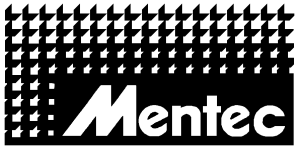
RT–11
System Release Notes

Order Number: AA–5286M–TC

October 1998

This manual contains the technical changes made to the RT–11 Operating System since Version 5.6 and supplemental notes from the Version 5.6 Release Notes applicable to this version.

Revision/Update Information:	This revised manual amends *RT–11 System Release Notes* AA–5286L–TC.
Operating System:	RT–11 Version 5.7



Computers & Communications

October 1998

Printed in the U.S.A.

Mentec, Inc.
Nashua, New Hampshire

Phone: +1(603)883-7711
FAX: +1(603)883-7799

Mentec Ltd.
Dublin, Ireland

Phone: +353 1 205 9797
FAX: +353 1 205 9798

Contents

Chapter 2 Restrictions

Chapter 3 Documentation Updates

Appendix D Accessories

Appendix E RT–11 V5.7 Year 2000 Testing Summary

Tables

Preface

Manual Objectives

The *RT–11 System Release Notes* contains important information for using the RT–11 Version 5.7 Operating System. Read this document before installing and using your system. For ease of transition for those users who are upgrading from versions prior to Version 5.6, edited parts of the RT-11 Version 5.6 System Release Notes are included in the Appendices.

This manual contains the following categories of information:

- Descriptions of new software features as well as changes to existing software features.
- Descriptions of software and hardware restrictions
- Supplementary information which appeared in earlier release notes but is still applicable.
- Corrections to Documentation and Updates

Intended Audience

This manual is intended for all users of the RT–11 operating system.

Document Structure

Chapter 1 describes the new and changed features in RT-11 Version 5.7 including the problems corrected

Chapter 2 describes the restrictions in Version 5.7

Chapter 3 describes documentation updates for Version 5.7

Chapter 4 describes Installation, Bootstrap and Setup Procedures

Appendix A describes the procedures for Submitting a Problem Report

Appendix B describes the Changes from Version 5.5 to 5.6

Appendix C describes the Changes from Version 5.0 to 5.5

Appendix D describes the Unsupported Accessories

Appendix E describes the Year 2000 Testing Summary for RT–11

Associated Documents

The RT–11 Documentation Set consists of the following associated documents:

Basic Books

- *Introduction to RT–11*
- *Guide to RT–11 Documentation*
- *PDP–11 Keypad Editor User's Guide*
- *PDP–11 Keypad Editor Reference Card*
- *RT–11 Commands Manual*
- *RT–11 Mini-Reference Manual*
- *RT–11 Master Index*
- *RT–11 System Message Manual*

Installation Specific Books

- *RT–11 Automatic Installation Guide*
- *RT–11 Installation Guide*
- *RT–11 System Generation Guide*

Programmer Oriented Books

- *RT–11 IND Control Files Manual*
- *RT–11 System Utilities Manual*
- *RT–11 System Macro Library Manual*
- *RT–11 System Subroutine Library Manual*
- *RT–11 System Internals Manual*
- *RT–11 Device Handlers Manual*
- *RT–11 Volume and File Formats Manual*
- *DBG–11 Symbolic Debugger User's Guide*

Conventions used in this Manual

The following conventions are used in this manual

Convention	Meaning
Braces ({ })	In command syntax examples, braces enclose options that are mutually exclusive. You can choose only one option from the group of options that appears in braces.
Brackets ([ ])	Square brackets in a format line represent optional parameters, qualifiers, or values.
[/option1 /option2]	Square brackets in an option summary surrounding two or more options mean that you can specify any one of the enclosed options or any combination of them. The following example means you can specify either the /BEFORE or the /SINCE option or a combination of both to express one or two ranges of dates: [/BEFORE[:date] /SINCE[:date]]
Bold options	**Bold** options in an option summary indicate default options; that is, the option that RT–11 uses if you do not specify any choice of action.
lowercase characters	In command syntax examples, lowercase characters represent elements of a command for which you supply a value. For example: `DELETE filespec`
UPPERCASE characters	In command syntax examples, uppercase characters represent elements of a command that should be entered exactly as given.
RET	RET in examples represents the RETURN key. Unless the manual indicates otherwise, terminate all commands or command strings by pressing RET.
RETURN	RETURN in the text represents the RETURN key.
CTRL/*x*	CTRL/*x* indicates a control-key sequence. While pressing CTRL key, press another key. For example: CTRL/C

1 Features New and Changed in RT–11 Version 5.7

This chapter describes the new features and changes introduced in RT-11 Version 5.7 including new hardware support.

1.1 Summary of New 5.7 Features

These are the main features of RT-11 Version 5.7:

- Year 2000 readiness in the RT-11 monitors and the utilities supporting dates from 1972 through 2099.
- Updated provisions for new and old hardware
- Updated provisions for the Time Of Year (TOY) clocks on the KDJ11-E processors (11/93 and 11/94 based systems)
- Use of the SB monitor for Tape Installation
- Utilities Enhancements
- Corrections to problems in the Monitors, Utilities and Libraries
- Updated Distribution Kit
- Accessories Kit

1.2 Updated Hardware List for Version 5.7

The following hardware has been added:

- SCSI adapter (added but undocumented for Version 5.6)
- SCSI disk (added but undocumented for Version 5.6)
- SCSI tape (added but undocumented for Version 5.6)
- Mentec M-series processors
  - M100 (J11 based)
  - M90 (J11 based)
  - M80 (J11 based)
  - M70 (J11 based)
  - M11 (J11 emulation)
- Time of Year (TOY) clock for Digital's KDJ11-E based processors
- Previously retired devices (1)
- Other SCSI devices(1)

(1) Subject to restrictions - see below

1.2.1 New Devices

Devices from manufacturers other than Digital are being tested on PDP-11 systems as a part of a longevity program by Mentec for potential use in versions of the PDP operating systems. Some of these devices have been used with RT–11 and are mentioned in this manual.

