character construction using COMPOSE. See Section 4.5.3 for all valid compose sequences for the various NRC set keyboards.

When you set NOCOMPOSE, the COMPOSE CHARACTER key on your keyboard does not function. The default.

DATA
TYPE   Applies to Professional 300 series processor keyboards only. When you set DATA, the data processing version of the current national replacement character (NRC) set is used. The default.

When you set TYPE, the typewriter version of the current national replacement character (NRC) set is used.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

LANDSCAPE
PORTRAIT

Applies to LN03 laser printers only. When you set LANDSCAPE, the LN03 printer is set to print across the wider dimension of the page. If the paper dimensions are 8-1/2 X 11 inches, LANDSCAPE sets the LN03 to print across the 11-inch dimension.

PORTRAIT (the default), sets the LN03 printer to print across the narrower dimension of the page. If the paper dimensions are 8-1/2 X 11 inches, PORTRAIT sets the LN03 to print across the 8-1/2 inch dimension.

RETAIN

Applies to Professional 300 series processors only. When you set RETAIN, the setup utility keeps the language code previously selected by the SETUP LANG:code command. Setup discards all unused language codes except USA. Use of RETAIN mode requires the unsupported SPLIT utility be on the system device. If the SPLIT.SAV file is not on your system device, place it there prior to issuing the SETUP RETAIN command. Issue the SETUP RETAIN command only on your working system; do not issue it on your distribution or backup media.

RETAIN must be the last SETUP mode listed in a series of SETUP modes in a single SETUP command.

SETCOLOR color [pcolor:value,pcolor:value,pcolor:value]

Applies to Professional 350 and 380 series color terminals only. Lets you mix any or all of the three primary colors (pcolor) to create and store the exact shade you want for any of the eight colors (color) specified in the SETUP [color1] ON [color2] command.

The three primary colors are: BLUE, GREEN, and RED. The eight colors are: BLACK, BLUE, CYAN, GREEN, MAGENTA, RED, WHITE, and YELLOW.

Value is a decimal number that determines the amount of primary color in the created color. A lower number specifies a smaller amount and a higher number a larger amount of that primary color in the mix.

The range of values for primary colors is 0-15, and is the same for both the Professional 350 and 380 processors. However, the actual incremental change in intensity between one primary color and the next differs between processors. The range of values is the same for both processors so either processor will produce approximately the same color from the primary colors.

For either processor, any primary color not specified is assigned the value zero. If all primary colors are assigned zero value, the color is BLACK.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

The Professional 350 has eight primary color values for GREEN and RED, and four values for BLUE. Primary color intensity for GREEN and RED changes with values 0, 2, 4, 6, 8, 10, 12, and 14, and for BLUE changes with values 0, 4, 8, and 12. Refer to the following table:

Value	Blue	Green	Red
0	change	change	change
1	-	-	-
2	-	change	change
3	-	-	-
4	change	change	change
5	-	-	-
6	-	change	change
7	-	-	-
8	change	change	change
9	-	-	-
10	-	change	change
11	-	-	-
12	change	change	change
13	-	-	-
14	-	change	change
15	-	-	-

The Professional 380 implements the full range of values (0-15) in pcolor amounts.

For example

    SETUP SETCOLOR BLUE GREEN:1,BLUE:7

sets BLUE to medium, with a little GREEN and no RED. The command SETUP BLUE then generates a medium blue text that contains a little green.

SETCOLOR color FACTORY

Sets the specified color to the distributed setting.

SHOW

COMMANDS
Sends a list of created UCL commands to the terminal.

/OUTPUT:filespec
Is now supported for SHOW COMMANDS. It sends a list of created UCL commands to the specified file. The option is position-dependent, and must follow COMMANDS (SHOW COMMANDS/OUTPUT:filespec).

/PRINTER
Is now supported for SHOW COMMANDS. It sends a list of created UCL commands to the line printer. The option is position-dependent, and must follow COMMANDS (SHOW COMMANDS/PRINTER).

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

MEMORY | Displays the organization of physical memory: physical addresses of loaded jobs, loaded device handlers, KMON, USR, and the type of region in the extended memory map.

SUBSET | Lists logical disk assignments for physical disks that contain logical disks.

TYPE

/BEFORE[:date] | Types, on the console, files created before the specified date.

/DATE[:date] | Types, on the console, files created on the specified date.

/INFORMATION | Causes an informational rather than fatal message to print when a file in the command line is not found. If used in an indirect command file, the remainder of the file is processed.

/SINCE[:date] | Types, on the console, files created on or after the specified date.

UNPROTECT

/BEFORE[:date] | Removes protection from files created before the specified date.

/DATE[:date] | Removes protection from files created on the specified date.

/EXCLUDE | Removes protection from all files on a volume except the ones you specify.

/INFORMATION | Causes an informational rather than fatal message to print when a file in the command line is not found. If used in an indirect command file, the remainder of the file is processed.

/LOG | Lists on the terminal console the names of the files affected by the current UNPROTECT command.

/NOLOG | Does not list on the terminal console the names of the files affected by the current UNPROTECT command.

/NEWFILES | Removes protection from files created on the current system date.

/QUERY | Requests confirmation before removing protection from each file.

/SINCE[:date] | Removes protection from files created on or after the specified date.

/SYSTEM | Lets you remove protection from system (.SYS) files when you use wildcards in the file specification.

/WAIT   Initiates the UNPROTECT operation, then pauses and
                 waits for you to change volumes. For example, you
                 may need to temporarily replace the system volume
                 with a data volume.

2.4 DEVICE CODES

The following device codes have been assigned.

Name	Code	Device
MU	60	TMSCP magtape class handler
NC/NQ	61	Ethernet class 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 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

        SET DD: SUCCES

        SET DD: NOSUCCES

    See Section 2.3.3 for a description of those commands.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

2.5.2 DL

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

    SET DL: CSR=n

    SET DL: RETRY=n

    SET DL: SUCCES

    SET DL: NOSUCCES

    SET DL: VECTOR=n

    See Section 2.3.3 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 SET commands are valid for the DM handler:

    SET DM: RETRY=n

    SET DM: SUCCES

    SET DM: NOSUCCES

    See Section 2.3.3 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

- For compatibility with the MU handler, DU supports the new SPFUN 360. For DU, SPFUN 360 is functionally equivalent to SPFUN 371 (direct MSCP access).

- 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.

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

        SET DU: CSR=n

        SET DU: VECTOR=n

        SET DU: CSR2=n

        SET DU: VEC2=n

        SET DU: CSR3=n

        SET DU: VEC3=n

        SET DU: CSR4=n

        SET DU: VEC4=n

        SET DU: RETRY=n

        SET DU: SUCCES

        SET DU: NOSUCCES

        SET DUn: WRITE

        SET DUn: NOWRITE

        SET DUn: PART=x

        SET DUn: PORT=x

        SET DUn: UNIT=x

See Section 2.3.3 for more information on those commands.

2.5.5 DW

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

        SET DW: SUCCES

        SET DW: NOSUCCES

        SET DW: WCHECK

        SET DW: NOWCHECK

        SET DW: WRITE

        SET DW: NOWRITE

See Section 2.3.3 for more information on those commands.

2.5.6 DX

The following SET commands are valid for the DX handler:

        SET DX: CSR=n

        SET DX: RETRY=n

        SET DX: SUCCES

        SET DX: NOSUCCES

        SET DX: VECTOR=n

        SET DXn: WRITE

        SET DXn: NOWRITE

See Section 2.3.3 for more information on those commands.

2.5.7 DY

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

            SET DY: CSR=n

            SET DY: RETRY=n

            SET DY: SUCCES

            SET DY: NOSUCCES

            SET DY: VECTOR=n

            SET DYn: WRITE

            SET DYn: NOWRITE

      See Section 2.3.3 for more information on those commands.

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

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

2.5.8 DZ

The following SET commands have been removed from the DZ handler because RX50 diskettes provide hardware write protection:

    SET DZ: WRITE

    SET DZ: NOWRITE

The following SET commands are valid for the DZ handler:

    SET DZ: RETRY=n

    SET DZ: SUCCES

    SET DZ: NOSUCCES

See Section 2.3.3 for more information on those commands.

2.5.9 LD

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

      SET LD EMPTY

  SET LD EMPTY dismounts all mounted logical disks at one time.

2.5.10 LP

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

    SET LP BIT8

    SET LP NOBIT8

    SET LP ENDPAG=0

    SET LP ENDPAG=n

See Section 2.3.3 for more information on those commands

2.5.11 LS

- The default setting for the LS handler has been changed from NOCTRL to CTRL.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

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

    SET LS BIT8

    SET LS NOBIT8

    SET LS ENDPAG=0

    SET LS ENDPAG=n

    SET LS SPEED=n

  See Section 2.3.3 for more information on those commands.

2.5.12 MU

The following SET commands are valid for the new MU handler:

    SET MUx: CSR=n

    SET MUx: VECTOR=n

See Section 2.3.3 for more information on those commands.

2.5.13 PI

- 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 chapter 5 of this manual for a description of the interface between PI and the PRO/GIDIS utility.

- The RT-11 V5.2 secondary DA (device attributes) reply for the Professional 300 series is now <ESC>[>7;503c for the FB monitor, and <ESC>[>8;503c for the XM monitor.

2.5.14 RK

The following SET commands are now valid for the RK handler:

    SET RK: CSR=n

    SET RK: RETRY=n

    SET RK: SUCCES

    SET RK: NOSUCCESS

    SET RK: VECTOR=n

See Section 2.3.3 for more information on those commands.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

2.5.15 SP

The following SET commands are now valid for the SP handler:

    SET SP ENDPAG=0

    SET SP ENDPAG=n

See Section 2.3.3 for more information on those commands.

2.5.16 XC

- 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 Section 2.3.3 for more information on that command.

2.5.17 XL

- 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

        SET XL VECTOR=n

    See Section 2.3.3 for more information on those commands.

2.5.18 VM

The following changes have been made to the VM handler.

- An RT-11 V5.2 XM monitor can now be booted from the extended memory VM device.

    When booting an XM monitor with VM as the system device, set the base address of the VMX handler using the SET VM BASE=nnnnnn keyboard command. Use the SHOW MEMORY command display, under the header "Address", to find the VMX handler base address value (nnnnnn). Enter that value in the SET VM BASE=nnnnnn command.

2.6 AUTOMATIC INSTALLATION

The following changes and additions have been made to the automatic installation 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 file structure, the procedure warns you and gives you the opportunity to continue or stop the formatting.

  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 formatting.

  If .STRUCTURE finds an unknown file structure, the disk is formatted.

  See Section 2.15 for a description of the new .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 1.6.1 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.

- 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.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

- 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 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.

2.10 DUP

The following changes have been made to the device utility program (DUP).

- The /H option is now invalid when combined with the /K or /B option. That option combination caused data to be written to the suspected bad block. Therefore, when soft errors occurred, invalid data was written to the block, destroying it.

  To verify whether a bad block is caused by a hard or soft error, you can perform two bad-block scans and compare the results to see if any blocks reported as bad were able to recover.

- 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 in order 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 Sections 2.20 and 2.25 in 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.

2.11 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/RD51/RD52 disk) and DZ (RX50 diskette) handlers for the Professional 325 and 350.

- 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.

2.12 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.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

2.13 FORMAT

The following changes have been made to the volume formatting (FORMAT) program.

- Formatting RD50, 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 of this manual for a description of formatting an unformatted hard disk on Professional 300 series processors.

- 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. This 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.

2.14 HELP

The files HELP.TXT and HELP.EXE, which together make up the program HELP.SAV, are no longer provided on the distribution kit. Therefore, if you want to change your HELP text, as described in Section 2.7.14 of the _RT-11 Installation Guide_, you must first recreate HELP.TXT and HELP.EXE from HELP.SAV using the unsupported utility SPLIT.

To recreate those files, type this command:

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

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

2.15 IND

The following additions and changes have been made to the indirect control file (IND) processor.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

2.15.1 IND Partially Resident in Extended Memory (XM Only)

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.

When IND performs a normal exit, that portion of IND resident in extended memory is removed, and the extended memory becomes available to other programs. When you exit IND using a double CTRL/C, that portion of IND remains in extended memory. Execute the following command in response to the monitor prompt (.) to remove IND from extended memory:

    REMOVE IND

Attempting to remove IND while running IND, or when IND is being run by another program, returns an error message.

2.15.1.1 Suppressing Dynamic Allocation of a Region for IND

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 the following customization patch.

In the customization, ..ALRG is the value for that symbol from the file CUSTOM.TXT on your distribution kit. Enter the value 0 (zero) for symbol n to suppress dynamic allocation, or the value 1 for symbol n to dynamically allocate a region.

```
.R SIPP<RET>
*IND.SAV<RET>
Segment?  0<RET>
Base?     0<RET>
Offset?   ..ALRG<RET>

        Base      Offset      Old       New?
        000000    ..ALRG      000001    n<RET>
        000000    ..ALRG+2    ??????    <CTRL/Y><RET>
*<CTRL/C>
.
```

2.15.2 .STRUCTURE

The .STRUCTURE directive has been added to IND to determine the file structure of a specified file-structured device.

The syntax of the .STRUCTURE directive is

>   .STRUCTURE numsym dev

where:

> numsym represents an octal symbol in which the file structure type is stored.   The following is a list of values for file structure types represented by the symbol returned in numsym:

Octal	Values
000	Unknown
001-007	Reserved
010-017	PDP-8 (12-bit systems)
010	WPS-8
011	OS-8
012	COS
020-027	RT-11 systems
020	RT-11 V5
030-037	RSTS systems
030	RSTS V8
040	Reserved
041-047	FILES-11 systems
041	FILES-11 Level 1
042	FILES-11 Level 2
050-057	UNIX systems
050	UNIX
060-067	DSM systems
060	DSM V3
070-077	Reserved
100-107	CP/M systems
100	CP/M
110-117	UCSD p-systems
110	UCSD p-system
120-127	MS-DOS systems
120	MS-DOS
130-177	Reserved
200-377	CSS and customer file structures

dev   represents the device for which the file structure is to be determined. The device can be specified by a string symbol in single quotes (when substitution mode is enabled) or a character string. A colon following the device mnemonic is ignored.

The volume in the device must conform to DIGITAL's boot standard for IND to determine the file structure of the device. If a volume does not conform to DIGITAL's boot standard, .STRUCTURE returns the value 000.

The following example returns #20 in symbol MULE if DW is an RT-11 V5 disk:

```
.STRUCTURE MULE DW:
;'MULE'
.EXIT
```

2.15.3 .TESTDEVICE

The .TESTDEVICE directive has been modified to include ATT or NAT (attached or not attached) in field nine of special symbol <EXSTRI>.

The ATT attribute is returned in <EXSTRI> if the requested device is attached to a specific job. A magtape, cassette, or any other sequential device that has a currently open file returns the ATT attribute. An attached device assumes that LOD and ONL are also true, and returns them in <EXSTRI> along with ATT.

The NAT attribute is returned in <EXSTRI> if the requested device is not attached to any specific job.

Issuing a .TESTDEVICE directive to a variable size device that is attached, such as a logical disk, returns the smallest valid device size.

Refer to the following example control file:

```
LOAD DZ:=F
ASSIGN DZ TOP
.TESTDEVICE TOP
;'<EXSTRI>'                    returns:  DZ0,800.,0,0,0,LOD,ONL,NMT,ATT,
.TESTDEVICE DZ
;'<EXSTRI>'                    returns:  DZ0,800.,0,0,0,LOD,ONL,NMT,ATT,
.TESTDEVICE DZ1
;'<EXSTRI>'                    returns:  DZ1,800.,0,0,0,LOD,OFL,NMT,NAT,
LOAD XC:=VTCOM
.TESTDEVICE XC
;'<EXSTRI>'                    returns:  XC0,0.,0,0,0,LOD,ONL,NMT,ATT,
.STOP
```

2.15.4 .ONERR

The .ONERR directive has been modified to include processing integer under/over flow conditions.

2.15.5 .SETL

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.

In the following example, the control file contains this line:

    .SETL MONITR SJ .OR FB .OR XM

If any of the three logical symbols (SJ,FB, or XM) is set to true, the logical symbol MONITR is set to true. If none of the three is set to true, MONITR is set to false.

2.15.6 .ENABLE/.DISABLE

The following changes have been made to the .ENABLE and .DISABLE directives. These changes also apply to the .IFENABLED and .IFDISABLED directives.

2.15.6.1 .ENABLE/.DISABLE ABORT - This new operating mode lets you enable or disable double CTRL/C aborts. When .DISABLE ABORT is in effect, CTRL/C characters are ignored until the currently executing control file exits. When .ENABLE ABORT is in effect, CTRL/C characters are recognized and processed. .DISABLE ABORT is valid only if you have included global .SCCA support in your system through system generation.

.DISABLE ABORT also disables CTRL/Z aborts.

See Chapter 5 of the RT-11 System User's Guide for more information on using .ENABLE and .DISABLE ABORT.

2.15.6.2 .ENABLE/.DISABLE CONTROL-Z - This new operating mode lets you enable or disable CTRL/Z aborts. Typing CTRL/Z in response to an .ASK, .ASKN, or .ASKS prompt causes the control file to abort. When .DISABLE CONTROL-Z is in effect, CTRL/Z characters are ignored until the currently executing control file exits. When .ENABLE CONTROL-Z is in effect, CTRL/Z characters are recognized and processed.

See Chapter 5 of the RT-11 System User's Guide for more information on using .ENABLE and .DISABLE CONTROL-Z.

2.15.6.3 .ENABLE/.DISABLE TYPEAHEAD - This new operating mode causes IND to accept or ignore type-ahead. When .ENABLE TYPEAHEAD is in effect, IND accepts and stores characters you type to answer an .ASK, .ASKS, or .ASKN prompt even before the prompt is displayed. When .DISABLE TYPEAHEAD is in effect, IND discards all characters that have been stored before processing .ASK, .ASKN, and .ASKS directives. If you have answered a prompt prematurely, your response is discarded.

See Chapter 5 of the RT-11 System User's Guide for more information on using .ENABLE and .DISABLE TYPEAHEAD.

2.15.6.4 .ENABLE/.DISABLE TIMEOUT - In RT-11 V5.1, 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 more information on using .ENABLE and .DISABLE TIMEOUT.

2.16 KED

The following changes have been made to the keypad editor (KED).

- <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 of those 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.

2.17 LIBR

The following changes have been made to the RT-11 librarian (LIBR) program.

- For RT-11 V5.2, a new global symbol ($OVTAB) must be excluded when merging the FORLIB.OBJ modules with SYSLIB.OBJ. Failure to exclude $OVTAB when merging the libraries causes multiple values for the same global symbol, and FORTRAN programs do not link properly. Issue the following command in response to the monitor prompt (.) to merge FORLIB.OBJ with SYSLIB.OBJ:

  ```
  .LIBRARY/INSERT/REMOVE SYSLIB FORLIB<RET>
  Global?  $ERRS<RET>
  Global?  $ERRTB<RET>
  Global?  $OVRH<RET>
  Global?  $OVTAB<RET>
  Global?  <RET>
  ```

- 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.

2.18 LINK

The following changes have been made to the RT-11 linker (LINK) program.

- 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 /R and /V options are no longer mutually exclusive; privileged foreground jobs can be linked with virtual overlays. The root and low memory overlays are now located just below the lowest window created for virtual overlays.

- 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.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

2.19 MACRO-11

The following changes have been made to MACRO-11.