Reference in these Release Notes to a device not in the Software Product Description indicates that the device has been used on RT-11 but not comprehensively tested and certified. Users may encounter unforeseen problems in their use. The use of uncertified hardware is at the users risk. Consult the Software Product Description for a list of formally supported and certified hardware.

Software Performance Reports, or requests for help referring to the use of devices not in the Software Product Description will be considered as suggestions only.

References to a device or manufacturer does not imply any recommendation or endorsement for the device or manufacturer

1.2.2 Digital SCSI Device Support

Support for SCSI devices was added with Version 5.6 but was not documented. The following sections describe the SCSI support of Digital SCSI products added to RT-11.

SCSI Host Adapter Support

Support is included in RT-11 for the Digital RQZX1 SCSI host adapter.

This host adapter appears to the PDP-11 processor as an MSCP controller and / or TMSCP controller for disk and tape respectively.

The RQZX1 adapter can support as many as 7 SCSI devices and 2 RX33 floppy devices, but only one tape unit is supported for use at a time.

SCSI Disk Support

Digital SCSI disks up to 4.3GB are supported by RT–11, these include the RZ23L, RZ24L, RZ26 and RZ29. RT–11 can theoretically provide up to 256 partitions on a drive (approximately 8.6GB) and can address non-file structured disks beyond this size. Use of disks larger than the 4.3GB RZ29 is not supported.

SCSI Tape Support

Digital SCSI tape drives TZ30, TLZ07 are supported by RT-11. Note that only one tape drive should be connected to the RQZX1 host adapter.

1.2.3 Other SCSI Devices

Other devices of potential interest to RT–11 users have been tried with RT–11 Version 5.7 and found to operate satisfactorily under the limited circumstances of the trial. These devices are not necessarily certified for use with RT-11; consult the Software Product Description for currently supported certified hardware.

A brief list is shown below.

- SCSI Controllers
  - SCDC from Andromeda Systems for Q-bus systems
  - CQD series from CMD for Q-bus systems
  - CDU series from CMD for UNIBUS systems
- SCSI Disks from Quantum, Seagate and Western Digital up to 4.3 GB
- Removable disks, such as the Iomega ZIP drive and JAZZ drives
- SCSI tapes including 4mm DAT tape drives of various manufacture
- 3.5" Floppy Disks have been tested on the Andromeda SCDC controller.

Note: Problems may occur if disk drives are interchanged between controllers of different manufacture due to different allocation of the RCT tables.

1.2.4 Mentec M-Series Processors

Support for Mentec M70, M80, M90, M100, and M11 processors has been added to RT–11 Version 5.7 , including access to the On-board serial lines (as DL-11 ports)

These processors will be reported as an 11/73A

1.2.5 TOY Clock Support

Support for the TOY clock on KDJ11-E processors (PDP-11/93 and PDP-11/94) had been updated to continue past the year 2000. Additionally the CTI-bus based processor clock code has been updated for year 2000.

1.2.6 Retired Devices

Handlers for certain retired hardware were removed in RT–11 Version 5.6. They have been returned to the distribution kit for Version 5.7 as unsupported AS-IS software. They are supplied solely on an untested As-Is basis. Support requests and Software Performance Reports (SPRs) referencing these drivers will be treated as suggestions only.

The drivers included are listed below

CR - Card Reader
CT - TA-11 Cassette
DD - DECtape II
DP - RP11/RPR02/RP03
DS - RJS03/RJS04
DT - DECtape
PC - High Speed Paper Tape
PD - PDT-11 Series
RF - RF-11 Fixed disk

1.3 The RT-11 Monitors

The Version 5.7 monitors are similar to the Version 5.6 monitors but different from those of earlier releases. See the *Introduction to RT–11* and the *RT–11 System Internals Manual* for the current descriptions of the monitors.

In RT–11 Version 5.7, any one of the following monitors can be used:

Single-Job Monitors

- RT11SB (A single-job, unmapped monitor, replacing the SJ monitor)
- RT11XB (A single-job, partially mapped monitor)
- RT11ZB (A single-job, fully mapped monitor)

Multi-Job Monitors

- RT11FB (A multi-job, unmapped monitor—the traditional FB monitor)
- RT11XM (A multi-job, partially mapped monitor—the traditional XM monitor)
- RT11ZM (A multi-job, fully mapped monitor)

Notes about the Monitor Descriptions

- A single-job monitor can run only one job at a time.
- An unmapped monitor does not have the extended-memory environment.
- A partially mapped monitor is one that has the traditional XM (extended-memory) environment.
- A fully mapped monitor is one that has both the traditional extended-memory environment and can run a job or jobs both in supervisor mode and in separated I-D space.
- A multi-job monitor can run up to eight jobs at a time.