2.19.1 Multinational Character Set Support

RT-11 V5.2 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.

2.19.2 Increasing Default Size of MACRO-11 Work file

The default 128 (decimal) block MACRO-11 work file provides enough storage for assembling most programs. A program that uses a great many symbols or macros can cause the default work file to overflow and return the error message ?MACRO-F-Storage limit exceeded. The following customization increases the work file size to a maximum of 256 (decimal) blocks.

In the customization, nnn represents the size of the work file, and can be a value between 128 and 256 (decimal). Substitute for the symbol WRKSIZ the value of that symbol given in the file CUSTOM.TXT on your distribution kit.

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

Segment	Base	Offset	Old	New?
000000	000000	WRKSIZ	000200	nnn<RET>
000000	000000	WRKSIZ+2	??????	<CTRL/Y><RET>

```
*<CTRL/C>
.
```


2.20 MDUP

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.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

2.21 PAT

When PAT finishes executing a command, control returns to the CSI (indicated by the asterisk prompt, *) rather than to the keyboard monitor.

2.22 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.

2.23 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.

2.24 RESORC

The following changes have been made to the resource (RESORC) program.

- RESORC for RT-11 V5.2 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. Six types of regions can be indicated.

  The region type displayed for regions that have the characteristics of more than one type uses the following order of precedence: private (PVT), permanent (PRM), AGE, hardware (HDW), shared (SHR), and local (LCL). For example, a shared region that also has automatic global elimination enabled, is displayed as type AGE. AGE takes precedence over SHR.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

The following table describes the six types of regions:

Symbol	Meaning	Description
AGE	AGE	Automatic global elimination. Created with the RS.AGE status argument set. Any program can attach to and access the region.
HDW	Hardware	Considered part of the hardware configuration. RT-11 runs only if this region is present.
LCL	Local	Created by a program for sole use of that program. When program exits or issues a .ELRG request, the region is eliminated.
PRM	Permanent	Permanently installed when RT-11 is bootstrapped. RT-11 runs only if this region is present.
PVT	Private	Created by a program that has not detached from the region. The creating program has sole use (possession) of the region. Cannot be attached to or accessed by another program.
SHR	Shared	Created by a program that has detached (and possibly reattached) from it. Can be attached to and accessed by any other program. The region remains after the creating program has exited. Can be eliminated by REMOVE keyboard command or by specifying the RS.EGR status argument when a program issues the .ELRG request.

● The /O option now shows if SYSGEN special feature FPU (floating point unit) support has been selected.

● RESORC displays the following new processor types:

    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 two-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.

2.25 SETUP

The following changes have been made to the hardware characteristics (SETUP) program. See Section 2.3.3 for a description of SETUP modes and controls new to RT-11 V5.2.

- 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.

  The KB destination sets the keyboard only to the specified national replacement character (NRC) set. The console screen font is unaffected and remains at current setting. The current setting of the DATA or TYPE mode remains in effect. Do not specify another destination when using KB destination.

  The following example sets the keyboard only to the German NRC set:

      SETUP KB LANG:D

  The VIDEO destination sets the console screen font only to the specified national replacement character (NRC) set. The keyboard is unaffected and remains at current setting. Do not specify another destination when using VIDEO.

The following example sets the console screen font to the German NRC set:

    SETUP VIDEO LANG:D

KB and VIDEO can be used to set the keyboard of a terminal to one NRC language, and the console screen font to another NRC language. Both destinations cannot be specified within the same SETUP command.

If KB or VIDEO is not specified, the SETUP LANG:code command sets the terminal (keyboard and screen) to the specified NRC set. See Chapter 7 of the RT-11 System User's Guide for a complete list of valid NRC languages.

● 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. Section 4.5.2 of this manual describes the use of Dead Diacritical keys and lists valid Dead Diacritical keys for each NRC language 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. Section 4.5.1 of this manual describes the use of the COMPOSE CHARACTER key.

Valid compose sequences are determined by the particular NRC set keyboard being used. Use the SETUP LANG:code command to select the appropriate keyboard (see the RT-11 System User's Guide for a complete description). A complete list of compose sequences for the various NRC keyboards is located in Section 4.5.3 of this manual.

● 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.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

2.26 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.

2.27 SL

Two new functions have been added to the single-line editor (SL).

2.27.1 Get Older

When you press the PF1 (GOLD) key, then the up-arrow (^) key, SL reproduces the line before the last line terminated with a carriage return. This extends the functionality of the up-arrow (^) key by letting you recall an older line for editing.

In the following example, pressing <GOLD><^> recalls the next-to-last command line for editing:

    .RENAME FILE1.MAC FILE1.BAK<RET>

    .<RET>

    .ASSIGN DL0: LOG:<RET>

    .<GOLD><^>RENAME FILE1.MAC FILE1.BAK

2.27.2 Save and Get Saved

SL now lets you save and recall a command. When you press the PF1 (GOLD) key, then the down-arrow (\/) key, SL saves any command that has been typed at the keyboard. To recall the command, press the down-arrow (\/) key. SL displays the recalled command on your terminal. Press RETURN to execute the recalled command.

The following example illustrates the use of that SL function.

    Type a keyboard command; for example:

    .DIRECTORY DZ1:

Press PF1 (GOLD), then down-arrow (\/). The next time you want
to enter the command, press the (\/) <RET>.

.DIRECTORY DZ1:

SL displays the command at the monitor prompt. SL stores the
saved command as long as the SL handler is loaded, or until you
turn the processor off.

2.28 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.

2.29 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.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

2.30 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.

2.31 SYSGEN

Changes have been made in SYSGEN.COM.

2.31.1 New Device Table

The following is the new available-device table:

DX	RX01 Single-Density Diskette	DY	RX02 Double-Density Diskette
DD	TU58 DECtape II	PD	PDT-11 Intelligent Terminal
RF	RF11 Disk	RK	RK05 Cartridge Disk
DL	RL01/RL02 Cartridge Disk	DP	RP11 Disk Pack
DM	RK06/RK07 Cartridge Disk	MT	TM11 (UNIBUS) Magtape
MS	TS11,TU80/TSV05/TSU05 Tape	MM	TJU16 (MASSBUS) Magtape
CR	Card Reader	NL	Null Handler
LP	Line Printer	DU	MSCP Disk Class Handler
DZ	RX50 PRO Series	DW	RD50/RD51/RD52 PRO Series
SP	Transparent Spooling Handler	XL	VTCOM communications Handler
XC	VTCOM Handler PRO Series	NC	Ethernet Handler PRO Series
NQ	Ethernet Handler for Q-BUS	MU	TMSCP Magtape Class Handler*
LS	Serial Line Printer	DT	DECtape
DS	RJ03/4 Fixed-head Disk	CT	TA11 Cassette
PC	PC11 Paper Tape Reader/Punch	PR	PR11 Paper Tape Reader

* Not supported for RT-11 V5.2 (do not specify support for the MU handler when performing a SYSGEN for RT-11 V5.2)

2-52

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

2.31.2 New SYSGEN.COM Dialog Questions

The following dialog questions have been added to SYSGEN.COM to support the new NQ Ethernet handler.

153. What is the CSR address for NQX (174440) ?

 Enter the CSR address (octal) for NQX.

154. What is the vector address for NQX (120) ?

 Enter the vector address (octal) for NQX.

2.31.3 New System Conditionals

The following new system conditionals have been added to support the new MU and NQ handlers:

```
MU$FSM    = 1         ;TMSCP magtape file-structured support
MU$UN     = n         ;Number of TMSCP magtape units
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
MU$CS4    = nnnnnn    ;Status register of fifth TMSCP magtape unit
MU$VC4    = nnn       ;Vector of fifth TMSCP magtape unit
MU$CS5    = nnnnnn    ;Status register of sixth TMSCP magtape unit
MU$VC5    = nnn       ;Vector of sixth TMSCP magtape unit
MU$CS6    = nnnnnn    ;Status register of seventh TMSCP magtape unit
MU$VC6    = nnn       ;Vector of seventh TMSCP magtape unit
MU$CS7    = nnnnnn    ;Status register of eighth TMSCP magtape unit
MU$VC7    = nnn       ;Vector of eighth TMSCP magtape unit
NQ$CSR    = nnnnnn    ;Status register of Q-BUS Ethernet controller
NQ$VEC    = nnn       ;Vector of Q-BUS Ethernet controller
```


2.32 SYSTEM SUBROUTINE LIBRARY (SYSLIB)

The following changes have been made to the system subroutine library (SYSLIB).

2.32.1 Merging FORLIB.OBJ with SYSLIB.OBJ

For RT-11 V5.2, a new global symbol ($OVTAB) must be excluded when merging the FORLIB.OBJ modules with SYSLIB.OBJ. Failure to exclude $OVTAB when merging the libraries causes multiple values for the same global symbol, and FORTRAN programs do not link properly. Issue the following command in response to the monitor prompt (.) to merge FORLIB.OBJ with SYSLIB.OBJ:

```
.LIBRARY/INSERT/REMOVE SYSLIB FORLIB<RET>
Global?  $ERRS<RET>
Global?  $ERRTB<RET>
Global?  $OVRH<RET>
Global?  $OVTAB<RET>
Global?  <RET>
```

2.32.2 Changes to SYSLIB System Subroutines

The following routines have been changed in SYSLIB.

● 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'. Only the first character of the string need be specified, and the case of the character is unimportant.

Form: CALL GTLIN (result[,prompt$\left\{\begin{array}{c},\text{'term'}\\ ,\text{'plain'}\end{array}\right\}$)

Specify the 'term' argument to take input only from the console terminal, even if the program is running under the control of an indirect command file. Use the 'term' argument when direct response from the program user is required.

Specify the 'plain' argument to take unaltered input from the chain area, and pass the input to the array specified by the GTLIN result argument. The 'plain' argument is helpful when a program requires command input that is not a file specification, such as a SETUP command.

GTLIN with the 'plain' argument works in the following way:

GTLIN checks the word in location 510 in the chain area for a byte count higher than 1.

If location 510 does not contain a byte count higher than 1, GTLIN functions as though the 'plain' argument was not specified.

If location 510 contains a word count higher than 1, GTLIN copies the ASCIZ string beginning at location 512 into the result argument as specified in the GTLIN call. GTLIN then clears location 510.

GTLIN then takes the input (that has been converted by KMON) from the KMON buffer, thereby purging the buffer. Finally, GTLIN places the input from the KMON buffer into the chain area beginning in location 512.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

At the completion of the GTLIN call:

- The program has the unaltered input.

- Location 510 is clear.

- The KMON buffer is clear.

GTLIN can be called from MACRO-11 programs if the standard FORTRAN calling convention is followed. All register contents are destroyed across the call. GTLIN has no dependencies on FORTRAN code or routines.

To avoid possible problems, DIGITAL recommends that the GTLIN subroutine with the 'plain' argument not be used in a program that uses the .CSIGEN and .CSISPC requests or the GTLIN subroutine without the 'plain' argument.

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

- ICSTAT - This routine can now be used under the SJ monitor.

- 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.

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

- ISPFN subroutine calls - The ISPFN subroutine calls (ISPFN/ISPFNC/ISPFNP/ISPFNW) have had support added for the MU, DU, and PI handlers, for the following special functions:

Function	MU	DU	PI
Direct TMSCP access	360		
Direct MSCP access		360,371	
Read	370		370
Write	371		371
Off-line rewind	372		
Rewind to load point	373		
Write with extended file gap	374		
Backspace one block	375		
Forward one block	376		
Write EOF	377		

See Section 1.6.5.4 of this manual for more information on the MU handler.

The ISPFN subroutine calls (ISPFN/ISPFNC/ISPFNF/ISPFNW) have had support added for the NC and NQ Ethernet handlers, for the following special functions:

Function	NC	NQ
Allocate/deallocate unit	200	200
Enable/disable protocol type	202	202
Enable/disable multicast address	203	203
Transmit Ethernet frame	204	204
Receive Ethernet frame	205	205

The ISPFN subroutine calls (ISPFN/ISPFNC/ISPFNF/ISPFNW) have had support added for the DW and DZ handlers, for the following special functions:

Function	DW	DZ
Read absolute sector	377	377
Write absolute sector	376	376
Return volume size	373	

2.33 SYSTEM MACRO LIBRARY (SYSMAC)

The following programmed requests have been changed:

- .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

- .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.

  See the RT-11 Programmer's Reference Manual for more information on .DRDEF and .DSTATUS.

- .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.

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

- .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.

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

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

- .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.

  See the RT-11 Programmer's Reference Manual for more information on global .SCCA support.

- .SPFUN - has had support added for the MU, DU, and PI handlers, for the following special functions:

Function	MU	DU	PI
Direct TMSCP access	360		
Direct MSCP access		360,371	
Read	370		370
Write	371		371
Off-line rewind	372		
Rewind to load point	373		
Write with extended file gap	374		
Backspace one block	375		
Forward one block	376		
Write EOF	377		

  See Section 1.6.5.4 of this manual for more information on the MU handler.

  .SPFUN has had support added for the NC and NQ Ethernet handlers, for the following special functions:

Function	NC	NQ
Allocate/deallocate unit	200	200
Enable/disable protocol type	202	202
Enable/disable multicast address	203	203
Transmit Ethernet frame	204	204
Receive Ethernet frame	205	205

  .SPFUN has had support added for the DW and DZ handlers, for the following special functions:

Function	DW	DZ
Read absolute sector	377	377
Write absolute sector	376	376
Return volume size	373	

  See the RT-11 Programmer's Reference Manual for more information on these special functions.

- .TWAIT - can now be used under the SJ monitor.

2.34 UCL

The following changes have been made to the user command linkage (UCL) utility.

2.34.1 UCL.DAT Data Structure

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. Redefine the commands when you run RT-11 V5.2.

The default data file on disk is 9 blocks long, and can contain up to 31 (decimal) user-defined commands. The complete user-defined command line, including spaces and any control characters, cannot exceed 82 (decimal) characters. (Note for example that pressing the RETURN key generates a <CR><LF> pair, counting as two characters.) You can change the size of the data file using the customization patch described in Section 2.34.3. When first created, the data file is blank-filled.

The first block of the default data file contains the command table. The default command table contains 32 16-byte records. The first record contains an exclamation point (!) character followed by an ASCII representation of the maximum number of available UCL command entries. For the default data file, that number is 37 (octal).

Each command table entry is 16 bytes long, and can contain up to 16 characters and numbers including an optional asterisk (*). Use the asterisk in the command to indicate the point of unambiguous length. Any unused bytes are blank-filled.

The following eight blocks of the default data file contain the command entry definitions. There are four command entry definitions per block. Each entry has a fixed length of 128 characters with any unused space blank-filled.

When you delete a command, UCL blank-fills the command table and the command entry definitions.

2.34.2 New UCL Commands

The CSI (command string interpreter) interface is not supported for UCL; do not issue the R UCL command.

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.

SHOW COMMANDS/PRINTER sends a list of all UCL commands to the line printer.

SHOW COMMANDS/OUTPUT:filespec creates a file listing current commands and definitions under that filespec. Use that option to store current commands and definitions before you apply the customization patch to change the size of the UCL data file. After the customization patch is applied, the file created by /OUTPUT:filespec can be executed as a command file to redefine the commands into the new data file.

2.34.3 Changing the Size of UCL.DAT

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. Use the following procedure.

1. Execute the following command to create a command (.COM) file that lists current commands and definitions under that file specification (filespec). The file specification can have any name you choose, for example NEWUCL.COM.

   Execute the following command whether you have defined commands or not, as the created file is used to build the new command table file.

   .SHOW COMMANDS/OUTPUT:filespec

2. Perform the following customization patch. In the patch, substitute for the the symbol ..CMDS the value for that symbol found in the on-line file CUSTOM.TXT. Replace the symbol nnnnn with an octal value for the maximum number of defined (UCL) commands you want.

   ```
   .R SIPP<RET>
   *UCL.SAV/A<RET>
   Segment? <RET>
   Base?    ..CMDS<RET>
   Offset? 0<RET>

           Base      Offset      Old      New?
           ..CMDS    0            37      nnnnn<RET>
           ..CMDS    xxxxxx      xxxxxx   <CTRL/Y><RET>
   *<CTRL/C>
   .
   ```


3. Execute the following command to delete the contents of the old UCL.DAT, where SY: is your system device.

   .DELETE SY:UCL.DAT

4. If you wish to change the name of the UCL data file, execute the following customization patch. With this patch, you can also change the default device on which the UCL data file resides.

   The file specification for the UCL data file is stored in four RAD50 words: The first word is the device name without the colon (:). You must leave a space if you use a two-letter device name. The second word is the first three letters of the file name. The third word is the second three letters of the file name (if required). The fourth word is the file type. Do not include the period (.) between the file name and file type.

In the patch, ..UCLD is a symbol value for the first RAD50 word. Substitute for ..UCLD the value for that word found in the on-line file CUSTOM.TXT. Substitute for nnn each part of the file specification that you changed when you created a new UCL data file.

```
.R SIPP<RET>
*UCL.SAV/A<RET>
Base?  0<RET>
Offset? ..UCLD<RET>
```

Base	Offset	Old	New?
0000	..UCLD	xxxxxx	;R<RET>
0000	..UCLD	xxxxxx	;Rnnn<RET>
0000	..UCLD+2	xxxxxx	;Rnnn<RET>
0000	..UCLD+4	xxxxxx	;Rnnn<RET>
0000	..UCLD+6	xxxxxx	;Rnnn<RET>
0000	..UCLD+8	xxxxxx	<CTRL/Y><RET>

```
*<CTRL/C>
.
```

5. Create the new UCL data file by executing the following command. In the command, filespec is the file specification you used with the SHOW COMMANDS/OUTPUT:filespec command.

   ```
   .@filespec
   ```

   Any user-defined commands in the old UCL data file are transferred to the new UCL data file, and the new UCL data file can hold the maximum number of commands you specified in the customization patch.

2.34.4 Optimizing UCL

You can optimize the UCL utility by combining the save image program (.SAV) file and the data (.DAT) file. The optimization increases the speed of UCL command processing. If you intend to increase the size of the UCL data file, perform that customization patch before optimizing UCL.

To optimize UCL, execute the following keyboard command in response to the monitor prompt (.):

```
.COPY UCL.(SAV+DAT) UCL.SAV
```

To increase the size of the UCL data file after optimizing UCL:

1. Create a UCL data file by issuing the SHOW COMMANDS/OUTPUT:filespec keyboard command.

2. Delete the optimized UCL.SAV file.

3. Copy the UCL.SAV file from your software distribution kit to your working system volume.

4. Perform the customization patch to increase the size of the UCL data file.

5. Execute the keyboard command to optimize UCL.

2.34.5 Removal of Informational Messages and Prompts

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 (.).

Use the new keyboard command SHOW COMMANDS to verify that a UCL command (in the form you want) is located in the UCL data file. The display produced by that command also verifies that the UCL operation to define, redefine, or delete a UCL command produced the required result.

UCL error messages are listed in Appendix A of this manual.

2.34.6 Creating UCF.SAV from UCL.SAV

The keyboard monitor (KMON) checks for valid UCL command syntax after first checking for valid IND, DCL, and CCL command syntax. To force the keyboard monitor to check for UCL syntax after IND and before DCL, copy or rename UCL.SAV to UCF.SAV. That greatly increases the KMON processing speed of UCL commands.

See Section 1.6.2.10 of this manual for information on copying or renaming UCL.SAV to UCF.SAV.

2.35 VTCOM

The following changes have been made to the virtual terminal communication (VTCOM) package.