1.4 Utilities

See the *RT–11 Utilities Manual, Part I* and *Part II* for the current descriptions of RT–11 utilities. See the *RT–11 Device Handlers Manual* for the current descriptions of RT–11 device handlers.

The following list describes some of the enhancements made to the Commands and Utilities for RT-11 V5.7:

1.4.1 All Supplied Utilities

All RT-11 Utilities which accept dates on the command line will now accept either 2 digit years in the date field to cover the period 1972-1999, or 4 digit years to cover the period 1972-2099.

Note that dates above 1999 must be specified in the 4 digit format. Dates in the range 00-72 will be considered invalid.

All utilities which display dates do so in a 4 digit year formats except MACRO-11 which in the headers of the list files will continue to display 2 digit year fields.

Dates supplied to utilities taking CSI format dates must continue to use colon (:) separators for dates with 2 or 4 digit years.

```
12:Jan:2000
1:May:99
```

1.4.2 BUP Utility

The BUP Backup Utility now displays dates with 4 digit years in its listings. The ANSI labels generated by BUP will be written correctly past the year 2000.

1.4.3 DATE Command

The DATE command now takes years in a two digit format 72 to 99 to represent the years 1972 to 1999, and in a four digit format to represent the years 2000 to 2099.

```
.date 21-Oct-98
```

```
.date 1-Jan-2010
```

For examples of valid and invalid date specifications, see Appendix E

DATE without parameters will show the date in a 4 digit year format only.

```
.date
21-Oct-1998
```

1.4.4 DATIME Utility

DATIME will now correctly accept 4 digit years and correctly set the epoch bits in the date word.

1.4.5 DIR Utility

DIR/SORT now works significantly faster on long directories. Tests on a 1100 file directory were six times as fast, and on an 1800 file directory were 12 times as fast.

DIR now processes 4 digit years on input and output and infers years ’72 to ’99 only as 1972-1999, eliminating inconsistencies in date processing

```
.DIR DU2:START*.COM
 21-Oct-1998
STARTS.COM     1P 21-Apr-1998    STARTF.COM     5P 21-Apr-1998
STARTX.COM     8P 21-Apr-1998    STARTB.COM     5P 21-Apr-1998
STARTA.COM    46P 21-Apr-1998
 5 Files, 65 Blocks
 11792 Free blocks
```

```
.dir *.com/since:21:oct:98
 21-Oct-1998
MTB   .COM    16  21-Oct-1998    MSB   .COM    16  21-Oct-1998
MS1B  .COM    16  21-Oct-1998    MUB   .COM    16  21-Oct-1998
KEDTMP.COM     1  21-Oct-1998    DBGSYM.COM     1  21-Oct-1998
 6 Files, 66 Blocks
 11792 Free blocks
```

```
.dir *.com/since:14:oct:1998
 21-Oct-1998
SYSGEN.COM   286  15-Oct-1998    STRTAI.COM    46  15-Oct-1998
1193  .COM     1  20-Oct-1998    MTB   .COM    16  21-Oct-1998
MSB   .COM    16  21-Oct-1998    MS1B  .COM    16  21-Oct-1998
MUB   .COM    16  21-Oct-1998    KEDTMP.COM     1  21-Oct-1998
DBGSYM.COM     1  21-Oct-1998
 9 Files, 399 Blocks
 11792 Free blocks

.dir ms*.com/before:1:jan:2010
 21-Oct-1998
MSBOOT.COM     1P 31-Oct-1992    MSCPCK.COM     1P 31-Oct-1992
MSBONE.COM    15  08-Aug-1997    MSB1  .COM    15  28-Aug-1997
MSB   .COM    16  21-Oct-1998    MS1B  .COM    16  21-Oct-1998
 6 Files, 64 Blocks
 11792 Free blocks
```

DIR now recognizes ANSI tape labels for the years 2000 to 2099 and displays them correctly.

DIR previously output extraneous <NUL> characters under certain circumstances confusing some terminal devices. The characters have been removed.

1.4.6 ERROR Logging

The error logger utility and the error logger display program ERROUT now correctly handle years past 2000, and the ERROUT utility accepts years in two or four digit formats and outputs the From and To dates in the report.

1.4.7 FORMAT Utility

The FORMAT utility has been modified to permit formatting removable RX23 and RX50 media using the Andromeda SCDC Controller (See Section 1.2.1 regarding Support). RX50 diskettes should be formatted using the FORMAT /SINGLEDENSITY command. RX50 and RX33 media should not be interchanged during format, due to differences in the density. Formatting RX33 media as an RX50 may not work satisfactorily in spite of the higher capacity of the RX33.

1.4.8 Handlers

The Ethernet handlers NC, NI, NQ and NU have new special function calls 206, 207 and 210 added for generalized Ethernet applications, and in particular for TCP/IP applications. See the chapter *Documentation Updates* later in these Release Notes for specific details on how to use these calls.

Updated ANSI magtape support to handle the years 2000 through 2099 has been added to the MS, MT, MM and MU magtape handlers.

1.4.9 IND—Indirect Control File Processor Utility

INDIRECT has been modified to cope with 4 digit dates, by adding the <DATE4Y> symbol, which contains the current system date as a with a 4 digit year in an 11 character string, such as "28-AUG-1997" or "4-JUL-2047"

For compatibility with older command files which expect the date in a 9 character string, the DATE symbol has been preserved and will return the last two digits of the year.