- 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 SELECT command provides the interface between you and the Mini-Exchange. When you issue the SELECT command, VTCOM prompts by requesting a Mini-Exchange port. Valid responses are ports 1 through 8, and the letter M or R, indicating that port 8 is connected to a modem. Respond with a port number or a letter.

  After you enter the requested port number, or the letter M or R, VTCOM responds with the status of that port. The four possible status responses are:

      A for accepted
      B for rejected
      C for no device
      Z for busy

  A Z status response will queue you at the requested port. When the requested port becomes available, the connection is made, and you are notified with an A status response.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

If you do not receive a status response within 3 seconds after making a port connection request, check for one of the following problems:

The selected port is not connected to any device.

The selected port is the originating port.

The requested device is malfunctioning or powered off. Check the requested device.

The cable on the requested device is inoperative. Check or replace the cable.

The requested Mini-Exchange port is malfunctioning. Perform diagnostic tests found in the Mini-Exchange documentation.

The following are valid baud rates for establishing a connection through the Mini-Exchange. Set the baud rate using the command SET XC SPEED=n where n is one of the following:

```
300
600
1200
2400
4800
9600
19200
```

Other baud rates are valid once the connection to a device has been established.

The following is an example of an accepted VTCOM request for port 7:

```
TT::VTCOM> SELECT <RET>
TT::VTCOM> PORT? 7 <RET>
A
```

The following is an example of an accepted VTCOM request for port 8 as the modem port:

```
TT::VTCOM> SELECT <RET>
TT::VTCOM> PORT?  M <RET>
A
```

- The shortest valid abbreviation for the SELECT command is SEL, and for the SEND command is SEN.

- VTCOM includes the following new command:

  HANGUP   Drops DTR, holds it low for two seconds (breaking the modem connection) and then raises it. You can then dial another number or issue the DIAL command, to connect with another host computer. HANGUP has the same effect as setting the DATA/TALK button on a modem to TALK for two seconds, then restoring it to DATA.

- Enhancements made to the file transfer code in VTCOM can increase the speed of a transfer through more efficient use of the serial line.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

● The VTCOM SEND command sends ASCII files at two speeds: SLOW or FAST. Use SLOW if the host terminal service does not support XON/XOFF and FAST if it does support XON/XOFF.

The following customization patch lets you set the default speed at which VTCOM sends ASCII files. In the customization, ..FAST is the value of that symbol given in the file CUSTOM.TXT on your distribution kit. Set the value for n to 0 to specify SLOW, or to 1 to specify FAST.

For VTCOM.REL:

```
.RUN SIPP<RET>
*VTCOM.REL<RET>
Base?      0<RET>
Offset?    ..FAST<RET>

        Base       Offset        Old       New?
        000000     ..FAST        xxxxxx    n<RET>
        000000     ..FAST+2      ??????    <CTRL/Y><RET>
* <CTRL/C>
.
```

For VTCOM.SAV:

```
.RUN SIPP<RET>
*VTCOM.SAV<RET>
Segment?  1<RET>
Base?      0<RET>
Offset?    ..FAST<RET>

Segment Base       Offset        Old       New?
000001  000000     ..FAST        xxxxxx    n<RET>
000001  000000     ..FAST+2      ??????    <CTRL/Y><RET>
* <CTRL/C>
.
```

● 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.

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

In the customization, the symbol ..DIAL represents the address of the first character in the dial string. You can find the correct value for the symbol ..DIAL in the file CUSTOM.TXT on your distribution kit. Replace the symbols a, b, and so on, with the actual characters that make up your dial string. Add a mandatory null byte (a zero) following the last character of your dial string.

CHANGES AND ADDITIONS TO EXISTING COMPONENTS

To modify VTCOM.REL:

```
.RUN SIPP<RET>
*VTCOM.REL<RET>
Base?      0<RET>
Offset?    ..DIAL<RET>
```

Base	Offset	Old	New?
000000	..DIAL	xxxxxx	;Aa<RET>
000000	..DIAL+1	xxx	;Ab<RET>
000000	..DIAL+2	xxx	;Ac<RET>
.	.	.	.
.	.	.	.
.	.	.	.
000000	..DIAL+50	xxx	0<RET>
000000	..DIAL+51	xxx	<CTRL/Y><RET>

```
*<CTRL/C>
.
```

To modify VTCOM.SAV:

```
.RUN SIPP<RET>
*VTCOM.SAV<RET>
Segment?  1<RET>
Base?     0<RET>
Offset?   ..DIAL<RET>
```

Segment	Base	Offset	Old	New?
000001	000000	..DIAL	xxxxxx	;Aa<RET>
000001	000000	..DIAL+1	xxx	;Ab<RET>
000001	000000	..DIAL+2	xxx	;Ac<RET>
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
000001	000000	..DIAL+50	xxx	0<RET>
000001	000000	..DIAL+51	xxx	<CTRL/Y><RET>

```
*<CTRL/C>
.
```

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 V5.2, observe the following restrictions.

- The CSI (command string interpreter) interface is not supported for UCL; do not issue the R UCL command.

- When booting an XM monitor with VM as the system device, set the base address of the VMX handler using the SET VM BASE=nnnnnn keyboard command. Use the SHOW MEMORY command display to determine the VM handler base address value (nnnnnn). Enter the displayed value for nnnnnn in that command.

- 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).

- If interlace mode is to be used with GIDIS, the SETUP INTERLACE command must be issued prior to running GIDIS. If you are using interlace mode and running GIDIS, you must exit GIDIS before issuing the SETUP NOINTERLACE command.

- 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.

CURRENT RESTRICTIONS AND CORRECTED PROBLEMS

- 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.

- 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 TERM WRAP command or use the terminal hardware setup feature.

- You may encounter problems when customizing the 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.

- Responses for the BUP system utility prompts must be entered at the terminal keyboard. DIGITAL does not recommend entering BUP prompt responses from command files.

- Handlers built by RT-11 V5.1C or V5.2 should not be run on earlier versions of RT-11.

- When you run BUP under the FB monitor, unload any unnecessary foreground jobs to gain more memory. That produces more efficient magtape streaming.

- Programs being run by VBGEXE (unsupported) must not contain interrupt service routines.

- Programs that fetch Ethernet handlers (NC and NQ) must place them above the PAR1 space. The LOAD command automatically places NC and NQ above the PAR1 space.

- The MS handler does not support magtape streaming when running under the XM monitor.

- When running under the XM monitor, programs that fetch magtape handlers must place them above the PAR1 space. The LOAD command automatically places magtape handlers above the PAR1 space.

- 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 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. Alternatively, you can use CCL.

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
```

● 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.

● 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.

● If 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.

● The VM handler may not be used on a PDP-11/23 processor with MSV11 memory, strapped for a 2K word I/O page. The VM handler installs, but any attempt to perform I/O to it immediately returns a hard error. That occurs because the VM handler must turn on the memory management unit to operate. However, when the MMU is enabled, the I/O page reverts to 4K words and the top 2K words of the monitor disappear.

Note that the VMX handler has no problems with this environment.

● 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.

This 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.

● 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).

● 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.

CURRENT RESTRICTIONS AND CORRECTED PROBLEMS

- DIGITAL recommends that you do not use the command SET LS NOHANG. When you use SET LS NOHANG, printers with very large buffers may abort before they are through printing.

- 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 contained on 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.

- There are problems 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. There is no correction for that problem; however, the DIGITAL-recommended work-around is to SET SL KMON (rather than SET SL ON).

- 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.

- 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 may be run on processors that have a monochrome monitor.

- INITIALIZE MS: issued without a tape in the drive can crash the system.

- 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.

- Device handlers must now be linked with the /NOBITMAP option.

- The CSR and vector for the LS handler are factory set for Professional 300 series processors. On other PDP-11 processors, you must set the CSR and vector.

- 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.

- SL does not work with virtual jobs.

- SL and BATCH cannot be used together.

- 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.

- 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).

CURRENT RESTRICTIONS AND CORRECTED PROBLEMS

3.1.1 Current Restrictions When Running FORTRAN under RT-11 V5.2

When running FORTRAN under RT-11 V5.2, observe the following restrictions.

● **Merging FORLIB.OBJ with SYSLIB.OBJ**

For RT-11 V5.2, a new global symbol ($OVTAB) must be excluded when merging the FORLIB.OBJ modules with SYSLIB.OBJ. Failure to exclude $OVTAB when merging the libraries causes multiple values for the same global symbol, and FORTRAN programs do not link properly. Issue the following command in response to the monitor prompt (.) to merge FORLIB.OBJ with SYSLIB.OBJ:

```
.LIBRARY/INSERT/REMOVE SYSLIB FORLIB<RET>
Global?  ERRS<RET>
Global?  ERRTB<RET>
Global?  $OVRH<RET>
Global?  $OVTAB<RET>
Global?  <RET>
```

3.2 CURRENT RESTRICTIONS FOR PROFESSIONAL 300 SERIES PROCESSORS ONLY

When running RT-11 V5.2 on Professional 300 series processors, observe the following restrictions.

● The terminal console display may 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 file UNSUP.TXT.

● 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.

CURRENT RESTRICTIONS AND CORRECTED PROBLEMS

- 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 monitor is 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 a Professional 325 or 350. 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 V5.2 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)

Interlace 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 V5.2.

3.3.1 Automatic Installation

The following problem in automatic installation has been corrected.

CONFIG.COM   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.

CURRENT RESTRICTIONS AND CORRECTED PROBLEMS

3.3.2 Monitors

The following monitor problems have been corrected.

BSTRAP   The system no longer halts when a monitor without memory parity support is booted on a KDJ11-A and a cache parity error occurs.

KMON   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.

MTTINT   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).

RMON   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.

The .ABTIO programmed request now properly aborts outstanding I/O requests for a job.

USR   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.

BINCOM   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. For RT-11 V5.2, 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.

BUP   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. RT-11 V5.2 removes that restriction. 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. For RT-11 V5.2, backup is no longer restricted to a single volume.

When backing up an RLØ2 to a file-structured TSVØ5, BUP gave an incorrect BUP-F-Read error message, although the transfer was completed correctly. For RT-11 V5.2, BUP no longer gives the incorrect error message.

When backing up to a second RLØ2 disk, BUP gave an incorrect ?BUP-W-Incompatible output volume size error message. For RT-11 V5.2, BUP no longer gives the incorrect error message.

When running under the XM monitor, BUP gave an incorrect EOT message when backing up to multiple tape volumes. For RT-11 V5.2, BUP no longer gives the incorrect EOT message.

DIR
DIR no longer prints a ?MON-F- output error message instead of a ?DIR-F- error message if the output device is write-locked.

DUMP
The DUMP/ONLY:n DMØ: command previously caused the DUMP-F-Invalid option /E error message to be displayed when the desired block number was less than 32768 (decimal). The DUMP/ONLY:n DMØ: 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.

DUP
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 ?DUP-F-Invalid command error message.

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.

CURRENT RESTRICTIONS AND CORRECTED PROBLEMS

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? prompt twice. In RT-11 V5.1, the prompt is printed only once.

Error Logger   The error logger did not report the correct time of error. The time of error is now reported correctly.

HELP   Previously, the command HELP ? returned an incorrect error message. Now, that command returns the error message ?HELP-F-HELP not available for ?.

IND   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 ?IND-F-Device error and ?IND-F-Device read error messages 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 ?IND-F-Date file error, Invalid device or unit. Now, the .VOL directive returns the error ?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.

KED   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.

LIBR

The LIBR utility failed if a forms library directory exceeded one block. LIBR now allows forms library directories to exceed one block.

LINK

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. This 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.

LD

Issuing an invalid option to LD occasionally caused an odd address trap. Now, LD correctly returns an invalid option error message and does not trap.

PIP

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.

QUEUE
QUEMAN displayed an error message when you issued the two-slash (//) line terminator option on a line by itself. QUEMAN now accepts the two-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.

RESORC
RESORC now correctly displays the PRO ROM ID.

SETUP
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.

SRCCOM
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.

TRANSF
TRANSF now forces the terminal to NOCRLF when you run RT-11 under a non-multiterminal monitor.

If you type a CTRL/O during a file transfer, the CSI prompt character is now displayed when the transfer is completed.

UCL
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.

3.3.4 Device Handlers

The following device handler problems have been corrected.

DL handler
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.

CURRENT RESTRICTIONS AND CORRECTED PROBLEMS

DU handler
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 (special function bypass) was 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 are no longer required to be in the low 28K words of memory; that restriction is removed.

DY handler
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.

LD handler
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.

LD can now be named with suffixes other than M or X.

LS/LP handler
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.

PI handler
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.

3-14

CURRENT RESTRICTIONS AND CORRECTED PROBLEMS

VM handler
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.

3.3.5 System Generation

The following problems in the system generation program SYSGEN have been corrected.

SYSGEN.COM
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 problem with a programmed request in the system macro library (SYSMAC) has been corrected.

.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.

IASIGN   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.

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.

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.

3.3.8 MACRO-11 Assembler

The following corrections have been made to the MACRO-11 assembler. See Appendix J of the PDP-11 MACRO-11 Language Reference Manual, Update 1, for other MACRO-11 release note information.

* Symbolic expressions were not marked as complex when they contained a symbol from a relocatable psect and a symbol from an absolute psect. That resulted in incorrect linker output if the base of the absolute psect was not zero. Those symbolic expressions are now correctly marked as complex.

* Symbolic expressions were not marked as complex when they contained two symbols from different absolute psects. That resulted in incorrect linker output if the base of either psect was not zero. Those symbolic expressions are now correctly marked as complex.

* When a global symbol was directly assigned the value of the current location counter symbol '.' in an absolute psect, MACRO-11 incorrectly bound the global symbol to the .ABS. psect. MACRO-11 now correctly binds the global symbol to the absolute psect from which the direct assignment was executed.

* MACRO-11 hung in an infinite loop if it read a record with an invalid record size. MACRO-11 now correctly returns an error.

- Labels containing invalid characters caused MACRO-11 to hang in an infinite loop if a macro existed with the same name as the valid part of the label name. MACRO-11 now correctly returns an error.

- When running in memory configurations smaller than 8K words, MACRO-11 exhibited behaviour such as trapping with an invalid EMT error indicating the input .MAC file was not found (when the file did exist), or displaying assembly errors that were not correct. MACRO-11 now runs correctly in memory configurations smaller than 8K words.

- MACRO was processing some index deferred arguments as floating point numbers by default. MACRO now processes all index deferred arguments as octal by default.

- Internal displaced relocatable statements were not being marked as relocatable with an apostrophe (') in the assembly listing. They are now correctly marked.

- Bit 3, an unused bit, was being set in all .PSECT object records. Bit 3 is no longer set. That change causes object files created with the new version of the MACRO-11 assembler to differ from those created with previous versions, resulting in different PAT checksums. However, differences between object files do not always cause differences in task or .SAV image files, as is true in this case.

3.3.9 Miscellaneous Corrections

The following miscellaneous problems have been corrected.

DISMT1.COM   The COPY commands for the files MMHD.SYS, MTHD.SYS, and MSHD.SYS have been removed from the command file DISMT1.COM. Those files are the hardware versions of the magtape handlers and do not contain the FSM. They can be built by SYSGEN from the source modules if needed.

DISMT2.COM   The DISMT2.COM file has been modified (correction of spelling UPSUP) to change the command:

COPY DIS:UNSUP.TXT TAP:UPSUP.TXT/POSITION:-1

to

COPY DIS:UNSUP.TXT TAP:UNSUP.TXT/POSITION:-1

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 new information, and information that was incorrect in or inadvertently left out of the RT-11 documentation set.

3.4.1 RT-11 System Message Manual

Appendix A of this manual contains new system error messages not included in the RT-11 System Message Manual. Under each message is a description of the problem that caused the message to appear and a solution to the problem.

3.4.2 RT-11 Installation Guide

This section describes changes and additions to the RT-11 Installation Guide.

- In Section 2.7.17, Setting VTCOM Default Dial String, the customization patch for the VTCOM.SAV program was left out of the manual. The customization patches for VTCOM.REL and VTCOM.SAV have been modified for RT-11 V5.2, and are included in Section 2.35 of this manual.

- In Section B.5, Table B-6, of the RT-11 Installation Guide, the Contents column for the RK07 DECpack bootstrap loader has incorrect contents for two addresses. The correct contents for the addresses are:

Address	Contents
001024	012711
001026	002021

    The contents for the other listed addresses are correct.

3.4.3 RT-11 System Utilities Manual

This section describes changes and additions to the RT-11 System Utilities Manual.

- Section 11.6.10, Linker Memory Size Option (/K:n), contains inaccurate information. The LINK/K:n option is ignored by RT-11, but is included for compatibility with the RSTS/E operating system. The following is an accurate description of the LINK/K:n option:

    The K:n option lets you insert a value into word 56 of block 0 of the image file. The argument n represents the number of 1K words of memory required by the program; n is an integer in the range 2-28 (decimal). You cannot use the /K:n option with the /R option. The /K:n option is provided for compatibility with the RSTS/E operating system. RT-11 ignores the information provided by the /K:n option, although word 56 of block 0 of the image file is modified.

CURRENT RESTRICTIONS AND CORRECTED PROBLEMS

- Section 19.1, Communication Hardware (hard-wired serial communication):

  The following information clarifies the use of VTCOM to transfer files using a hard-wired serial communication line.

  VTCOM can be used for hard-wired (cabled) serial communication between two processors running RT-11. Hard-wired serial communication between processors is particularly useful when the processors do not have compatible media. An example is transferring files between a Professional 300 series processor using RX50 diskettes, and a PDP-11/23 PLUS processor using an RL02 disk. However, the following description applies to any two interconnected supported PDP-11 processors.

  When transferring ASCII files (using VTCOM) or binary files (using TRANSF), connect the communication port of the Professional processor with a serial line to the console terminal port of the PDP-11/23 PLUS. That connection effectively converts the Professional processor into a terminal for the PDP-11/23 PLUS. With other supported PDP-11 processors, connect the serial line from a serial port of the processor that becomes the terminal, to the console terminal port of the processor that becomes the host. Files can then be transferred between the two processors as described in Chapter 19.

- Section 19.3.2, Loading and Unloading the Handler (VTCOM):

  Section 19.3.2 describes editing out semicolon (;) comment delimiters in a start-up command file. The comment delimiter in start-up command files is in fact an exclamation point (!).

- Section 19.4.2, Table 19-1, VTCOM commands:

  A circumflex mark was omitted before the x in the first command listed in the table. The command should be:

Command	Function
^x	Lets VTCOM transmit CTRL characters.....

- Section 19.5, Transferring ASCII files with VTCOM

  Because some terminals cannot process the form-feed (<FF>) character, the host processor terminal service often converts an embedded form-feed character into a carriage return/multiple line-feed combination before sending a file. Tabs are also often converted. Execute the following command on the host processor before starting a transfer to eliminate those character conversions:

  SET TERM/FORM/TAB

  That command can be included in a log-in or start-up command file on the host system. There is no need to execute that command on the stand-alone system.

- Section 19.6, TRANSF File Transfer Program:

  The following is information on using TRANSF under RTEM. Topics include creating and entering your RTEM account on the host processor, using the RTEM utility FIP to transfer files, and exiting RTEM.

  You must be in your RTEM account on the host processor to transfer files using TRANSF. Once RTEM is installed on the host processor, issue the following commands to enter your RTEM account:

  On an RSX-11 host processor, type:

  >RUN $RTEM

  On a VMS host processor, type:

  $ RUN SYS$SYSTEM:RTEM

  With either operating system, RTEM then prompts RTEM>. Type the start-up option /VS in response to the prompt, in the following manner:

  RTEM>/VS

  RTEM notifies you as it creates your private RTEM account, then returns the RTEM monitor prompt (.).

  Files residing in your host processor account must be transferred to RTEM before they can be sent by TRANSF to a stand-alone system. Files residing on your stand-alone processor are sent by TRANSF to your RTEM account, and must be transferred from RTEM to the host. Use the RTEM utility FIP to transfer files between the host processor and your RTEM account, and between your RTEM account and the host processor. An on-line help file in your RTEM account provides information on running FIP. To view the help file, type:

  .HELP FIP

  To exit RTEM and return to your host processor account, use the RTEM utility JOAT with the /E option, in the following manner:

  .JOAT /E

3.4.4 Introduction to RT-11

On page 11-7 in the <u>Introduction to RT-11</u>, the command for assembling the SUM.MAC program is changed from:

.MACRO SUM/LIST/CROSSREFERENCE

to

.MACRO SUM/LIST/CROSSREFERENCE/DISABLE:GBL

3.4.5 RT-11 System User's Guide

- The LINK/LIMIT:n command option described on page 4-151 contains inaccurate information. The LINK/LIMIT:n option is ignored by RT-11, but is included for compatibility with the RSTS/E operating system. RT-11 stores the specified value n, but ignores that value.

- In the RT-11 System User's Guide, Section 5.5.9, Indirect Control File Processor (IND), Send Data to File (.DATA), contains an inaccurate line in the example on page 5-35. The example that begins:

    .OPEN COPY,TMP

    incorrectly copies DY1:FILE.TST to device DY:. The fourth line of the example should read:

    .DATA DY1:FILE.TST=DY:FILE.TST/W/Y

    This line correctly copies FILE.TST from device DY: to device DY1.

- In Chapter 7, Hardware Characteristics Program (SETUP), the LANG:code description incorrectly states that this mode does not apply to Professional 300 series terminals. In fact, LANG:code mode only applies to Professional 300 series terminals. The beginning of that mode description should read:

    LANG:code   Applies to Professional 300 series computers only.

    For a complete description of LANG:code mode and the national character replacement set, see Sections 2.3.3 (SETUP), 2.25, and 4.5 of this manual.

3.4.6 RT-11 Programmer's Reference Manual

This section contains corrections to the RT-11 Programmer's Reference Manual.

- Section 1.2.5, Linking with FORLIB does not contain complete information. To add FORLIB.OBJ modules to the default library SYSLIB.OBJ, use the following command:

```
.LIBRARY/INSERT/REMOVE SYSLIB FORLIB<RET>
Global?  $ERRS<RET>
Global?  $ERRTB<RET>
Global?  $OVRH<RET>
Global?  $OVTAB<RET>
Global?  <RET>
```

CURRENT RESTRICTIONS AND CORRECTED PROBLEMS

- The following Chapter 2 programmed requests contain the following undocumented error codes.

  .ENTER (Section 2.32)

  2   nonshareable device is already in use by another job.

  4   File sequence number was not found.

  5   File sequence number is invalid or file name is null.

  .LOOKUP (Section 2.46)

  2   File already open on a nonshareable device; for example, magtape.

  5   Magtape file sequence number is invalid.

- .MTSTAT (Section 2.59) contains an inaccurate description of the third address word of the eight-word array where multiterminal status information is returned. It should read:

  addr(3)   The value (0-16 decimal) of the highest numbered terminal control block (TCB) built into the system.

- IFPROT (Section 3.27) contains an inaccurate argument description (filsp should be dblk) and an inaccurate example. The following is the accurate argument description:

  Form: i = IFPROT (chan,dblk,prot)

  where:

  dblk   is the four-word Radix-50 descriptor of the file to be operated on.

  The following is the accurate example:

  This example protects the file SY:RT11FB.SYS against deletion.