The use of the <DATE4Y> symbol has been included in DATIME, SYSGEN, and STRTAI

1.4.10 LIBR—The Librarian

LIBR now displays 4 digit years in its directory listings of libraries.

```
.libr/list:tt: f77ots.olb
RT-11 LIBRARIAN V05.12  WED 21-OCT-1998 15:29:33
DK:F77OTS.OLB           THU 21-JAN-1988 23:41:48

MODULE          GLOBALS         GLOBALS         GLOBALS

                $QSET           $OTI
...
```

1.4.11 LINK—The RT-11 Linker

Link now displays 4 digit years in its map files.

1.4.12 MACRO-11—Macro Assembler

MACRO correctly displays years past the year 2000 in its list file headers in a 2 digit format. It does not display a 4 year format.

1.4.13 PIP Utility

PIP now recognizes ANSI tape header labels for the years 2000 to 2099 and processes them correctly and displays file dates correctly also.

PIP will perform date restricted operations correctly.

1.4.14 QUEUE Utility and QUEMAN

The QUEUEing utility will accept dates with 2 or 4 digit years in accordance with the standard formats wherever a date entry is required.

4 digit dates are now printed on banner pages.

1.4.15 SET dev Command

There appears to be some confusion when reading the documentation regarding the use of the SET commands for setting the CSR and vector of devices and in particular for multi-port devices. Also there is some confusion in the use of the SET command for multi-unit, multi-partition devices.

As noted elsewhere, changes from these commands will not be seen until the system is rebooted, because the changes do not affect the running copy of the driver.

Here are some examples which should help alleviate the confusion ...

```
!
! This command sets the csr and vector for the first DU controller in
! the system, and is referred to as PORT 0
!
.SET DU CSR 172150 VEC 154

!
! This command sets the csr and vector for the second DU controller in
! the system, and is referred to as PORT 1
!
.SET DU CSR1 160334 VEC1 150

! Similarly, the 3rd and 4th controllers will be CSR2 and CSR3
! on PORT 2 and PORT 3 respectively

!
! These commands will put 3 64K block partitions on DU Unit 0 in order
!
.SET DU0 UNIT 0 PART 0  ! DU0: A hardware bootable partition
.SET DU1 UNIT 0 PART 1  ! DU1:
.SET DU2 UNIT 0 PART 2  ! DU2:

!
! Say we want to add another 3 partition drive as unit 1
!
.SET DU3 UNIT 1 PART 0  ! DU3:
.SET DU4 UNIT 1 PART 1  ! DU4:
.SET DU5 UNIT 1 PART 2  ! DU5:

!
! Say we want to add a floppy drive on another controller CSR 160334 VEC 150
!
.SET DU CSR1 160334 VEC 150
.SET DU6 PORT 1 UNIT 0 PART 0 ! DU6:

!
! If we do a SHOW DEV:DU at this point nothing has changed from before if
! DU is our boot device ... We must reboot the system
!
.BOOT DU:

! A SHOW DEV:DU at this stage will show the new DU devices as we set them up
```

1.4.16 SETUP Utility

The SETUP utility DATE function now accepts 4 digit years and correctly sets the PRO and KDJ11-E (11/93 and 11/94) TOY clocks, including the day of the week for the period 1972-2071. Between 2072 to 2099 the utility will indicate the potential ambiguity, because the clock cannot distish the century. The user should check the date on the next system reboot.

```
.setup date:12:jun:98
.date
12-Jun-1998

.setup date:12:jun:2001
.date
12-Jun-2001

.setup date:21:oct:1998
.date
21-Oct-1998
```

1.5 SPOOL handler SP

The SPOOL handler SP now has two new SET OPTIONS

SET SP BIGTIM
SET SP NOBIGTIM

This option will enable and disable the printing of the time and date in large characters on printed banner pages.

1.6 SYSGEN—System Generation Procedure (SYSGEN)

See the *RT-11 System Generation Guide* for the current description of the SYSGEN procedure, sample answer files, a worksheet to help create a system, and a summary description of the RT–11 conditionals used in creating a system.

Features

- Support for the V5.7 monitors (FB, SB, XB, XM, ZB, and ZM).
- Three new distributed answer files (SBFB.ANS, XBXM.ANS, and ZBZM.ANS).
- One updated answer file (XMEL.ANS).
- The <DATE4Y> symbol has been used to display 4 digit years in SYSGEN

1.7 System Macro Library (SYSMAC)

See the *RT–11 System Macro Library Manual* for the current descriptions of all the macros in the system macro library.

1.8 Documentation Kit

The RT–11 Documentation Kit for Version 5.7 remains unchanged from Version 5.6 except for this volume (*System Release Notes*). Corrections to other V5.6 documents are included in these Release Notes.

1.9 Distribution Kit

The RT–11 software distribution kit has changed. See your binary distribution kit and the cover letter sent with it for the contents.

See the *RT–11 Installation Guide* for a brief description of each file that is distributed.

1.10 Corrections in Version 5.7

This section describes the corrections made to RT-11 Version 5.7 in the Monitors, Utilities, Handlers, SYSLIB routines, and SYSGEN command files.