  ```
  REAL*4 DBLK(2)
  DATA DBLK/6RSY0RT11FBSYS/
  ICHAN = IGETC()                    !ALLOCATE CHANNEL
  IF (ICHAN.LT.0) STOP 'CANNOT ALLOCATE CHANNEL'
  I=IFPROT(ICHAN,DBLK,1)
  .
  .
  .
  END
  ```

- The IPEEK subroutine description (Section 3.37) contains inaccurate information. The IPEEK subroutine does not necessarily return the contents of the word located at a specified absolute 16-bit memory address. Instead, it returns the contents of the word located at a specified 16-bit address in the current job's address space.

CURRENT RESTRICTIONS AND CORRECTED PROBLEMS

The following is an accurate description of the IPEEK subroutine:

The IPEEK subroutine returns the contents of the word located at a specified 16-bit address in the current job's address space. The subroutine can examine device registers if the registers are located in the current job's address space.

Form: i = IPEEK(iaddr)

where:

> iaddr   is the integer specification of the 16-bit address in the current job's address space to be examined. If this argument is not an even value, a trap results (except on an LSI-11 or a PDP-11/23).

Function Result:

The function result (i) is set to the value of the word examined.

Example:

        ISWIT = IPEEK("44)   !GET VALUE OF JSW

- The IPEEKB subroutine description (Section 3.38) contains inaccurate information. The IPEEKB subroutine does not necessarily return the contents of the byte located at a specified absolute 16-bit memory address. Instead it returns the contents of the byte located at a specified 16-bit address in the current job's address space.

The following is an accurate description of the IPEEKB subroutine:

The IPEEKB subroutine returns the contents of the byte located at a specified 16-bit address in the current job's address space. Since this subroutine operates in a byte mode, the address supplied can be odd or even. The subroutine can examine device registers if the registers are located in the current job's address space. The return is zero extended, that is, the high byte is 0.

Form: i = IPEEKB(iaddr)

where:

> iaddr   is the integer specification of the 16-bit address in the current job's address space to be examined. Unlike the IPEEK subroutine, the IPEEKB subroutine allows odd addresses.

Function Result:

The function result (i) is set to the value of the byte examined.

Example:

        IERR = IPEEKB("53)   !Get error byte

3-23

CURRENT RESTRICTIONS AND CORRECTED PROBLEMS

- The IPOKE subroutine description (Section 3.39) contains inaccurate information. The IPOKE subroutine does not necessarily store a specified 16-bit integer value into a specified absolute memory address. Instead, it stores a specified 16-bit integer value at a 16-bit address in the current job's address space.

  The following is an accurate description of the IPOKE subroutine:

  The IPOKE subroutine stores a specified 16-bit integer value into a 16-bit address in the current job's address space. The subroutine can store values in device registers if the register is in the current job's address space.

  Form: CALL IPOKE(iaddr,ivalue)

  where:

  iaddr   is the integer specification of the 16-bit address in the current job's address space to be modified. If this argument is not an even value, a trap results (except on an LSI-11 or a PDP-11/23).

  ivalue   is the integer value to be stored in the address specified by the iaddr argument.

  Errors:

  None.

  Example:

  To set bit 12 in the JSW without zeroing any other bits in the JSW, use the following procedure.

      CALL IPOKE("44,"10000.OR.IPEEK("44))

- The IPOKEB subroutine description (Section 3.40) contains inaccurate information. The IPOKEB subroutine does not necessarily store a specified 8-bit integer value into a specified absolute memory address. Instead, it stores a specified 8-bit integer value at a 16-bit address in the current job's address space.

  The following is an accurate description of the IPOKE subroutine:

  The IPOKEB subroutine stores a specified 8-bit integer value into a 16-bit address in the current job's address space. Since this subroutine operates in a byte mode, the address supplied can be odd or even. The subroutine can store values in device registers if the register location is in the current job's address space.

3-24

Form: CALL IPOKEB(iaddr,ivalue)

where:

iaddr	is the integer specification of the 16-bit address in the current job's address space to be modified. Unlike the IPOKE subroutine, the IPOKEB subroutine allows odd addresses.
ivalue	is the integer value to be stored in the address specified by the iaddr argument. Only the low byte of ivalue is actually stored.

Errors:

None.

Example:

CALL IPOKEB("53,"20)   !Tell KMON unconditionally fatal

- LOOKUP (Section 3.79) for file I/O operations contains an inaccurate argument description (the count argument does not exist). The following is an accurate description of the LOOKUP function for file I/O operations:

     i = LOOKUP (chan,dblk[,seqnum])

  Disregard the description of the count argument.

3.4.7 RT-11 Mini-Reference Manual

This section contains corrections to the RT-11 Mini-Reference Manual.

- The following programmed requests contain valid undocumented error codes, and incorrect error codes.

  .FPROT

2	Invalid operation (that error code is incorrectly identified as 3).
3	Invalid value for PROT.

  .MTxxx   (all multiterminal programmed requests)

3	Invalid request; user-built emt area in multiterminal programmed request has invalid function code.

  .SFDAT

3	File is protected. (That error code does not exist for .SFDAT.)

- The following BUP (binary compare program) option is not listed in the BUP utilities section:

/Y	Used with /Z to suppress the BUP confirmation message printed during initialization.

● The LOOKUP system subroutine (Section 3.79) function description, File-oriented Operations, contains inaccurate information. The count argument does not exist, and different LOOKUP functions are used for file or system job I/O operations. The following is the accurate description of the LOOKUP functions:

LOOKUP i=LOOKUP (chan,dblk[,segment])

>   Opens an existing file for I/O with the specified channel

>   i=LOOKUP (chan,jobdes)

>   Opens an existing system job for I/O with the specified channel

3.4.8 Software Support Manual

The following section contains corrections to the RT-11 Software Support Manual.

● .WDBBK Macro Example

In Section 4.6.3.3, .WDBBK Macro, Figure 4-43: .WDBBK Macro Example, contains an inaccurate line. The line beginning with symbol WNADR: should be:

WNADR: .WDBBK   5,2*KMMU,2*KMMU,0,0,WS.MAP   ;Window

The rest of the example is correct.

● Multi-Terminal Feature Restrictions

In Section 5.12, Restrictions, restriction 9 contains incomplete information. The restriction applies to the DD and XL handlers also, and should read:

If you plan to devote a terminal line to the DD, LS, or XL handlers, do not specify that terminal's DL interface in the SYSGEN dialog for a multi-terminal system. Do not attempt to attach the terminal from a multi-terminal application program, either.

● Special Directory Operations

Section 7.8.5, Devices with Special Directories, contains incomplete information on the number of special directory operation function codes. Code 5, .RENAME request is left out of the list. The list should read:

Code	Function
1	Close
2	Delete
3	Lookup
4	Enter
5	Rename

Other information in the section concerning the .RENAME request is accurate.

- **Calculating Number of Entries in a Directory Segment**

    Section 9.1.4, Size and Number of Files, contains inaccurate information. If you use no extra words, each segment contains 72 entries. However, the maximum number of entries includes three that are reserved by the operating system and not available to the user. Those three entries in each segment are for end-of-segment, empty (but reserved), and a reserved entry for use when creating a tentative file. Therefore, the maximum number of usable entries per directory segment is 69.

    The formula for calculating the maximum number of entries per device is incorrect. Use the following formula to calculate the maximum number of directory entries, and thus, the maximum number of files.

    31 * ((512 - 5/7 + N) - 3)

    N represents the number of extra information words per directory entry. If N is 0, the maximum number of files you can store on the device is 2139. Disregard the discussion of why the inaccurate -2 is used, and use the corrected (-3) value when solving for S.

    The rest of the discussion of the size and number of files is accurate.

3.4.9 MACRO-11 Language Reference Manual

This section contains corrections to the MACRO-11 Language Reference Manual.

- Section 6.1.4, .IDENT Directive, contains incomplete information concerning inclusion of the .IDENT character string in the Librarian directory listing. That string appears only in RSX librarian directory listings, and not in RT-11 librarian directory listings.

- Section 7.8, MACRO LIBRARY DIRECTIVE: .MCALL, contains incomplete information. Macro definitions called by the .MCALL directive are defined during the first pass of the MACRO assembler, and are not redefined during the second pass. In-line macros are defined during both passes. Therefore, a macro that redefines itself during program execution may not work in both the in-line and .MCALL case.

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 MANUAL INSTALLATION OF THE TK5Ø SOFTWARE DISTRIBUTION

The procedure to manually install the software distribution from a TK5Ø magtape is essentially the same as that described in Chapter 8 of the RT-11 Installation Guide. However, the TK5Ø magtape uses the new MU device handler, so you must specify the new magtape build program MDUP.MU (not included with the RT-11 V5.2 software distribution kit) in response to the MS boot prompt. The rest of the procedure is the same.

4.4 PROCEDURES FOR LINE PRINTERS AND TERMINALS

This section describes procedures you may need to follow to use your line printer or terminal.

4.4.1 SETUP Modes and Controls for Line Printers

The following table lists SETUP modes and controls for line printers supported by the SETUP utility. A Yes in the table indicates support for the indicated mode or control. Dashes (---) indicate no support.

Modes and Controls	LA5Ø Printer	LA1ØØ Printer	LA21Ø Printer	LNØ3 Printer	LQPØ2 Printer
[NO]BOLD	Yes	Yes	---	Yes	Yes
CLEAR	Yes	Yes	Yes	Yes	Yes
DENSE	---	Yes	---	Yes	---
DOWN	Yes	Yes	Yes	Yes	Yes
DRAFT	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	---
LETTER	---	Yes	---	Yes	---
LISTING	Yes	Yes	Yes	Yes	---
MEMO	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

4.4.2 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 it 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

4.4.3 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.

4.4.4 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.

4.4.4.1 **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

4.4.4.2 **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.

4.5 COMPOSE SEQUENCES AND DEAD DIACRITICAL KEYS FOR NRC LANGUAGE KEYBOARDS

This section describes composing characters using the COMPOSE CHARACTER key and the dead diacritical keys. This section also lists the national replacement character (NRC) set language keyboards. Compose sequences and dead diacritical keys vary among language keyboards. Composing characters is only supported for the Professional 300 series processors.

4.5.1 Composing Characters Using the COMPOSE CHARACTER Key

To create a composed character, press the COMPOSE CHARACTER key. That turns on the red Compose light at the top of the keyboard. Next, type the two characters that together form the character you want. Neither typed character is displayed; only the resulting composed character is displayed on the screen. The Compose light goes off.

If you type the wrong character after pressing COMPOSE CHARACTER, press COMPOSE CHARACTER to start again. Pressing the delete key, located at the top right of the main keyboard, cancels COMPOSE CHARACTER.

If you type an invalid sequence of characters, the keyboard beeps and you can retype the characters. Nothing is displayed on the screen.

Valid compose sequences are determined by the particular national replacement character (NRC) set keyboard being used. Use the SETUP KB LANG:code command to choose keyboards (see the RT-11 System User's Guide for complete information).

4.5.2 Composing Characters Using the Dead Diacritical Keys

The COMPOSE CHARACTER key is not used with dead diacritical keys.

Dead diacritical keys are used to construct characters. First press a dead diacritical key and then press another character key. Pressing a dead diacritical key turns on the Compose light; pressing the second character turns the light off. Nothing is displayed until the second key is pressed. Dead diacritical keys are used only with other character keys; they do not function by themselves.

When you press a dead diacritical key during a compose sequence (using the COMPOSE CHARACTER key), the dead diacritical key functions as a normal character key in the compose sequence.

4.5.3 National Replacement Character (NRC) Set Language Keyboards

This section describes the various national replacement character (NRC) set language keyboards for Professional 300 series processors.

Keyboards for 15 languages are supported:

        Canadian (French)
        Danish
        Dutch
        English (United Kingdom)
        English (USA)
        Finnish
        Flemish (Belgium)
        French/Belgian
        German
        Italian
        Norwegian
        Spanish
        Swedish
        Swiss (French)
        Swiss (German)

Subsections 4.5.3.1 through 4.5.3.15 describe those keyboards. The subsection for each language keyboard lists the code(s) that set the keyboard to that language, and a table showing:

- The character created by the compose sequence, listed under the header "Character".

- A short description of that character, listed under the header "Description".

- Whether the character created by a compose sequence is on the keyboard for that language, listed under the header "On Keyboard" as Y for yes or N for no.

- The compose sequences used to create the character, listed under the header "Sequences". When the order of keys pressed is irrelevant, sequences are shown separately on the same line or as one sequence below the other.

- Any dead diacritical keys supported by the language keyboard, listed under the header "Dead Diacritical Keys".

4.5.3.1 **Canadian (French) Language keyboard** – This keyboard is generated by code CDNF.

Char-acter	Description	On Keyboard	Sequences	
"	quotation mark	Y	quotation-mark space space quotation-mark	
#	number sign	Y	+ +	
'	apostrophe	Y	' space	space '
@	commercial at	Y	a a	A A
[	opening bracket	N	((	
\	backslash	Y	/ /	
]	closing bracket	N	))	
^	circumflex	Y	circumflex space space circumflex	
`	grave accent	Y	` space	space `
{	opening brace	N	(-	- (
|	vertical line	Y	/ ^	^ /
}	closing brace	N	) -	-)
~	tilde	Y	~ space	space ~
à	small a with grave	N	grave a	a grave
â	small a with circumflex	N	circumflex a a circumflex	
ç	small c with cedilla	Y	c ,	,c
ê	small e with circumflex	N	circumflex e e circumflex	
î	small i with circumflex	N	circumflex i i circumflex	
ô	small o with circumflex	N	circumflex o o circumflex	
é	small e with acute	Y	acute e	e acute
ù	small u with grave	N	grave u	u grave
è	small e with grave	N	grave e	e grave
û	small u with circumflex	N	circumflex u u circumflex	

Dead Diacritical Keys

` grave accent
^ circumflex
~ tilde

4.5.3.2 Danish Language Keyboard - This keyboard is generated by code DK.

Char- acter	Description	On Keyboard	Sequences	
"	quotation mark	Y	quotation-mark space space quotation-mark	
#	number sign	N	+ +	
'	apostrophe	Y	' space	space '
@	commercial at	N	a a	A A
[	opening bracket	N	((	
\	backslash	N	/ /	
]	closing bracket	N	))	
^	circumflex	Y	circumflex space space circumflex	
`	grave accent	Y	` space	space `
{	opening brace	N	(-	- (
|	vertical line	N	/ ^	^ /
}	closing brace	N	) -	-)
~	tilde	Y	~ space	space ~
Ä	capital A with umlaut	N	umlaut A	A umlaut
Æ	capital AE diphthong	Y	A E	
Ø	capital O with slash	Y	O /	/ O
Å	capital A with ring	Y	A asterisk	asterisk A
Ü	capital U with umlaut	N	umlaut U	U umlaut
ä	small a with umlaut	N	umlaut a	a umlaut
æ	small ae diphthong	Y	a e	
ø	small o with slash	Y	o /	/ o
å	small a with ring	Y	a asterisk	asterisk a
ü	small u with umlaut	N	umlaut u	u umlaut

Dead Diacritical Keys

´ acute accent
` grave accent
·· umlaut
^ circumflex
~ tilde

4.5.3.3 Dutch Language keyboard – This keyboard is generated by code NL.

Char-acter	Description	On Keyboard	Sequences			
"	quotation mark	Y	quotation-mark space space quotation-mark			
#	number sign	Y	+ +			
'	apostrophe	Y	' space		space '	
@	commercial at	Y	a a		A A	
[	opening bracket	N	((			
\	backslash	N	/ /			
]	closing bracket	N	))			
^	circumflex	Y	circumflex space space circumflex			
`	grave accent	Y	` space		space `	
{	opening brace	N	(-		- (	
|	vertical line	Y	/ ^		^ /	
}	closing brace	N	) -		-)	
~	tilde	Y	~ space		space ~	
£	UK pound	Y	- L	L -	- 1	1 -
3/4	3/4 sign	N	3 4		4 3	
ij	IJ sign	N	i j		j i	
1/2	1/2 sign	Y	1 2		2 1	
ƒ	florin	N	f -		- f	
1/4	1/4 sign	Y	1 4		4 1	

Dead Diacritical Keys

^ circumflex
·· umlaut
` grave accent
´ acute accent
~ tilde

4.5.3.4 **English (United Kingdom) Language Keyboard** - This keyboard is generated by code IRL or GB.

Char-acter	Description	On Keyboard	Sequences	
"	quotation mark	Y	quotation-mark space	
			space quotation-mark	
#	number sign	N	+ +	
'	apostrophe	Y	' space	space '
@	commercial at	Y	a a	A A
[	opening bracket	Y	((	
\	backslash	N	/ /	
]	closing bracket	Y	))	
^	circumflex	Y	circumflex space	
			space circumflex	
`	grave accent	Y	` space	space `
{	opening brace	N	(-	- (
|	vertical line	N	/ ^	^ /
}	closing brace	N	) -	-)
~	tilde	Y	~ space	space ~

Dead Diacritical Keys

~ tilde
^ circumflex
` grave accent

4.5.3.5 English (USA) Language Keyboard - This keyboard is generated by code AUS, CDNE, NZ, or USA.

Char-acter	Description	On Keyboard	Sequences	
"	quotation mark	Y	quotation-mark space space quotation-mark	
#	number sign	Y	+ +	
'	apostrophe	Y	' space	space '
@	commercial at	Y	a a	A A
[	opening bracket	Y	((	
\	backslash	Y	/ /	
]	closing bracket	Y	))	
^	circumflex	Y	circumflex space space circumflex	
`	grave accent	Y	` space	space `
{	opening brace	Y	(-	- (
|	vertical line	Y	/ ^	^ /
}	closing brace	Y	) -	-)
~	tilde	Y	~ space	space ~

Dead Diacritical Keys

None

4.5.3.6 Finnish Language Keyboard - This keyboard is generated by code SF.

Character	Description	On Keyboard	Sequences		
"	quotation mark	Y	quotation-mark space space quotation-mark		
#	number sign	N	+ +		
'	apostrophe	Y	' space	space '	
@	commercial at	N	a a	A A	
[	opening bracket	N	((		
\	backslash	N	/ /		
]	closing bracket	N	))		
^	circumflex	Y	circumflex space space circumflex		
`	grave accent	Y	` space	space `	
{	opening brace	N	(-	- (	
|	vertical line	N	/ ^	^ /	
}	closing brace	N	) -	-)	
~	tilde	Y	~ space	space ~	
Ä	capital A with umlaut	Y	quotation-mark A A quotation-mark		
Ö	capital O with umlaut	Y	quotation-mark O O quotation-mark		
Å	capital A with ring	Y	asterisk A	A asterisk	
Ü	capital U with umlaut	Y	quotation-mark U U quotation-mark		
é	small e with acute	N	acute e	e acute	
ä	small a with umlaut	Y	quotation-mark a a quotation-mark		
ö	small o with umlaut	Y	quotation-mark o o quotation-mark		
å	small a with ring	Y	asterisk a	a asterisk	
ü	small u with umlaut	Y	quotation-mark u u quotation-mark		

Dead Diacritical Keys

~ tilde
^ circumflex
` grave accent

4.5.3.7 Flemish (Belgium) Language Keyboard – This   keyboard   is generated by code B.

Character	Description	On Keyboard	Sequences		
"	quotation mark	Y	quotation-mark space	space quotation-mark	
#	number sign	Y	+ +		
'	apostrophe	Y	' space	space '	
@	commercial at	Y	a a	A A	
[	opening bracket	N	((		
\	backslash	N	/ /		
]	closing bracket	N	))		
^	circumflex	Y	circumflex space	space circumflex	
`	grave accent	Y	` space	space `	
{	opening brace	N	(–	– (	
|	vertical line	N	/ ^	^ /	
}	closing brace	N	) –	–)	
~	tilde	Y	~ space	space ~	
£	UK pound	Y	– L L –	– l l –	
à	small a with grave	Y	grave a	a grave	
°	degree sign	Y	ø ^	^ ø	
			space ring	ring space	
			space *	* space	
ç	small c with cedilla	Y	c ,	, c	
§	section	Y	! s s !	! S S !	
			o s s o	o S S o	
			O s s O	O S S O	
			ø s s ø	ø S S ø	
é	small e with acute	Y	acute e	e acute	
ù	small u with grave	Y	grave u	u grave	
è	small e with grave	Y	grave e	e grave	

Dead Diacritical Keys

~ tilde
` grave accent
^ circumflex
•• umlaut

4.5.3.8 **French/Belgian Language Keyboard** - This keyboard is generated by code F.

Char-acter	Description	On Keyboard	Sequences			
"	quotation mark	Y	quotation-mark space			
			space quotation-mark			
#	number sign	Y	+ +			
'	apostrophe	Y	' space		space '	
@	commercial at	Y	a a		A A	
[	opening bracket	N	((			
\	backslash	N	/ /			
]	closing bracket	N	))			
^	circumflex	Y	circumflex space			
			space circumflex			
`	grave accent	Y	` space		space `	
{	opening brace	N	(-		- (	
|	vertical line	N	/ ^		^ /	
}	closing brace	N	) -		-)	
~	tilde	Y	~ space		space ~	
£	UK pound	Y	- L	L -	- 1	1 -
à	small a with grave	Y	grave a		a grave	
°	degree sign	Y	ø ^		^ ø	
			space ring		ring space	
			space *		* space	
ç	small c with cedilla	Y	c ,		, c	
§	section	Y	! s	s !	! S	S !
			o s	s o	o S	S o
			O s	s O	O S	S O
			ø s	s ø	ø S	S ø
é	small e with acute	Y	acute e		e acute	
ù	small u with grave	Y	grave u		u grave	
è	small e with grave	Y	grave e		e grave	

Dead Diacritical Keys

~ tilde
^ circumflex
` grave accent
•• umlaut

4.5.3.9 German Language Keyboard - This keyboard is generated by code A or D.

Char-acter	Description	On Keyboard	Sequences
"	quotation mark	Y	quotation-mark space space quotation-mark
#	number sign	Y	+ +
'	apostrophe	Y	' space space '
@	commercial at	N	a a A A
[	opening bracket	N	((
\	backslash	N	/ /
]	closing bracket	N	))
^	circumflex	Y	circumflex space space circumflex
`	grave accent	Y	` space    space `
{	opening brace	N	(- - (
|	vertical line	N	/ ^ ^ /
}	closing brace	N	) - -)
~	tilde	Y	~ space space ~
§	section	Y	! s s ! ! S S ! o s s o o S S o O s s O O S S O ø s s ø ø S S ø
Ä	capital A with umlaut	Y	quotation-mark A A quotation-mark
Ö	capital O with umlaut	Y	quotation-mark O O quotation-mark
Ü	capital U with umlaut	Y	quotation-mark U U quotation-mark
ä	small a with umlaut	Y	quotation-mark a a quotation-mark
ö	small o with umlaut	Y	quotation-mark o o quotation-mark
ü	small u with umlaut	Y	quotation-mark u u quotation-mark
ß	German sharp s	Y	s s

Dead Diacritical Keys

~ tilde
^ circumflex
` grave accent
′ acute accent

4.5.3.10 Italian Language Keyboard— This keyboard is generated by
code I.

Character	Description	On Keyboard	Sequences				
"	quotation mark	Y	quotation-mark space			
			space quotation-mark			
#	number sign	N	+ +			
'	apostrophe	Y	' space		space '	
@	commercial at	N	a a		A A	
[	opening bracket	N	((			
\	backslash	N	/ /			
]	closing bracket	N	))			
^	circumflex	Y	circumflex space			
			space circumflex			
`	grave accent	Y	` space		space `	
{	opening brace	N	(-		- (	
|	vertical line	N	/ ^		^ /	
}	closing brace	N	) -		-)	
~	tilde	Y	~ space		space ~	
£	UK pound	Y	- L	L -	- 1	1 -
§	section	Y	! s	s !	! S	S !
			o s	s o	o S	S o
			O s	s O	O S	S O
			Ø s	s Ø	Ø S	S Ø
°	degree sign	Y	Ø ^		^ Ø	
ç	small c with cedilla	Y	c ,		, c	
é	small e with acute	Y	acute e		e acute	
ù	small u with grave	Y	grave u		u grave	
à	small a with grave	Y	grave a		a grave	
ò	small o with grave	Y	grave o		o grave	
è	small e with grave	Y	grave e		e grave	
ì	small i with grave	Y	grave i		i grave	

Dead Diacritical Keys

~ tilde
^ circumflex
` grave accent

4.5.3.11 **Norwegian Language Keyboard** - This keyboard is generated by code N.

Character	Description	On Keyboard	Sequences	
"	quotation mark	Y	quotation-mark space space quotation-mark	
#	number sign	N	+ +	
'	apostrophe	Y	' space	space '
@	commercial at	N	a a	A A
[	opening bracket	N	((	
\	backslash	N	/ /	
]	closing bracket	N	))	
^	circumflex	Y	circumflex space space circumflex	
`	grave accent	Y	` space	space `
{	opening brace	N	(-	- (
|	vertical line	N	/ ^	^ /
}	closing brace	N	) -	-)
~	tilde	Y	~ space	space ~
Ä	capital A with umlaut	N	umlaut A	A umlaut
Æ	capital AE diphthong	Y	A E	
Ø	capital O with slash	Y	O /	/ O
Å	capital A with ring	Y	asterisk A	A asterisk
Ü	capital U with umlaut	N	umlaut U	U umlaut
ä	small a with umlaut	N	umlaut a	a umlaut
æ	small ae diphthong	Y	a e	
ø	small o with slash	Y	o /	/ o
å	small a with ring	Y	asterisk a	a asterisk
ü	small u with umlaut	N	umlaut u	u umlaut

Dead Diacritical Keys

~ tilde
′ acute accent
` grave accent
·· umlaut
^ circumflex

4.5.3.12 Spanish Language Keyboard – This keyboard is generated by code SP.

Character	Description	On Keyboard	Sequences	
"	quotation mark	Y	quotation-mark space	space quotation-mark
#	number sign	Y	+ +	
'	apostrophe	Y	' space	space '
@	commercial at	N	a a	A A
[	opening bracket	N	((	
\	backslash	N	/ /	
]	closing bracket	N	))	
^	circumflex	Y	circumflex space	space circumflex
`	grave accent	Y	` space	space `
{	opening brace	N	(-	- (
|	vertical line	N	/ ^	^ /
}	closing brace	N	) -	-)
~	tilde	Y	~ space	space ~
£	UK pound	N	- L L -	- l l -
§	section	N	! s s !	! S S !
			o s s o	o S S o
			O s s O	O S S O
			Ø s s Ø	Ø S S Ø
¡	inverted !	Y	! !	
Ñ	capital N with tilde	Y	tilde N	N tilde
¿	inverted ?	Y	? ?	
°	degree sign	Y	Ø ^	^ Ø
			space ring	ring space
			space *	* space
ñ	small n with tilde	Y	tilde n	n tilde
ç	small c with cedilla	Y	c ,	, c

Dead Diacritical Keys

~ tilde
^ circumflex
` grave accent
´ acute accent
·· umlaut

4.5.3.13 **Swedish Language Keyboard** – This keyboard is generated by code S.

Char-acter	Description	On Keyboard	Sequences	
"	quotation mark	Y	quotation-mark space space quotation-mark	
#	number sign	N	+ +	
'	apostrophe	Y	' space	space '
@	commercial at	N	a a	A A
[	opening bracket	N	((	
\	backslash	N	/ /	
]	closing bracket	N	))	
^	circumflex	Y	circumflex space space circumflex	
`	grave accent	Y	` space	space `
{	opening brace	N	(-	- (
|	vertical line	N	/ ^	^ /
}	closing brace	N	) -	-)
~	tilde	Y	~ space	space ~
É	capital E with acute	N	acute E	E acute
Ä	capital A with umlaut	Y	quotation-mark A A quotation-mark	
Ö	capital O with umlaut	Y	quotation-mark O O quotation-mark	
Å	capital A with ring	Y	asterisk A	A asterisk
Ü	capital U with umlaut	Y	quotation-mark U U quotation-mark	
é	small e with acute	N	acute e	e acute
ä	small a with umlaut	Y	quotation-mark a a quotation-mark	
ö	small o with umlaut	Y	quotation-mark o o quotation-mark	
å	small a with ring	Y	asterisk a	a asterisk
ü	small u with umlaut	Y	quotation-mark u u quotation-mark	

Dead Diacritical Keys

~ tilde
^ circumflex
` grave accent

4.5.3.14 Swiss (French) Language Keyboard - This keyboard is generated by code CHF.

Char- acter	Description	On Keyboard	Sequences	
"	quotation mark	Y	quotation-mark space	space quotation-mark
#	number sign	N	+ +	
'	apostrophe	Y	' space	space '
@	commercial at	N	a a	A A
[	opening bracket	N	((	
\	backslash	N	/ /	
]	closing bracket	N	))	
^	circumflex	Y	circumflex space	space circumflex
`	grave accent	Y	` space	space `
{	opening brace	N	(-	- (
|	vertical line	N	/ ^	^ /
}	closing brace	N	) -	-)
~	tilde	Y	~ space	space ~
ù	small u with grave	N	grave u	u grave
à	small a with grave	Y	grave a	a grave
è	small e with grave	Y	grave e	e grave
ç	small c with cedilla	Y	c ,	, c
ê	small e with circumflex	N	circumflex e e circumflex	
î	small i with circumflex	N	circumflex i i circumflex	
é	small e with acute	Y	acute e	e acute
ô	small o with circumflex	N	circumflex o o circumflex	
ä	small a with umlaut	Y	umlaut a	a umlaut
ö	small o with umlaut	Y	umlaut o	o umlaut
ü	small u with umlaut	Y	umlaut u	u umlaut
û	small u with circumflex	N	circumflex u u circumflex	

Dead Diacritical Keys

~ tilde
^ circumflex
·· umlaut
` grave accent

4.5.3.15 Swiss (German) Language Keyboard - This keyboard is generated by code CHD.

Character	Description	On Keyboard	Sequences	
"	quotation mark	Y	quotation-mark space space quotation-mark	
#	number sign	N	+ +	
'	apostrophe	Y	' space	space '
@	commercial at	N	a a	A A
[	opening bracket	N	((	
\	backslash	N	/ /	
]	closing bracket	N	))	
^	circumflex	Y	circumflex space space circumflex	
`	grave accent	Y	` space	space `
{	opening brace	N	(-	- (
|	vertical line	N	/ ^	^ /
}	closing brace	N	) -	-)
~	tilde	Y	~ space	space ~
ù	small u with grave	N	grave u	u grave
à	small a with grave	Y	grave a	a grave
é	small e with acute	Y	acute e	e acute
ç	small c with cedilla	Y	c ,	, c
ê	small e with circumflex	N	circumflex e e circumflex	
î	small i with circumflex	N	circumflex i i circumflex	
è	small e with grave	Y	grave e	e grave
ô	small o with circumflex	N	circumflex o o circumflex	
ä	small a with umlaut	Y	umlaut a	a umlaut
ö	small o with umlaut	Y	umlaut o	o umlaut
ü	small u with umlaut	Y	umlaut u	u umlaut
û	small u with circumflex	N	circumflex u u circumflex	

Dead Diacritical Keys

~ tilde
^ circumflex
` grave accent
·· umlaut

CHAPTER 5

RUNNING PRO/GIDIS UNDER RT-11

This chapter describes how to use PRO/GIDIS under the RT-11 operating system. Consult the PRO/GIDIS Manual, AD-Y660A-T1, for general information on using PRO/GIDIS. Use this chapter instead of Chapter 2 of that manual.

RT-11 runs PRO/GIDIS only as the foreground job under the XM monitor. RT-11 requires that FPU (floating-point unit) hardware be installed on the Professional 300 series processor running PRO/GIDIS.

PRO/GIDIS requires two files: GIDIS.SAV and ALPH00.FNT. GIDIS.SAV is the utility save image and ALPH00.FNT is the alphabet data structure font table. Both files must be on the system (SY:) device.

Issue the following command to start PRO/GIDIS and make it available to application programs:

    .FRUN GIDIS.SAV

RT-11 provides software access to PRO/GIDIS using three interfaces:

- The MACRO-11 interface (.SPFUN programmed request).

  The MACRO-11 interface to PRO/GIDIS is described in Section 5.2. The .SPFUN programmed request is located in the distributed RT-11 system MACRO library SYSMAC.SML.

- The GIDCAL interface (GIDIS call routines).

  GIDCAL consists of four FORTRAN system subroutines, GIOPEN, GIWRIT, GIREAD, and GICLOS, and is described in Section 5.3. The GIDCAL subroutines are located in the distributed RT-11 system subroutine library SYSLIB.OBJ.

- The FORTRAN interface (ISPFN/ISPFNC/ISPFNF/ISPFNW system subroutines).

  Those system subroutines are used in the same manner as the MACRO-11 .SPFUN programmed requests, and are described in Chapter 3 of the RT-11 Programmer's Reference Manual. An example FORTRAN program using the ISPFNW system subroutine is located in Section 5.4 of this manual. ISPFN/ISPFNC/ISPFNF/ISPFNW are located in the distributed RT-11 system subroutine library SYSLIB.OBJ.

The Professional interface (PI) handler controls the operation of PRO/GIDIS and is transparent to the you. PRO/GIDIS instructions from application programs are sent to and received from PI using any of the above interfaces.

5.1 RT-11 PRO/GIDIS INSTRUCTION SYNTAX

The Professional interface (PI) handler accepts a stream of PRO/GIDIS instructions consisting of operation code (opcode) words and associated parameter blocks. The PI handler passes the opcodes and associated parameter blocks to the GIDIS utility for processing.

Delimiters are not required between PRO/GIDIS instructions. To ensure that PRO/GIDIS treats each individual instruction properly, the opcode word for each instruction contains a length value that specifies the number of parameter words that follow it.