1.10.1 Corrections to the Monitors

The following corrections have been made to the monitors indicated

All Monitors

The .SDTTM programmed requests after the year 2035 were being ignored because of a set sign bit. Fixed.

The date rollover code was not properly rolling between epochs.

Mapped Monitors

The mapped monitors in V5.6 improperly attempted to access the MMR3 register on processors where this register does not exist (11/34, 11/35, 11/40, 11/60) This has been fixed.

1.10.2 Corrections to the Utilities

The following corrections have been made to the utilities for RT-11

BINCOM—Binary File Comparison

BINCOM can compare the contents of devices in non-file structured mode. In particular, if the devices were 65535 blocks long, BINCOM would fail erratically when attempting to compare the last block by not actually comparing the blocks, miscomparing the blocks, or entering an infinite loop. This has been fixed, and BINCOM will now reliably compare this last block.

DBGSYM

DBGSYM would fail when used on processors on processors without the SOB instruction. The requirement for the SOB instruction has been removed.

DIR

DIR could send extraneous ASCII NUL characters to the output device. This has been fixed.

FORMAT

FORMAT would fail when used on processors on processors without the SOB instruction. The requirement for the SOB instruction has been removed.

KED

KED did not handle terminals whose width was set to an odd number of columns. KED will now work correctly with terminals set to an odd number of columns.

RESORC

SHOW DEV:DU and SHOW DEV:MU using the RESORC utility, would display the unit and partition information in octal form while the corresponding SET commands accepted the unit and partition information in decimal form. SHOW DEV now shows this information in decimal form to correspond with the SET command.

SRCCOM

SRCCOM would produce a different list of differences between compared files when used with a .SLP file was requested from when it was not requested. This has been fixed.

SRCCOM would produce an empty .SLP file if the /L:n option was specified. This has been fixed.

1.10.3 Corrections to the Handlers

The following corrections have been made to the handlers for RT-11

MS Tape handler

The MS tape handler would not install properly on processors without the SOB instruction. The requirement for the SOB instruction has been removed.

UNIBUS Map Handler

The UB handler was not being correctly installed on systems with between 128K and 256K Words of memory. The handler will now load correctly.

SP and SPX Spooler handlers

The Spooler handlers have been updated to print 4 digit banner pages.

1.10.4 Corrections to SYSLIB Routines

The following corrections have been made to the SYSLIB routines

IPEEK

The IPEEK routine would incorrectly return the location address, but now correctly returns the value of the location specified.

IREADF, ISDATF, ISPFNF and IWRITF

These FORTRAN completion routines would fail in various manners, including nonsense error codes, or trapping out. These now work correctly.

1.10.5 Corrections to the SYSGEN Command files

The following correction has been made to the SYSGEN command procedure

In Version 5.6, the NQ driver was not in the list of drivers which have set overlays. This has been corrected.

2 Restrictions

This chapter lists the current restrictions applicable to the RT-11 Version 5.7 software.

2.1 Monitors

The following restrictions apply to RT-11 V5.7 monitors:

- RT-11 serializes access to the USR between jobs, but not within a job, and specifically, it does not serialize between a program's mainline code and any completion routines. If mainline code and completion routines both issue USR requests, the completion routine request may interrupt a mainline request and cause unpredictable errors.

  Do not issue USR requests in both mainline and completion routines in the same program, unless the program design precludes the completion routine from issuing a USR request while a request from the mainline is in progress.

- Do not append unit numbers to the system device logical SY or the default data device logical DK. For example, do not use SY0, DK0, SY1 or DK3.

- Processors using the DCJ11 CPU may indicate an incorrect PC value in the message displayed for a monitor trap to 4. See the Processor Restrictions section for details.

- FORTRAN virtual arrays and VM using extended memory cannot be used concurrently when running under the unmapped monitors.

- When running under a mapped monitor, the BA, NQ, NU, SD, and ST handlers must be explicitly loaded with the LOAD command; they cannot be fetched.

- Vector addresses 470 and 474 are reserved for internal use by RT-11.

- Do not FRUN or SRUN privileged foreground jobs containing extended memory overlays; this will cause the system to crash.

- Privileged foreground jobs cannot use virtual overlays because they can crash the system. This does not apply to virtual foreground jobs.

- 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.

- Programs which do not check for wildcard filenames can permit filenames with wildcard characters to be embedded in the name.

- All monitors are built with the bit FBMON$ in the first configuration word $CNFG1 set. You should not access this bit in new programs to test for monitor characteristics. If an existing unmapped user application accesses this bit, and cannot be rebuilt, in order to run, the SET command SET [NO]SJ will change the FBMON$ bit.

- The error message *?KMON-U-Insufficient memory for region* can occur when there is insufficient memory to hold a job's virtual arrays. When this occurs while trying to FRUN or SRUN a job, the system is left in a degraded state. Memory for the USR will have increased and the job's impure area cannot be released. The only way to resolve this problem is to reboot the system.

2.2 Handlers

This section describes the Restrictions applicable the the RT-11 Version 5.7 device handlers.

All Handlers

The following restrictions apply to all RT-11 V5.7 handlers:

- Handlers that are sysgened with device timeout support must be loaded under mapped monitors in order to fix PAR1 in memory. If the handler is fetched into PAR1, PAR1 may not be mapped to the handler when the handler is accessed.
- A number of handlers, in particular LS and XL, are distributed with the same default CSR and vector addresses. If you intend to use two or more of those handlers, before doing so you should:

  Use the appropriate SET command conditions to assign a unique CSR and vector address to each handler

  or

  Perform a system generation (SYSGEN) procedure and specify a unique CSR and vector address for each handler.
- Handlers must be named with two alphabetic characters. A handler named with an alphabetic character and a numeric character will install and load correctly during system boot but is rejected by the INSTALL and REMOVE commands.
- Mapped handlers for bootable devices cannot use virtual overlays. Those handler characteristics and the virtual overlay code make conflicting usage of location 66 in the handler file image. RT-11 for Version 5.7 distributes no handler that is runnable and bootable. Therefore, this restriction does not apply to any distributed RT-11 handler.
- Unloading a device handler which has outstanding I/O requests can hang or crash the system. The job using the handler must be aborted prior to unloading the handler.
- You must LOAD any device handler that is used by a system job. For example, if your system device (SY) uses a different handler than your data device (DK), you must LOAD the data device handler before running any system job that uses that data device.
- You must LOAD any device handler that is used by a foreground job.
- Handlers built by RT-11 V5.5 (or subsequent) or distributed with V5.5 (or subsequent) should not be run on and cannot be built by earlier versions of RT-11.

- You can encounter problems when customizing the SD, SL, or ST handler file name suffix. It is recommended not to use of any suffixes other than X for a mapped monitor with those handlers.
- Device handlers must now be linked with the /NOBITMAP option.
- Some handlers have code executed during UNLOAD which prevents them from being unloaded if they have outstanding I/O (NQ, NU, and XL).
- Only partition 0 of an MSCP or other multipartition drive is hardware bootable. All partitions may be software booted.
- Do not run the utility MSCPCK on a (T)MSCP controller in an environment with MSCP error logging enabled. MSCPCK clears all host-settable controller flags and consequently disables error logging.

DU

- For an MSCP device to be hard-bootable, the MSCP unit number for that device must be the same as the RT-11 unit number.
- Support for DU multiport booting requires a V5.3 (or subsequent) DU handler running under a V5.3 (or subsequent) monitor.

LD

- Do not open the same file on multiple LD units.

LP/LS

- Issuing a PRINT command to the LP or LS handler while the printer is off line or XOFFed (LS only) causes the handler (and possibly the system) to appear hung. Place the printer on line or clear the XOFF condition, or press CTRL/C twice to free the handler.
- It is recommended that you do not use the command SET LS NOHANG. When you use SET LS NOHANG, printers with large buffers can cause the handler to abort the I/O request before they are through printing.

MM/MT

- The SET MM/MT NOODDPAR command is provided only for media compatibility with other operating systems; use it only to exchange magtapes with another system that requires even parity. Do not set even parity (SET MM/MT NOODDPAR) on MM or MT 9-track magtapes for use with Digital controllers.

  Magtapes written for Digital controllers require at least one nonzero bit per byte to ensure that at least one bit is set for a null byte. Even parity violates that requirement. When reading even parity tapes, Digital controllers will either lose (fail to distinguish) null bytes or set bit 4, storing an octal value of 020 with bad parity. The MM and MT handlers cannot distinguish bytes with such created parity errors from those with true parity errors, making magtapes created by BUP, DUP, and PIP unreadable. Similarly, magtape-to-magtape copies will fail.

MU

- It is recommended that the MU handler is LOADed when performing BUP or COPY operations with TMSCP magtape devices.

NQ and NU

- The NQ and NU handlers should be explicitly loaded using the LOAD command; they cannot be fetched.
- An RT-11 monitor will fail to boot on a UNIBUS processor, and hang at the NU boot-time install code, under the following circumstances:
  - — The monitor is built for device timeout support, thereby enabling bootstrap installation of NU.
  - — The UNIBUS processor does not contain a working clock.

NL

- Issuing directory operations to NL (implicitly or explicitly) can produce anomalous results.

TT

- The SET TT NOSCOPE condition has no effect when running the single-line (SL) command editor. Explicitly SET SL OFF to set the terminal to NOSCOPE.
- Unit numbers other than 0 are not supported for TT devices. I/O requests to TTn, where n is other than 0, return a hard error.

VM

- Do not use VM under any circumstances with a 2K I/O page processor.
- When a VM device is created to subsequently become the system (SY) device, the VM handler on the current system device must match the base and size of the VM handler being copied to the VM device. Failure to match the base address of the VM handler and VM device can crash the system when the VM device is booted. Mismatched sizes can cause the VM device to fail to boot. See the *Introduction to RT-11* for more information.
- Do not use unit numbers with VM for the ASSIGN command. For example, do not attempt to assign VM7 to a logical name such as SFD. RT-11 accepts the command without returning an error but does not accept I/O. Note however that you can use unit numbers with VM for the LOAD command.
- Do not attempt to direct I/O to VM units other than 0 (VM or VM0).

XL

- The XL handler should be explicitly loaded, using the LOAD command. It should not fetch be fetched. If VTCOM is run as a system job, it must be loaded.