```
       HIGH BYTE         LOW BYTE
    ┌──────────────┬──────────────┐
    │    OPCODE     │    LENGTH     │
    └──────────────┴──────────────┘
```
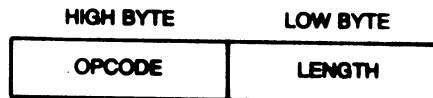
5.1.1 Operation Codes

RT-11 maps PRO/GIDIS instruction names to specific decimal numeric operation codes (opcode). For example, the INITIALIZE instruction has an opcode of 1, while the SET_PRIMARY_COLOR instruction opcode is 21. Appendix A of the PRO/GIDIS Manual provides a list of PRO/GIDIS instructions and their corresponding decimal numeric opcodes.

Programs can define PRO/GIDIS instruction names as numeric constants. For example, in MACRO-11, this could be:

```
    G$INIT=:1.
    G$PRIM=:21.
```

5.1.2 Parameter Blocks

PRO/GIDIS instructions can require a specific (fixed-length) or a variable number of parameters in the parameter block. Indicate a fixed-length parameter block by length values in the range 0 to 254 (decimal). Indicate a variable-length parameter block by an initial length value of 255 (decimal) with a terminating value of -32768 (decimal). Terminating value -32768 is the opcode word for the END_LIST instruction.

Appendix A of the PRO/GIDIS Manual contains a list of length values for all PRO/GIDIS instructions.

The following example illustrates the fixed-length (two parameter words) parameter block for the SET_POSITION instruction:

```
.BYTE    2.,29.     ;Instruction data block length = 2
                    ;Opcode for SET_POSITION instruction = 29
.WORD    100.       ;Horizontal coordinate for current position
.WORD    350.       ;Vertical coordinate for current position
                    ;Following execution of this instruction,
                    ;the current position is 100, 350.
```

The following example illustrates the variable-length parameter block for the DRAW_REL_LINES instruction:

```
.BYTE    255.,26.   ;Variable-length,opcode=DRAW_REL_LINES
.WORD    10.        ;dx1
.WORD    -30.       ;dy1
.WORD    20.        ;dx2
.WORD    +60.       ;dy2
.WORD    -32768.    ;END_LIST instruction opcode word
```

Specifying the wrong type of parameter block causes PRO/GIDIS to ignore that parameter block's instruction but not abort the program. An instruction with insufficient parameters is not executed. If PRO/GIDIS does not recognize an instruction, that instruction and any data following are ignored. After counting and discarding the erroneous instruction's data, PRO/GIDIS proceeds to the next sequential instruction. In each case, the status flag returned by the REQUEST_STATUS instruction is set to FAILURE.

An instruction with extra (too many) parameters is executed as though the extra parameters do not exist. The extra parameters are skipped and PRO/GIDIS proceeds to the next instruction. The status flag returned by the REQUEST_STATUS instruction is set to SUCCESS.

5.2 THE MACRO-11 PRO/GIDIS INTERFACE

.SPFUN 371 writes (sends) PRO/GIDIS instructions, and .SPFUN 370 reads (returns) PRO/GIDIS report-request information. For low-overhead and high-speed device interaction, a number of PRO/GIDIS instructions can be passed between the application program and the Professional interface (PI) handler at one time. You can pass a maximum of 2048 (decimal) words to the PI handler in one SPFUN 371 request.

RT-11 requires PRO/GIDIS to be the highest priority job.

The following is a simplified illustration of the RT-11 PRO/GIDIS data path:
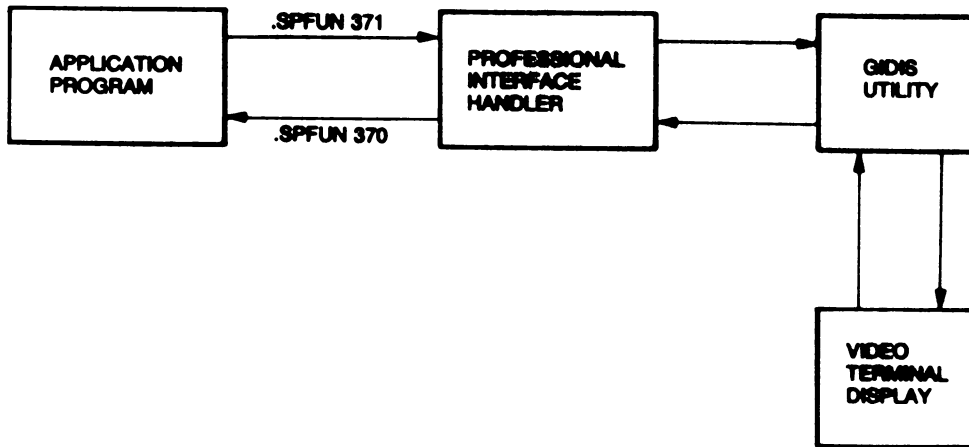
5.2.1 .SPFUN 371

.SPFUN 371 writes (sends) one or more PRO/GIDIS instructions and their associated parameter values to the Professional (PI) interface handler in a buffer. The buffer must begin at an even address. The PI handler passes the buffer to the GIDIS utility for processing.

The following is the structure of the .SPFUN 371 programmed request when used with the PI handler.

Macro Call: .SPFUN area,chan,func,buf,wcnt,blk

where:

area	is the address of a six-word EMT argument block.
chan	is the channel number in the range 0 to 376 (octal).
func	is 371.
buf	is the address of the buffer containing input to the GIDIS utility. Buf must start on a word boundary.
wcnt	is the number of words of information being sent.
blk	is 0 (zero).

.SPFUN 371 can return error codes; see the RT-11 Programmer's Reference Manual for complete information.

Issuing the PRO/GIDIS REQUEST_STATUS instruction returns a report on the success or failure of an instruction sent by .SPFUN 371. Check the carry bit on return from .SPFUN 371 to determine whether the instruction was sent successfully to PRO/GIDIS.

5.2.2 .SPFUN 370

.SPFUN 370 reads (returns) a buffer of information generated from a PRO/GIDIS report handling instruction (sent using .SPFUN 371). PRO/GIDIS report handling instructions are REQUEST_CURRENT_POSITION, REQUEST_STATUS, REQUEST_CELL_STANDARD, etc. The buffer must begin at an even address. The Professional interface (PI) handler passes the buffer address to PRO/GIDIS, and PRO/GIDIS loads the information into the buffer.

The following is the structure of the .SPFUN 370 programmed request when used with the PI handler.

Macro Call: .SPFUN area,chan,func,buf,wcnt,blk

where:

area	is the address of a six-word EMT argument block.
chan	is the channel number in the range 0 to 376(octal).
func	is 370.
buf	is the address of the buffer where the output from the GIDIS utility is to be placed. Buf must start on a word boundary.
wcnt	is the maximum number of words the GIDIS utility can place in the buffer.
blk	is 0 (zero).

.SPFUN 370 can return error codes; see the RT-11 Programmer's Reference Manual for complete information.

5.2.3 Sample MACRO-11 Program

The following example returns the current position of the cursor.

```
        G$RCP=:55.                      ;Specify instruction opcode

        .LOOKUP  #IOAREA,#0,#PIBLK      ;Open PI on channel 0
        BCS      ERROR                  ;Check for success
                                        ;
        .SPFUN   #IOAREA,#0,#371,#REQPOS,#1,#0
                                        ;Send the instruction
                                        ;to REQUEST_CURRENT_POSITION.
        BCS      ERROR                  ;Check for success
                                        ;
        .SPFUN   #IOAREA,#0,#370,#REPBUF,#3,#0
                                        ;Read the current
                                        ;position.
        BCS      ERROR                  ;Check for success
                                        ;
```

```
                    ;.SPFUN 370 causes the following report to be placed
                    ; in REPBUF:
                    ;
                    ; BYTE 2.        (number of data words following).
                    ; BYTE 1.        (CURRENT_POSITION_REPORT identifier).
                    ; WORD x         (PRO/GIDIS coordinates
                    ; WORD y          for current position).
                    ;
                    ; The current position of the cursor will be in the
                    ; second and third words of REPBUF.
             .
             .
             .
IOAREA: .BLKW    6         ; .SPFUN EMT argument block
PIBLK:  .RAD50   /PI /     ; File name in Radix-50 characters
        .WORD    0,0,0     ;
REQPOS: .BYTE    0,G$RCP   ; Length=0,opcode=REQUEST_CURRENT_POSITION.
REPBUF: .BLKB    6         ; Buffer for info returned from GIDIS.
ERROR:                     ; Error handling routine.
```

5.3 THE GIDCAL PRO/GIDIS INTERFACE

GIDCAL, the GIDIS call routines, consists of four FORTRAN system
subroutines that can be used instead of the .SPFUN programmed
requests. For RT-11 V5.2, the only device addressable by GIDCAL is
the full Professional video screen. The four subroutines are located
in the system subroutine library SYSLIB.OBJ.

The following is the correct order for issuing the GIDCAL subroutines:

    1.   GIOPEN   Initiate the GIDIS operation.

    2.   GIWRIT   Send a buffer of GIDIS command data.

    3.   GIREAD   Return a GIDIS report to a buffer.

    4.   GICLOS   End the GIDIS operation.

Each GIDCAL subroutine can return error information in a 2-word status
array. Relevant error codes are listed with each subroutine. A
description of each error code is given in Section 5.3.5.

5.3.1 GIOPEN System Subroutine

The GIOPEN subroutine initiates contact with the Professional
interface (PI) handler, assigns a logical unit number (LUN) for this
GIDIS operation, and initializes the Professional video screen.
Execution of the GIOPEN subroutine produces the same state as
executing the INITIALIZE -1 (complete initialization) instruction
followed by the NEW_PICTURE instruction.

Call: GIOPEN (status,lun,message,msglen,devtype,driver)

where:

        status   is a 2-word integer array used to return a code
                         indicating the results of the requested operation.

lun	is the unit number to associate with this GIOPEN; an integer number from 0 through 15.
	If lun is already connected to a GIDIS operation, status is set to (-5, -4).
message	is a word containing a 0.
msglen	is the number of words in the message; normally a 1. If msglen is less than 0 or greater than 128 (decimal) words, status is set to (-5, -3).
devtype	is 6. Integer values 0 through 5, 7, and 8 are reserved.
driver	is 0.

5.3.2 GIWRIT System Subroutine

The GIWRIT subroutine sends the buffer of GIDIS command data to the Professional interface handler (PI). You can pass a maximum of 2048 (decimal) words to PI in one GIWRIT system subroutine. The data in the buffer need not start or end on a command boundary.

Call: GIWRIT (status,lun,message,msglen)

where:

status	is a 2-word integer array used to return a code indicating the results of the requested operation.
lun	is the unit number assigned by GIOPEN. If no GIOPEN has been executed for the specified value, status is set to (-5, -1).
message	is the command data to send.
msglen	is the number of words in the message. If less than 0 or greater than 2048 (decimal) words, status is set to (-5, -3).

5.3.3 GIREAD System Subroutine

The GIREAD subroutine returns a report from GIDIS requested by a report handling instruction sent by GIWRIT. GIREAD waits until GIDIS returns a report, then places the report in the buffer. If the report is longer than the buffer, the excess is lost. If the report is shorter than the buffer, the trailing words of the buffer are left unchanged; the buffer is not blank-filled. The first byte of the report contains the number of data words in the report.

Call: GIREAD (status,lun,buffer,buflen)

where:

status	is a 2-word integer array used to return a code indicating the results of the requested operation.

lun   is the unit number assigned by GIOPEN. If no
                           GIOPEN has been issued for the specified value,
                           status is set to (-5, -1).

buffer   is the buffer for the report returned by GIDIS.

buflen   is the number of words in the report buffer.

5.3.4 GICLOS System Subroutine

The GICLOS subroutine ends the GIDIS connection to the Professional
interface handler. The output device treats a GICLOS subroutine as an
END_PICTURE instruction. Control is returned to the calling program
once all data specified by the GIWRIT subroutine has been sent to the
output device.

Call: GICLOS (status,lun)

where:

status   is a 2-word integer array used to return a code
                           indicating the results of the requested operation.

lun   is the unit number assigned by GIOPEN to
                           terminate. If no GIOPEN has been sent for the
                           specified value, status is set to (-5, -1).

5.3.5 GIDCAL Error Codes

GIDCAL subroutines can return the following error codes and subcodes
in the 2-word status array. The error code specifies the class of
error and is returned in the first word of the status array. The
subcode specifies the actual error and is returned in the second word
of the status array.

Directive error code (-1)

    The directive error code can return the following subcode:

    -1 No handler. The output device handler is not loaded.

Interface error code (-5)

    The interface error code can return the following subcodes:

    -1 Channel not open. The logical unit number (LUN) for that
        GIDIS is not assigned.

    -2 DEVTYPE is out of range or invalid. The output device
        specified in a GIOPEN is invalid.

    -3 MSGLEN out of range. The message length in a GIOPEN or
        GIWRIT is out of range.

    -4 Channel in use. The logical unit number specified for that
        GIDIS is already in use.

RT-11 specific error code (-7)

The following subcodes can be returned by RT-11 during a GIDIS operation:

-1   Required argument missing. A required argument in a GIDCAL subroutine is not specified.

-2   Handler not loaded. The output device handler is not loaded.

-3   File not found. The indicated file was not found on the device.

-4   File open on nonshareable or non-file-structured device.

-5   An attempt was made to read or write past the end-of-file (EOF) mark.

-6   Hard error. The GIDIS operation experienced a hard error on the output device.

The following errors can be returned by RT-11 (error code is -7) if the .SERR programmed request is in effect:

-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 RMON

-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

5.3.6 Sample GIDCAL Program

The following FORTRAN program fragment uses the GIDCAL subroutines to request the current cursor position.

```
C
C        Declare storage.
C
         INTEGER*2      BUFLEN , LUN , MSGLEN , OCLEN , OPCODE
         INTEGER*2      BUFFER( 3 ) , MESSAG( 1 ) , STATUS( 2 )
C
C        User program begins here...
C
            .
            .
            .
C
C        Assign Logical Unit Number.
C
         LUN      =      5
C
C        Assign opcode (request_current_position) and opcode
C        length (0).
C
         OPCODE   =      55*256
         OCLEN    =      0
C
C        Insert opcode and opcode length into message buffer
C        (one word).
C
         MESSAG( 1 )      = OPCODE + OCLEN
         MSGLEN           = 1
C
C        Send the message to GIDIS
C
         CALL    GIWRIT( STATUS , LUN , MESSAG , MSGLEN )
C
C        Check for errors.
C
         IF      ( STATUS( 1 ) .LE. 0 ) GOTO 999
C
C        Assign buffer length for report.
C
         BUFLEN   =      3
C
C        Get a report from GIDIS.
C
         CALL    GIREAD( STATUS , LUN , BUFFER , BUFLEN )
C
C        Check for errors.
C
         IF      ( STATUS( 1 ) .LE. 0 ) GOTO 999
C
C        Contents of BUFFER after successful return:
C
C        BUFFER( 1 ) = 258 ( (1*256) + 2 )
C                        1 = Report header,
C                        2 = Number of data elements in buffer
C        BUFFER( 2 ) = Current "X" position in GIDIS output space
C        BUFFER( 3 ) = Current "Y" position in GIDIS output space
C
C
```

```
C          User program continues from here...
C
              .
              .
              .
C
C          Handle errors.
C
  999      ...
C
C          End of GIDCAL example.
C
           END
```

5.4 SAMPLE FORTRAN PROGRAM

The following FORTRAN program uses the ISPFNW system subroutine as an example of the FORTRAN PRO/GIDIS interface. It returns the current position of the cursor.

```
C
C          Sample FORTRAN program for PRO/GIDIS.
C
C          Declare storage.
C
           INTEGER*2      RDCPOS , RQCPOS
           INTEGER*2      WCNT , BLOCK , CHAN , STATUS
           INTEGER*2      FILSPC( 4 )

           BYTE           REPBUF( 6 ) , REQBUF( 2 )

           DATA           FILSPC/ 3RPI , 0 , 0 , 0 /
C
C          Assign SPFUN function codes ( Read, Request ).
C
           RDCPOS  =      "370
           RQCPOS  =      "371
C
C          Initialize default values.
C
           BLOCK   =      0
C
C          Get an RT-11 channel.
C
           STATUS  =      IGETC()
           IF             ( STATUS .EQ. -1 ) GOTO 900
           CHAN    =      STATUS
C
C          Open the PI handler.
C
           STATUS  =      LOOKUP( CHAN , FILSPC )
           IF             ( STATUS .NE. 0 ) GOTO 910
C
C          Send the instruction to request from PI the current position.
C
           CODE    =      RQCPOS
           WCNT    =      1
           STATUS  =      ISPFNW( CODE , CHAN , WCNT , REQBUF , BLOCK )
           IF             ( STATUS .NE. 0 ) GOTO 920
C
C          Read the current position.
```

```
C
            CODE      =         RDCPOS
            WCNT      =         3
            STATUS    =         ISPFNW( CODE , CHAN , WCNT , REPBUF , BLOCK )
            IF                  ( STATUS .NE. 0 ) GOTO 930
C
C        User program continues from here...
C
                      .
                      .
                      .
C
C        Close the channel.
C
            STATUS    =         ICLOSE( CHAN )
            IF                  ( STATUS .NE. 0 ) GOTO 940
C
C        Return the channel to RT-11.
C
            STATUS    =         IFREEC( CHAN )
            IF                  ( STATUS .NE. 0 ) GOTO 950
C
C        Go to common exit.
C
            GOTO                1000
C
C        Error messages begin.
C
     900    TYPE                1
       1    FORMAT              ( 1X , 'No channels available.' )
            GOTO                1000
C
     910    TYPE                2
       2    FORMAT              ( 1X , 'Lookup error on PI:.' )
            GOTO                1000
C
     920    TYPE                3
       3    FORMAT              ( 1X , 'Error requesting current position.' )
            GOTO                1000
C
     930    TYPE                4
       4    FORMAT              ( 1X , 'Error reading current position.' )
            GOTO                1000
C
     940    TYPE                5
       5    FORMAT              ( 1X , 'FATAL - SYSTEM ERROR.' )
            GOTO                1000
C
     950    TYPE                6 , CHAN
       6    FORMAT              ( 1X , 'Channel ' I2 ,
      1                            ' is not currently allocated.' )
C
C        Common Exit point.
C
    1000    CALL                EXIT
C
C        End of sample FORTRAN program for PRO/GIDIS.
C
            END
```

5.5 RESTRICTIONS

Observe the following restrictions when running PRO/GIDIS under RT-11:

- Run PRO/GIDIS only under the XM monitor.

- Run PRO/GIDIS only as the foreground job using the FRUN command.

- The area operation instruction PRINT_SCREEN is not supported.

- VT102 emulation is not supported.

CHAPTER 6

GLOBAL REGION SUPPORT IN EXTENDED MEMORY

6.1 GLOBAL REGIONS IN EXTENDED MEMORY

RT-11 Version 5.2 now includes support for creating global regions in extended memory. Global regions are areas of extended memory that are controlled by the operating system, rather than by a particular program. The following are some of the features:

- Global regions can remain allocated in extended memory after a program using the global region has exited.

- More than one program can get access to data stored in global regions.

- You can create up to ten global regions. A program can attach to a combination of up to six local and global regions.

- Programmed requests control the allocation of global regions in extended memory.

- User programs, system utilities, system device handlers, and monitors can create global regions.

- A permanent global region cannot be eliminated. For example, the input/output page resides in a permanent global region and cannot be eliminated.

- Global regions use the same programmed requests and general format as regions local to programs. However, additional arguments to the programmed requests are required. The window mapping and input/output definition blocks are the same. The general structure of the region definition block is the same; however, it contains two additional words (see Section 6.2.1). A new global region control block is added (see Section 6.2.2).

  Global regions use the same directives as local regions.

6.1.1 Creating a Global Region

There is no limit to the size of a global region other than available physical (extended) memory.

When a program creates and attaches to a global region, the program has sole use (possession) of the global region. The program's sole possession is indicated by bit 7 of the global region's control block GR.SHC byte being set.

GLOBAL REGION SUPPORT IN EXTENDED MEMORY

The program must detach from the global region with a .ELRG request to make the global region available for attachment by other programs. Detaching from a global region clears bit 7 of the control block GR.SHC byte. A program wishing to create and attach to a global region, and at the same time make the global region available to other programs, must execute a .ELRG request and then reattach to the global region.

Use the following procedure to create and map to a global region in extended memory.

- From a program, issue a .CRRG programmed request to set up a global region definition block. Specify the .RDBBK macro with a decimal value for R.GSIZ in the same manner as for local regions. Specify status arguments RS.GBL (attach to a global region) and RS.CGR (create a global region).

  Specify status argument RS.AGE if you want to enable automatic global elimination. See Section 6.1.5 for a description of automatic global elimination.

  Assign a name to the created global region.

- Check RS.CRR (bit 15, region definition status word R.GSTS) for success.

- If there is any possibility of a race condition (two programs attempting to attach at exactly the same time), include a .TWAIT request.

- Map to the created region in the same manner as you map to a region local to a program. Use the .CRAW and .MAP requests.

- Check RS.NEW (bit 12, region definition block) to confirm that the program did in fact create the global region, and is not attaching to an already created one.

- If required, load the global region with data.

- If you want to make the global region available to other programs at creation, execute a .ELRG request. Then reattach your program to the global region. If you want to make the global region private to your program, do not execute a .ELRG reattach sequence.

Successful creation of a global region sets RS.CRR and RS.NEW (bits 15 and 12 in global region definition block status word R.GSTS).

The .CRRG programmed request 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

The following example creates a 4K-word global region named MYDATA, and (because of the .ELRG request with a back loop) allows other programs to attach to it.