2.3 Peripherals

The following restrictions apply to the indicated RT-11 V5.7 peripheral devices:

DEQNA Ethernet Controller

- All DEQNA controllers used with RT-11 should be at revision E1/E2 or higher.
- All DEQNA Ethernet controllers used with RT-11 must have the sanity timer disabled. The DEQNA sanity timer reboots the RT-11 system after a period of time under certain circumstances. If you suspect your DEQNA sanity timer is enabled, consult your DEQNA user documentation for information about disabling it.

RC25 Disk Drive

- All RC25's should be spun up with loaded cartridges before boot time. Otherwise, the RT-11 bootstrap will not install them.

RQDX3 Disk Controller (DU)

- The RQDX3 disk controller cannot read data written to RD5x disks by the RQDX1 or RQDX2 controller. Therefore, when upgrading from the RQDX1 or RQDX2 controller to the RQDX3 controller, the RD5x disk must be reformatted. Perform the following procedure:

  1. Back up the disk written by the RQDX1 or RQDX2 controller, using that controller.
  2. Install the RQDX3 controller
  3. Format the disk with an appropriate formatting utility for RD series disks.
  4. Restore the disk from the backup.

RX33 Diskette Drive

- Do not attempt to acquire data on a device being run by a DU controller while you are formatting a diskette on an RX33 drive on the same controller. Formatting RX33 diskettes locks the DU controller for that RX33 diskette drive; all other devices on that controller are unavailable.

TK50 Magtape Drive

- Besides the following information, see also the *RT-11 Device Handlers Manual* for a description of RT-11's support for the MU handler.
- You cannot boot a TK50 magtape device on a UNIBUS processor that also contains a TE16 or TJU16 magtape device controller. The MBOOT primary bootstrap always finds the TE16/TJU16 controller, attempts to boot that device, and returns the error message, ?BOOT-U-I/O error.
- The TK50 magtape drive may not immediately recognize a change in the write protection status of a magtape cartridge if changed while loaded in the drive. Such a change will not be recognized until a significant tape event occurs.

To recognize a change in write protect status unload the magtape cartridge, changing the write protection on the cartridge, and then reload it. The cartridge need not be removed from the drive.

- When issuing a double CTRL/C, there may be a delay in receiving the monitor prompt (.), when aborting operation on the TK50 magtape drive. The TK50 magtape drive will not relinquish control to the monitor until the drive finishes performing an operation.

  The TK50 reads calibration tracks as the first operation on a newly mounted magtape cartridge. If you abort the TK50 while it is reading those tracks, you will not receive the monitor prompt until the calibration is complete.

2.4 Processors

The following restrictions apply to the indicated processors:

- The presence of symbol KXCPU$ in the second configuration word, $CNFG2, does not imply support for KXT11–C or KXJ11–C processors.
- All processors that contain the DCJ11 CPU may indicate an incorrect PC value when a trap occurs due to the following:

  — A word operation to a byte-aligned (odd) address

  — A reference to a nonexistent memory location

  The DCJ11 services aborts immediately rather than waiting until the end of the current macroinstruction. While it is documented that the address returned in the error message, ?MON-F-Trap to 4 <address>, is the address of the instruction following the instruction that caused the error, for the DCJ11, the address could be part of the instruction that caused the error if the error occurred within a multiword instruction.

  In the various debuggers such as DBG–11, ODT, and VDT, commands which expect to proceed from a particular instruction address may produce indeterminate and unexpected results.
- When running RT-11 on an SBC–11/21, the SBC–11/21 must be jumpered for map selection 0, as specified in the *SBC–11/21 Falcon User's Guide*.
- LS, to be used on the available SLU serial port of an SBC–11/21 or SBC–11/21 PLUS processor, must be rebuilt with LS$PRI = 5.
- XL, to be used on the available SLU serial port of an SBC–11/21 or SBC–11/21 PLUS processor, must be rebuilt with XL$PRI = 5 or XL$SBC = 1.
- DD, to be used on the available SLU serial port of an SBC–11/21 or SBC–11/21 PLUS processor, must be rebuilt with DD$PRI = 5.

2.5 System Subroutines and Functions

The following restrictions apply to RT-11 V5.7 system subroutines and functions:

IASIGN

Using FORTRAN-77, if the IASIGN function has been called, do not use the "NAME=" keyword in a subsequent OPEN statement. The FORTRAN-77 OTS will abort and return the error, "Duplicate file specifications". Instead, call IASIGN and specify the device and file specification by using the *idev* and *ifiltyp* parameters and then optionally open the file, using the OPEN statement without the *NAME*= keyword.

IGETR

GETR/IGETR is retained for backwards compatibility only. It may not function correctly. Use MGETR in its place.

IGETR cannot be called from a program built for separated I & D space.

2.6 Programmed Requests

The following restrictions apply to RT-11 V5.7 programmed requests:

- Under a mapped monitor, the initial mapping of a completion routine address specified in any programmed request that performs I/O, must not be changed until the completion routine is executed.
- Under a mapped monitor, programmed requests that perform I/O and pass an address using the *buff* (buffer) parameter must obey the following:
  - — The buffer at the specified address must be mapped as a virtual address, before the request is issued.
  - — Once the request is issued by the EMT (the EMT returns), the buffer can be unmapped by the program. The virtual to physical address translation has, at that point, been performed by the system, and the system stores the physical buffer address.
  - — If a buffer is unmapped after a read operation is issued, the buffer must be remapped before any read data can be accessed.
- Under a mapped monitor, programmed requests that perform I/O and pass an address using any parameter other than *buff* (buffer) must obey the following:
  - — The buffer at the specified address must be mapped as a virtual address before the request is issued.
  - — The system does not perform a virtual to physical address translation for an address in any parameter other than *buff*. Therefore, the virtual address mapping must not be changed during the entire time the request is outstanding, that is:
    - * For nonwait, noncompletion I/O, until a .WAIT request succeeds on the channel.
    - * For wait mode I/O, until the request returns.