```
          .MCALL   .CRAW,  .CRRG,   .ELRG,  .EXIT,  .MAP,    .TWAIT
          .MCALL   .RDBBK,  .WDBBK

ERRBYT  =       52
USERRB  =       53
SEVER$  =       10
XE.PRV  =       15


REGION:  .RDBBK   4096./32.,<RS.GBL!RS.CGR>,NAME=MYDATA
WINDOW:  .WDBBK   1,4096./32.
TIME:    .WORD    0,1.*60.                 ;1 second (60 ticks/second)
AREA:    .BLKW    6.                       ;Programmed request area

;+
; The following four entry points are the error handlers for the
; PLAS directives.  They may be expanded as is appropriate.
;-

CRRGER:
CRAWER:
ELRGER:
MAPERR:  BISB     #SEVER$,@#USERRB         ;Indicate severe error
                                           ;to system
         .EXIT                            ; and exit

START:   .CRRG    #AREA,#REGION           ;Create a 4K-word global
                                           ;region, "MYDATA"
         BCC      10$                     ;Branch on success
         CMPB     #XE.PRV,@#ERRBYT        ;Is global region
                                           ;privately owned?
         BNE      CRRGER                  ;If not, then goto
                                           ;.CRRG error handler
         .TWAIT   #AREA,#TIME             ;Wait a second
         BR       START                   ;Try again

;+
; We have now successfully allocated/created the "MYDATA" global
; region.  Now map it to a window, initialize if necessary, and
; then use it.
;-

10$:     MOV      REGION+R.GID,WINDOW+W.NRID ;Put region ID in the WDB
         .CRAW    #AREA,#WINDOW           ;Create a window
         BCS      CRAWER                  ;If error, go handle it
         .MAP     #AREA,#WINDOW           ;Map the window to the region
         BCS      MAPERR                  ;If error, go handle it
         BIT      #RS.NEW,REGION+R.GSTS   ;Did we just create the
                                           ;global region?
         BEQ      30$                     ;Branch if not
```

```
;+
; We just created global region "MYDATA".  We now initialize the
; region with zeros.
;-

        MOV     #4096.,R0                   ;Set up word count of 4K
        MOV     WINDOW+W.NBAS,R1            ;Get virtual address
                                           ;of base of region
20$:    CLR     (R1)+                      ;Clear a word
        SOB     R0,20$                     ; until we're done

;+
; Now detach from global region "MYDATA" to make it shareable.
; Then loop back and reattach to global region "MYDATA".
;-

        .ELRG   #AREA,#REGION              ;Detach from global region
        BCS     ELRGER                     ;Branch if there is an error
        BR      START                      ;Go attach to global
                                           ;region "MYDATA"

;+
; We are attached to a shareable global region.  If we also created
; it, we have initialized the region as well.  The following code can
; use global region "MYDATA" as it sees fit.
;-

30$:    .
        .
        .

        .END    START
```

6.1.2 Attaching to a Global Region

When a program attaches to a global region, RT-11 creates a local region within the global region. The area requested for the attached local region is taken from the memory allocated to the global region, not from the free memory list. The local region is attached to the global region at the base address of the global region. RS.CRR (bit 15 of status word R.GSTS in the global region definition block) is set.

A program cannot attach to a global region unless the program that created the global region has detached (and then probably reattached) from it. See Section 6.1.1 for information.

Specifying a zero value for R.GSIZ (byte offset 2 of region definition block) attaches a program to all memory allocated to an existing global region. The global region must have been previously created by another program. Attempts to attach in this manner to a local region, or to a global region not previously created, returns error code 10 (octal), and RS.CRR is cleared.

Attempting to attach to a local region larger than the global region returns octal error code 7. Attempting to attach to a nonexistent global region returns octal error code 12. Either case clears RS.CRR.

The procedure for attaching to a global region is the same as creating a global region, except that you should not specify the RS.CGR symbol in the status argument of the .RDBBK macro. The following shows the form of the .RDBBK macro for attaching to the global region created in Section 6.1.1:

```
REGION: .RDBBK   4096./32.,<RS.GBL>,NAME=MYDATA
```

Attaching a local region to a global region increments the count of attachments for that global region kept in GR.SHC (byte 5 in global region control block). The reference count in GR.SHC controls automatic global elimination (see Section 6.1.5).

6.1.3 Detaching from a Global Region

Use the .ELRG macro to detach a local region from a global region in the same manner as you use it to eliminate a local region. Memory allocated to the local region within the global region is retained by the global region, and is not returned to the free memory list. The following example illustrates detaching the local region attached in Section 6.1.2:

```
.ELRG    #AREA,#REGION
.
.
.
REGION: .RDBBK   4096./32.,<RS.GBL>,NAME=MYDATA
```

Detaching a local region from a global region decrements the count of attachments for that global region kept in GR.SHC (byte 5 in global region control block). The reference count in GR.SHC controls automatic global elimination (see Section 6.1.5).

6.1.4 Eliminating a Global Region

Use the .ELRG macro to eliminate a global region. Specify the status argument RS.EGR in the .RDBBK macro. The following example eliminates the global region created by the example in Section 6.1.1:

```
.ELRG    #AREA,#REGION
.
.
.
REGION: .RDBBK   4096./32.,<RS.GBL!RS.EGR>,NAME=MYDATA
```

Eliminating a global region returns the memory allocated to the global region to the free memory list.

Observe the following when you eliminate a global region:

- Your program must be attached to a global region to eliminate it.

- The same global region definition block used to create a global region is normally used to eliminate it.

- Permanent global regions cannot be eliminated.

● Attempting to eliminate a global region that is in use by another job returns error code 14 (octal). The global region is not eliminated, but the program requesting the elimination is detached.

The .ELRG programmed request can now return the following new octal error code:

14   Global region in use

6.1.5 Automatic Elimination of Global Regions (AGE)

When you create a global region for use by a number of programs, you can use AGE to automatically eliminate the global region when the last program has finished using it.

Specify the .RDBBK status argument RS.AGE when you create the global region. As explained in Section 6.1.1, a program has sole possession of a global region upon creation (bit 7 of GR.SHC is set). To make the global region available to other programs, the program must clear that bit by detaching with a .ELRG request. Automatic global elimination is turned on when the next program attaches to the global region.

As programs attach and detach local regions to the global region, a reference count is incremented and decremented in GR.SHC (byte 5) in the global region control block. AGE tracks the value in GR.SHC. When the value in GR.SHC is 0, AGE automatically eliminates the global region. The allocated memory is returned to the free memory list.

6.2 GLOBAL REGION DATA STRUCTURES

The following internal data structures are used by RT-11 to create and control the global regions in extended memory.

6.2.1 Region Definition Block

The region definition block is set up by the .RDBBK programmed request. The block has the same structure as a region definition block local to a program, but includes two additional words for the Radix-50 name of the global region. Six bits in the region status word previously unassigned are now assigned for global region status.

Region Definition Block:

R.GID
R.GSIZ
R.GSTS
R.NAME
R.NAME+2

Byte Offset	Symbol	Modifier	Contents
0	R.GID	Monitor's .CRRG routine	A unique region identification. Use it later to reference that region. The region identification is a pointer within the job's impure area to the region control block. The identification for the static region in a virtual job is 0.
2	R.GSIZ	.RDBBK macro or user program	The size of the region you need, in 32-word decimal units. When attaching a local region to an existing global region, specifying a zero value obtains the whole global region.
4	R.GSTS	Monitor's .CRRG routine	The region status word.
6	R.NAME	.RDBBK macro	First word of global region name in RADIX-50 characters.
10	R.NAME+2	.RDBBK macro	Second word of global region name in RADIX-50 characters.

6.2.1.1 **Region Status Word** - The region status word contains information on the status of local and global regions. The following table shows the bits in the region status word and their meaning. Bits 0 through 6 are reserved for future use by DIGITAL.

Bit	Name	Bit Pattern	Meaning When Set
15	RS.CRR	100000	The monitor created the region successfully. The .CRRG routine sets this bit; the .ELRG routine clears it.
14	RS.UNM	40000	One or more windows were unmapped as a result of eliminating the region. The .ELRG routine sets this bit when necessary.
13	RS.NAL	20000	Region was not previously allocated.
12	RS.NEW	10000	An attach request to a global region was made, and the global region was not found. The global region was created.
11	RS.GBL	4000	Create a local region within a global region. If global is region not found, returns error. (No error returned if RS.CGR is set).

Bit	Name	Bit Pattern	Meaning When Set
10	RS.CGR	2000	Create a local region within a global region. If a global region not found, create a global region.
9	RS.AGE	1000	Enable automatic global elimination. Eliminates global region when last job using global region detaches; when count in GR.SHC of global region control block is zero. RS.EGR need not be set.
8	RS.EGR	400	Eliminates global region. Global region area is returned to free memory list. Count in GR.SHC of global region control block must be zero.
7	RS.EXI	200	Eliminate global region when exiting or aborting from job.

6.2.1.2 .RDBDF Macro - The .RDBDF macro defines symbols for the local and global region definition block. It defines the symbolic offset names for the definition block and the names for the region status word bit patterns. In addition, the macro defines the length of the definition block by setting up the following symbol:

    R.GLGH = 10.

The .RDBDF macro does not reserve space for the region definition block.

The format of the .RDBDF macro is as follows:

    .RDBDF

The .RDBDF macro expands as follows:

    R.GID = 0
    R.GSIZ = 2.
    R.GSTS = 4.
    R.GNAM = 6.
    R.GLGH = 10.
    RS.CRR = 100000 (octal)
    RS.UNM = 40000 (octal)
    RS.NAL = 20000 (octal)
    RS.NEW = 10000 (octal)
    RS.GBL = 4000 (octal)
    RS.CGR = 2000 (octal)
    RS.AGE = 1000 (octal)
    RS.EGR = 400 (octal)
    RS.EXI = 200 (octal)

6.2.2 Global Region Control Block

The global region control block is a 5-word area in the resident monitor (RMON). It is set up by the .CRRG programmed request and maintained by the XM monitor. The global region control block contains the size, starting address, status, count of local region attachments, and name assigned to the global region. By default, RT-11 provides 10 (decimal) global region control blocks.

Global Region Control Block:

```
+---------------------+
|       GR.SIZ        |
+---------------------+
|       GR.ADD        |
+----------+----------+
|  GR.SHC  |  GR.STA  |
+----------+----------+
|       GR.NAM        |
+---------------------+
|      GR.NAM + 2     |
+---------------------+
```

Byte Offset	Symbol	Modifier	Contents
0	GR.SIZ	Monitor's .CRRG routine	The size of the global region, expressed in 32-word units. A 0 indicates the global region is not in use.
2	GR.ADD	Monitor's .CRRG routine	The starting physical address of the global region, expressed in 32-word units.
4	GR.STA	.RDBBK macro	The global region status byte.
5	GR.SHC	.RDBBK macro	A count kept of attachments to global region. The count is incremented with each successful .CRRG request, decremented with each successful .ELRG request. Bit 7 set, global region private (no new attachments honored until original attached local region is detached). Bit 7 clear, global region can be attached to by any job.
6	GR.NAM	.RDBBK macro	First name of global region in RADIX-50 characters.
10	. GR.NAM+2	.RDBBK macro	Second name of global region in RADIX-50 characters.

6.2.3 Global Region Control Block Status Byte (GR.STA)

The following table shows the names and meanings of the bits in the global region control block status byte. Bits 0-4 are reserved for future use by DIGITAL.

Bit	Name	Bit Pattern	Meaning When Set
7	GR.AGE	200	Enables automatic global elimination. The global region area is returned to the free memory list when the last job using the global region detaches. Count is kept in GR.SHC.
6	GR.PRM	100	The global region is permanent; it is never eliminated.
5	GR.NRF	40	Do not return the global region's allocated memory to the free memory list when the global region is eliminated.

6.2.3.1 .RDBBK Macro

.RDBBK Macro - The .RDBBK macro defines symbols for the local and global region definition block and reserves space for it. The .RDBBK macro invokes the .RDBDF macro. If you call the .RDBBK macro, you need not call the .RDBDF macro.

The format of the .RDBBK macro is as follows:

```
.RDBBK   RGSIZ,RGSTA,NAME
```

where:

RGSIZ	is the size of the dynamic region, expressed in 32-word decimal units.
RGSTA	is the region status byte.
NAME	is the name of the global region in Radix-50 characters.

CHAPTER 7

ETHERNET HANDLERS

This chapter describes the two new Ethernet handlers, NC and NQ, distributed with RT-11 V5.2. The Ethernet handlers provide support for Ethernet class controllers. The NC Ethernet handler supports the DECNA Ethernet controller for the Professional 300 series processors. The NQ Ethernet handler supports the DEQNA Ethernet controller for Q-BUS processors.

Each handler supports only one controller and a maximum of eight units. A unit is an RT-11 logical device unit (NC0 through NC7, NQ0 through NQ7).

7.1 RESTRICTIONS

Observe the following restrictions for the NC and NQ Ethernet handlers:

- The NC and NQ Ethernet handlers run only under the XM monitor.

- The NC and NQ Ethernet handlers, if called by a .FETCH request, must be placed clear of the PAR1 space. DIGITAL recommends using the LOAD keyboard command to insure the NC or NQ handlers are not loaded into the PAR1 space.

- Programs that call the Ethernet handlers must follow a certain order. Write such programs with the following elements in the order indicated.

    1. Use the .LOOKUP programmed request to open a channel to the device unit.

    2. Allocate the unit using SPFUN 200.

    3. Perform the Ethernet operation(s).

    4. Deallocate the unit using SPFUN 200.

    5. Use the .CLOSE programmed request to close the channel to the specified device unit.

7.2 USE OF SPECIAL FUNCTIONS (.SPFUN)

The NC and NQ Ethernet handlers operate the DECNA and DEQNA Ethernet controllers using the .SPFUN programmed request with the following function codes:

Function	Code
Allocate unit	200
Deallocate unit	200
Reserved	201
Enable protocol type	202
Disable protocol type	202
Enable multicast address	203
Disable multicast address	203
Transmit Ethernet frame	204
Receive Ethernet frame	205

Successful completion of an .SPFUN request clears the carry bit. Completion with error sets the carry bit, and the status word in the buffer contains an error code.

See the RT-11 Programmer's Reference Manual for a complete description of the .SPFUN programmed request.

7.2.1 Allocate/Deallocate Unit (.SPFUN 200)

The allocate unit special function allocates a unit of the Ethernet handler for a job's exclusive use.

The deallocate unit special function deallocates the unit so it can be used by another job.

Allocate Unit

The following is the form of the special function allocate unit (.SPFUN 200):

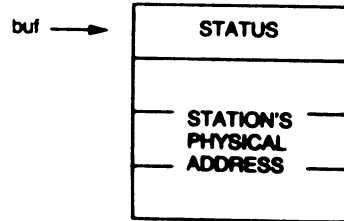Macro Call: .SPFUN   area,chan,func,buf,wcnt,blk[,crtn]

where:

   area   is the address of a six-word EMT argument block.

   chan   is a channel number in the range 0 to 376(octal).

func   is 200.

buf   is the address of a four-word buffer containing the status word and the station's physical address. The buffer contents are returned by the allocate unit special function.

```
buf ──▶  ┌──────────────────┐
         │     STATUS       │
         ├──────────────────┤
         │ ─ STATION'S ─    │
         │   PHYSICAL       │
         │ ─ ADDRESS ─      │
         │                  │
         └──────────────────┘
```

The high byte of the status word contains a 0. Allocate unit returns one of the following octal status codes in the low byte of the status word:

0   Success

2   Controller error while attempting to initialize the network interface (controller)

3   No resources (unit in use)

11   is reserved

wcnt   is 0.

blk   is 1.

crtn   is the entry point of a completion routine.

Deallocate Unit

The following is the form of the special function deallocate unit (.SPFUN 200):

Macro Call: .SPFUN   area,chan,func,buf,wcnt,blk[,crtn]
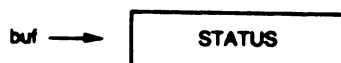
where:

area   is the address of a six-word EMT argument block.

chan   is a channel number in the range 0 to 376(octal).

func   is 200.

buf   is the address of a one-word buffer containing the status word.

```
buf ──▶  ┌──────────────────┐
         │     STATUS       │
         └──────────────────┘
```

The high byte of the status word contains a 0. Deallocate unit returns one of the following octal status codes in the low byte of the status word:

0   Success

1   Unknown unit. The specified unit was not opened by the job issuing the request

2   Controller error while attempting to initialize the network interface (controller)

11   Unit still active

wcnt   is 0.

blk   is 0.

crtn   is the entry point of a completion routine.

7.2.2 Enable/Disable Protocol Type

The enable protocol type special function adds a protocol type to the list of those to be recognized by the unit. For RT-11 V5.2, only one protocol type can be specified for each unit. At least one protocol type must be enabled to receive Ethernet frames.

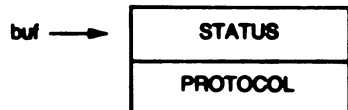The disable protocol special function removes the protocol type from the list of those recognized by the unit.

Enable Protocol Type

The following is the form of the special function enable protocol type (.SPFUN 202):

Macro Call: .SPFUN area,chan,func,buf,wcnt,blk[,crtn]

where:

area   is the address of a six-word EMT argument block.

chan   is a channel number in the range 0 to 376(octal).

func   is 202.

buf   is the address of a two-word buffer that contains the status word followed by the protocol type word.

buf ———▶
STATUS
PROTOCOL

The high byte of the status word contains a 0.   Enable protocol type returns one of the following octal status codes in the low byte of the status word:

0   Success

1   Unknown unit. The specified unit was not opened by the job issuing the request

2   Controller error while attempting to initialize the network interface (controller)

3   No resources (unit's protocol table is full)

6   is reserved

10   Protocol type in use

The protocol type is specified by the user.

wcnt   is 0.

blk   is 1.

crtn   is the entry point of a completion routine.

Disable Protocol Type

The following is the form of the special function disable protocol type (.SPFUN 202):

Macro Call: .SPFUN   area,chan,func,buf,wcnt,blk[,crtn]
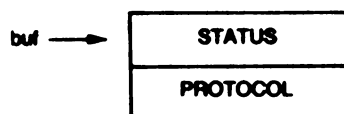
where:

area   is the address of a six-word EMT argument block.

chan   is a channel number in the range 0 to 376(octal).

func   is 202.

buf   is the address of a two-word buffer that contains the status word, followed by the protocol type word.

```
buf ──▶  ┌─────────────┐
         │   STATUS    │
         ├─────────────┤
         │  PROTOCOL   │
         └─────────────┘
```

The high byte of the status word contains a 0. Disable protocol returns one of the following octal status codes in the low byte of the status word:

0   Success

1   Unknown unit. The specified unit was not opened by the job issuing the request.

2   Controller error while attempting to initialize the network interface (controller)

wcnt   is 0.

blk   is 0.

crtn   is the entry point of a completion routine.

7.2.3 Enable/Disable Multicast Address

The enable multicast address special function adds a multicast address to the list of those to be recognized by that unit. You need not specify the unit's physical or broadcast address. RT-11 V5.2 supports only one multicast address per handler unit.

The disable multicast address special function removes a multicast address from the list of those to be recognized by the unit.

Enable Multicast Address

The following is the form of the special function enable multicast address (.SPFUN 203):

Macro Call: .SPFUN   area,chan,func,buf,wcnt,blk[,crtn]
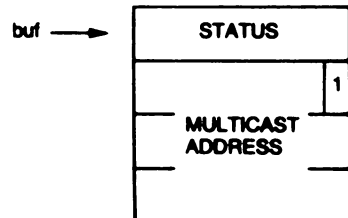
where:

area   is the address of a six-word EMT argument block.

chan   is a channel number in the range 0 to 376(octal).

func   is 203.

buf   is the address of a four-word buffer that contains the status word, followed by the three-word multicast address. The low-order bit of the first address word should be a 1.

The high byte of the status word contains a 0. Enable multicast address returns one of the following octal status codes in the low byte of the status word:

0   Success

1   Unknown unit. The specified unit was not opened by the job issuing the request.

2   Controller error while attempting to initialize the network interface (controller)

3   No resources (unit's address table is full, or hardware address table is full)

wcnt   is 0.

blk   is 1.

crtn   is the entry point of a completion routine.

Disable Multicast Address

The following is the form of the special function disable multicast address (.SPFUN 203):

Macro Call: .SPFUN   area,chan,func,buf,wcnt,blk[,crtn]
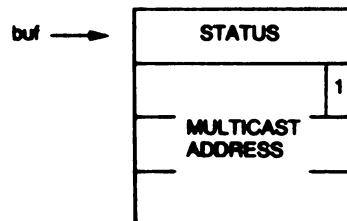
where:

area   is the address of a six-word EMT argument block.

chan   is a channel number in the range 0 to 376 (octal).

func   is 203.

buf   is the address of a four-word buffer that contains the status word, followed by the three-word multicast address. The low-order bit at first address word should be a 1.

buf ———▶ STATUS | 1 | MULTICAST ADDRESS

The high byte of the status word contains a 0. Disable
multicast address returns one of the following octal
status codes in the low byte of the status word:

0   Success

1   Unknown unit. The specified unit was not opened
      by the job issuing the request

2   Controller error while attempting to
      initialize the network interface
      (controller)

wcnt   is 0.

blk   is 0.

crtn   is the entry point of a completion routine.

7.2.4 Transmit Ethernet Frame

The special function transmit Ethernet frame (.SPFUN 204) transmits
the Ethernet frame passed in the buf argument. If the source address
field of the frame is nonzero, it is kept and used. If the source
field of the frame is zero, the unit's physical address is inserted in
the source field before transmission.

The following is the form of the special function transmit Ethernet
frame (.SPFUN 204):

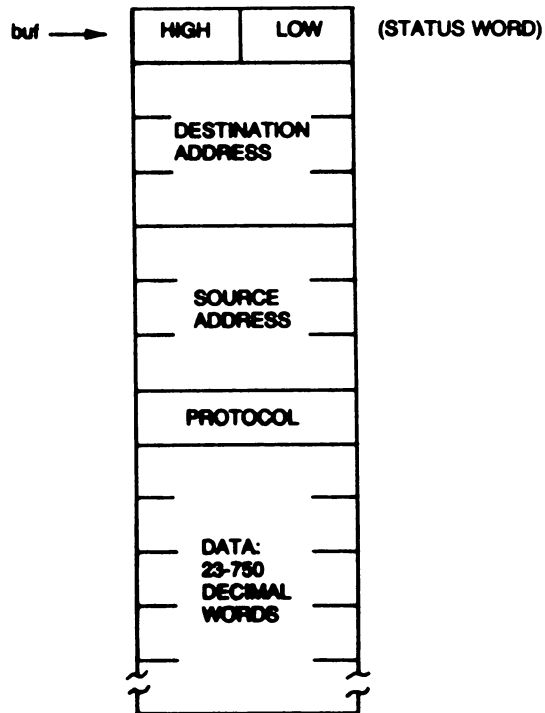Macro Call: .SPFUN area,chan,func,buf,wcnt,blk[,crtn]

where:

area   is the address of a six-word EMT argument block.

chan   is a channel number in the range 0 to 376(octal).

func   is 204.

buf   is the address of a variable-size buffer containing the
      status word, followed by the Ethernet packet to be
      transmitted.

```
buf ──▶  ┌────────┬────────┐  (STATUS WORD)
         │  HIGH  │  LOW   │
         ├────────┴────────┤
         │                 │
         │  DESTINATION    │
         │  ADDRESS        │
         │                 │
         ├─────────────────┤
         │                 │
         │  SOURCE         │
         │  ADDRESS        │
         │                 │
         ├─────────────────┤
         │  PROTOCOL       │
         ├─────────────────┤
         │                 │
         │  DATA:          │
         │  23-750         │
         │  DECIMAL        │
         │  WORDS          │
         │                 │
         └─────────────────┘
```

Transmit Ethernet frame returns one of the following octal status codes in the low byte of the status word:

0   Success

1   Unknown unit. The specified unit was not opened by the job issuing the request

2   Controller error while attempting to initialize the network interface (controller)

13   Transmit failed

When status code 13 is returned in the low byte of the status word, transmit Ethernet frame returns one of the following octal status subcodes in the high byte of the status word:

1   Invalid frame length

2   Excessive collisions

3   Carrier check failed

wcnt   is determined by the variable size of the user buffer (including the status word). The packet size (including the status word) can vary between 31 and 758 (decimal) words.

blk   is 0.

crtn   is the entry point of a completion routine.
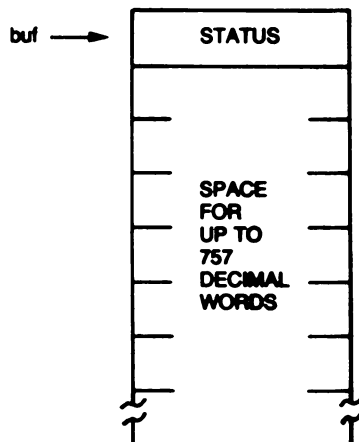
7.2.5 Receive Ethernet Frame

The receive Ethernet frame special function returns the next Ethernet packet with the desired unit address and protocol type to the buffer. The function does not return Ethernet frames that are received with errors.

The following is the form of the special function receive Ethernet frame (.SPFUN 205):

Macro Call: .SPFUN area,chan,func,buf,wcnt,blk[,crtn]

where:

    area   is the address of a six-word EMT argument block.

    chan   is a channel number in the range 0 to 376(octal).

    func   is 205.

    buf   is the address of a variable-size buffer containing the status word, followed by up to 757(decimal) words to receive the Ethernet frame. The buffer contents are returned by the receive Ethernet frame special function.

The high byte of the status word contains a 0. The receive Ethernet frame special function returns one of the following octal status codes in the low byte of the status word:

0   Success

1   Unknown unit. The specified unit was not opened by the job issuing the request

2   Controller error while attempting to initialize the network interface (controller)

wcnt   is the size of the user buffer including the status
            word.

blk   is 0.

crtn   is the entry point of a completion routine.

7.3 EXAMPLE OF ALLOCATING AN ETHERNET UNIT

The following example program presents a suggested method of
allocating a unit of the Ethernet handlers. It works for Professional
300 series processors and Q-BUS processors, and maintains job
separation.

```
            CONFG2  = 370                ;Config word 2 (RMON fixed offset)
            PROS$       = 020000 ;RT is running on a PRO-3xx
                .
                .
                .
            MOV     #<^RNQ >,DBLK        ;Assume non-PRO
            .GVAL   #AREA,#CONFG2        ;Get contents of config word 2
            BIT     #PROS$,R0            ;Running on a PRO?
            BEQ     5$                   ;Nope...
            MOV     #<^RNC >,DBLK        ;Yes, different device name on PRO
5$:         .GTJB   #AREA,#JOBDAT        ;Get info for this job
            MOV     JOBDAT,R0            ;R0 = Job number (*2)
            ASR     R0                   ;Convert to job number 0-7
            ADD     #36,R0               ;Make it final RAD50 digit
            ADD     R0,DBLK              ;And add it to the device name
            .LOOKUP #AREA,#0,#DBLK       ;Open a channel to the 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
                .
                .
                .
```

APPENDIX A

NEW ERROR MESSAGES

The following error messages are new for RT-11 V5.2.

?BUP-F-Output file exists and is protected

An attempt was made to restore a file to a device that contained a protected file of the same name.

Use the monitor UNPROTECT command or the PIP /Z option to change the protection level of the existing file.

?BUP-F-Unexpected error on .SPFUN

An attempt was made to stream a nonstreaming magtape device. The handler did not ignore the inappropriate .SPFUN programmed request.

This error message is not returned by distributed RT-11 magtape handlers.

Magtape handlers should be written to ignore inappropriate .SPFUN requests, and not return hard errors. In particular, magtape handlers should ignore .SPFUN requests to stream magtapes that do not stream.

?BUP-F-Verification error DEV:FILNAM.TYP

The BUP /VERIFY option found differences between the new and original copies of the transferred data.

Retry the operation. If the operation fails again, check the procedures for recovery from hard error conditions listed in Section 2.0. of the <u>RT-11 System Message Manual</u>.

?FORMAT-F-Invalid device for /VERIFY, or not formatted

An attempt was made to verify an unformatted disk.

Format the unformatted disk, install it, load it, and then verify it.

?FORMAT-I-Install DW

A disk must be installed after it is formatted.

This is an informational message; install the disk.

?GIDIS-F-Error attaching to global region VM:FILENAM.TYP

GIDIS was unable to attach to the specified global region.

Issue the SHOW MEMORY command and check if global region is located in extended memory. If not, be sure you are using RT-11 V5.2 or a later version of RT-11.

If the global region is shown in extended memory and the error persists, submit an SPR to DIGITAL. Include with the SPR a program listing and a machine-readable source program, if possible.

?GIDIS-F-Extended memory monitor required for GIDIS.SAV

An attempt was made to run GIDIS on a monitor other than the XM monitor.

Run GIDIS only under the XM monitor.

?GIDIS-F-File not found ALPH00.FNT

An attempt was made to run GIDIS without file ALPH00.FNT on data device DK:. ALPH00.FNT must be on data device DK: for GIDIS to run.

Copy file ALPH00.FNT to data device DK:, and then run GIDIS.

?GIDIS-F-GIDIS must be run as a foreground job

An attempt was made to run GIDIS other than as the foreground job.

GIDIS can be run only as the foreground job using the FRUN GIDIS command.

?GIDIS-F-GIDIS must be run on a PROFESSIONAL series processor

An attempt was made to run GIDIS on a processor other than a Professional 300 series.

GIDIS can only be run on a Professional 300 series processor.

?GIDIS-F-Input error DEV:FILNAM.TYP

GIDIS detected an input/output hard error.

Check the procedures for recovery from hard error conditions listed in Section 2.0 of the RT-11 System Message Manual.

A-2

?GIDIS-F-Insufficient memory

An attempt was made to run GIDIS with less than the minimum required extended memory.

GIDIS requires a minimum of 26K bytes of extended memory. Execute the SHOW MEMORY command and examine the extended memory allocation. Unload any unneeded jobs using extended memory until the required minimum is available to GIDIS.

?KMON-F-Region does not exist<NAME>

An attempt was made to eliminate a global region that does not exist.

Check for a typing error in the command line.

?KMON-F-Region is permanent or active<NAME>

An attempt was made to eliminate a global region while another job was attached to that global region.

A global region cannot be eliminated while another job is attached to it.

?LD-F-/A requires exactly 1 /L

An attempt was made to specify and mount more than one logical disk with a logical name assignment. LD exits and does not prompt for a new command.

Only one logical disk can be mounted when the /A switch is specified. Specify only one /L option with each /A option. Try the operation again.

?LD-F-Switch without required value /L

A unit number (n) was not specified in a SET LDn command.

Specify a unit number (n) for the logical disk when executing a SET LDn command. Try the operation again.

?LD-F-Unable to update LD handler

An attempt was made to execute a SET LD command when the device was write-locked.

Remove write-lock protection from the device and reissue the command.

?LD-W-Unable to update disk copy of LD handler

The system was booted from a write-locked system device. The system does boot.

Remove write-lock protection from system device.

?LINK-W-/S not allowed with STB or MAP files

The /S (/SLOWLY option) was used to produce STB and MAP files in the same linking operation. The files are not produced, but the linking operation continues.

Produce STB and MAP files in separate linking operations.

?MON-F-System read error

A hardware error occured when KMON was being read back into low memory.

Check the procedures for recovery from hard error conditions listed in Section 2.0 of the RT-11 System Message Manual.

?PIP-F-Directory I/O error

The PIP utility found a corrupted directory structure.

Attempt to recover data using the method described in Chapter 9, RT-11 Software Support Manual.

?SETUP-F-Invalid terminator for command: <command>

An attempt was made to execute a SETUP command terminated by an invalid character, such as a colon (:) or equal sign (=) with no parameter.

The only valid terminators for a command that doesn't take a parameter are space, comma (,), or carriage-return (<RET>).

?SYSGEN-E-Ethernet handler support available for XM only

An attempt was made to request Ethernet handler support for a monitor other than the XM monitor.

Request Ethernet handler support only when generating an XM monitor.

?UCL-F-Channel in use DEV:FILENAM.TYP

An internal UCL error occurred in the UCL data file.

Reboot your system and try the operation again. If the error occurs again, get another copy of UCL.SAV from your distribution kit, and retry the operation.

If the error persists, submit an SPR to DIGITAL. Include with the SPR a program listing and a machine-readable source program, if possible.

?UCL-F-Command does not exist

An attempt was made to execute or delete a UCL command that is not in the UCL data file.

Check for correct spelling of the command. Issue the command SHOW COMMANDS and examine the contents of the data file.

?UCL-F-Command too long

The UCL command or the definition of the UCL command contains too many characters.

The UCL command cannot contain more than 16 (decimal) characters including the asterisk. The UCL command definition cannot contain more than 128 (decimal) characters. Check the length of each.

?UCL-F-File not found DEV:FILNAM.TYP

The UCL data file was not found.

The UCL data file is created automatically when the first UCL command is created. Create a UCL command.

?UCL-F-Input error DEV:FILENAM.TYP

UCL detected a hardware error while attempting to read the UCL data file.

Check the procedures for recovery from hard error condition listed in Section 2.0 of the RT-11 System Message Manual.

A-5

?UCL-F-Insufficient memory available

There is insufficient memory available to run UCL.SAV.

Issue the SHOW MEMORY command. Examine the display and unload any unneeded jobs until enough memory is available to run UCL.SAV.

?UCL-F-Invalid command

Incorrect syntax was used to enter a UCL command.

Correct the command syntax and retry the command. A complete description of UCL command syntax is located in the RT-11 System User's Guide.

?UCL-F-Invalid format of DEV:FILNAM.TYP

An attempt was made to access the UCL data file and UCL found a first character other than an exclamation point (!).

The structure of the UCL data file was changed with RT-11 V5.2. See the UCL section in Chapter 2 of this manual for complete details of the new UCL data file.

?UCL-F-Invalid number of commands

Characters following the first character exclamation point (!) in the UCL data file do not form a legal octal number. The legal octal number must be in the range 1-77777.

Perform the appropriate customization patch in the UCL section of Chapter 2 of this manual. The customization patch will enter a legal octal number after the exclamation point in the UCL data file.

?UCL-F-No commands available in DEV:FILNAM.TYP

The UCL data file exists but contains no UCL commands.

Create a UCL command using the correct syntax.

?UCL-F-No more room in file DEV:FILNAM.TYP

The UCL data file is full; no more UCL commands can be entered.

See the UCL section of Chapter 2 of this manual for information on increasing the size of the UCL data file, or delete a UCL command in the file.

?UCL-F-No room on disk for DEV:FILNAM.TYP

The output volume does not have room for the UCL data file.

Refer to Section 3.0 of the RT-11 System Message Manual for information on increasing storage space. If you cannot increase storage space on the disk, you can perform the customization patch described in the UCL section of Chapter 2 of this manual to change the size of the UCL data file.

?UCL-F-Output error DEV:FILNAM.TYP

UCL detected an unrecoverable error while writing to a file.

Check the procedures for recovery from hard error conditions listed in Section 2.0 of the RT-11 System Message Manual.

?VTCOM-E-Invalid port selection

You entered an invalid response to the port selection prompt.

Reissue the SELECT command, and enter a valid port in response to the selection prompt.

?VTCOM-E-No port selected

You pressed the RETURN key in response to the port selection prompt.

Reissue the SELECT command, and enter a valid port in response to the selection prompt.

?VTCOM-F-Timer support required

An attempt was made to run VTCOM under an SJ monitor that did not have SJ timer support specified during the system generation process.

VTCOM requires timer support. Perform a system generation process, and specify SJ timer support.

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. When you encountered 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

  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 its located in your configuration.

- Any additional information you think may be relevant. A clear and precise SPR can expedite the response.

HOW TO ORDER
ADDITIONAL DOCUMENTATION

From	Call	Write
Chicago	312–640–5612 8:15 A.M. to 5:00 P.M. CT	Digital Equipment Corporation Accessories & Supplies Center 1050 East Remington Road Schaumburg, IL 60195
San Francisco	408–734–4915 8:15 A.M. to 5:00 P.M. PT	Digital Equipment Corporation Accessories & Supplies Center 632 Caribbean Drive Sunnyvale, CA 94086
Alaska, Hawaii	603–884–6660 8:30 A.M. to 6:00 P.M. ET or 408–734–4915 8:15 A.M. to 5:00 P.M. PT	
New Hampshire	603–884–6660 8:30 A.M. to 6:00 P.M. ET	Digital Equipment Corporation Accessories & Supplies Center P.O. Box CS2008 Nashua, NH 03061
Rest of U.S.A., Puerto Rico*	1–800–258–1710 8:30 A.M. to 6:00 P.M. ET	

*Prepaid orders from Puerto Rico must be placed with the local DIGITAL subsidiary (call 809–754–7575)

Canada British Columbia	1–800–267–6146 8:00 A.M. to 5:00 P.M. ET	Digital Equipment of Canada Ltd 940 Belfast Road Ottawa, Ontario K1G 4C2 Attn: A&SG Business Manager
Ottawa–Hull	613–234–7726 8:00 A.M. to 5:00 P.M. ET	
Elsewhere	112–800–267–6146 8:00 A.M. to 5:00 P.M. ET	

Elsewhere		Digital Equipment Corporation A&SG Business Manager*

*c/o DIGITAL's local subsidiary or approved distributor

Printed   in   U.S.A.