  * For completion mode I/O, until the completion routine is entered.

- Using the .CDFN macro call: The *RT-11 System Macro Library Manual* recommends that the .CDFN request be issued before any I/O is initiated. Any .CDFN must be issued, if required, before initiating I/O on a handler that can post an I/O request without necessarily expecting satisfaction of that request; a handler for which RMON does not decrement I.IOCT, such as MQ or the Ethernet handlers NQ, NU, and XL. Issuing a .CDFN after initiating I/O on such a handler is restricted.
- The following requests and subroutines perform I/O contiguously from the specified buffer starting address: .READx,.WRITx, IREADx, IWRITx, JREADx, JWRITx, .SPFUN 376 and .SPFUN 377, ISPFN 376 and ISPFN 377. If the specified buffer spans a discontiguously mapped page boundary, I/O is performed outside the mapped buffer, and no error message is returned.
- The .DRDEF macro call generates an incorrect value in offset 100 of block 0 if the handler name is AA and extended unit number support is requested. Use any other unused name for a user written handler.

2.7 Utilities

The following restrictions apply to the indicated RT-11 V5.6 utilities:

2.7.1 BINCOM

- After encountering an error, such as an I/O error on an input or output file, BINCOM will report the error but may not output the current buffer of differences, in which case, do not rely on having received a report of all the differences in that buffer.
- When using wildcards, the wildcard expression must be the same for both input file specifications. For example, the following expression is valid:

  ```
  .DIFF/BIN DU0:*.* DU1:*.*
  ```

  The following, on the other hand, is not valid:

  ```
  .DIFF/BIN DU0:*.* DU1:
  ```

2.7.2 BUP

- DL (RL01/02) and DM (RK06/07) disks require a two-step initialization process before being used as backup media. First, these disks must be initialized by DUP, using the INITIALIZE/REPLACE command which creates the software bad-block replacement table on the volume. Then, these disks must be initialized by BUP, using the BACKUP/INITIALIZE command, so bad-block replacement is performed transparently by the device handler during backup operations.

- Do not use BUP to back up DL or DM non-RT-11 file-structured volumes. DL and DM volumes contain RT-11-specific software bad-block replacement tables in their home block. BUP can back up but not restore such volumes.
- BUP does not support 100 in/s streaming for the TS05 magtapes under a mapped monitor. BUP does support 25 in/s operation for the TS05 magtapes under a mapped monitor.
- When you run BUP under the FB monitor, unload any unnecessary foreground jobs to gain more memory, which produces more efficient magtape streaming.

2.7.3 DBG–11

- To use the symbolic debugger, SD must be copied or renamed to SD.SYS or SDX.SYS, depending on the environment in which it is used. It may not be run under another name.
- If you run the soft-I/O version of SD with an active foreground job (for example, VTCOM), the system ignores the keyboard entries due to a context switch that occurs when SD gives up control to allow I/O to complete. The correct context is not restored, and RT–11 directs the keyboard input to the wrong buffer.
- Do not issue the SHOW ALL or SHOW CONFIGURATION commands when SD is loaded; doing so can return a trap through location 4.
- You cannot use a mapped monitor version of the soft I/O handler (SDSX.SYS) concurrently with a system job. The first access of the soft I/O handler crashes the system. Unload any system jobs before using the soft I/O version of the DBG–11 handler under a mapped monitor.
- All terminals used with DBG–11 must have hardware tabs.

2.7.4 DUP

- The command COPY/DEVICE under Version 5.7 cannot be used to copy images from magtapes to disks when the magtapes were created using RT-11 V4. Use RT-11 V4 for that operation.
- FORMAT/VER[:ONLY] with DUP

  After performing a FORMAT/VER[:ONLY] on RK06/07 and RL01/02 disks, you should perform an INITIALIZE of the disk to ensure that the home block (block 1) information regarding any bad-block replacement has been initialized.
- BOOT/FOREIGN XXn:

  The BOOT/FOREIGN XXn: command is restricted in the case where SY: is XXn. Unit "n" of Device XXn must be on the same controller as XXn.

  The BOOT/FOREIGN XXn: command can be used only for operating systems that tolerate interrupts enabled on device SSn: at hardware boot time. Specifically excluded from the BOOT/FOREIGN XXn: command is the case of XXDP on XXn:.

2.7.5 ERROR LOGGER

- You cannot run more than one Error Logger, and the job name for that single Error Logger must be ERRLOG.
- The error logger no longer logs memory errors although the error logger report continues to produce memory error statistics. (The counts are always 0)

2.7.6 IND - Indirect Control File Processor

- Executes only as a BG job.
- Is not supported under BATCH.
- When obtaining information on nonexistent DX and DY device units, the .TESTDEVICE directive incorrectly returns OFL (off-line), rather than the correct NSD (no such device) in special symbol <EXSTRI>.
- Entering an invalid device or file name for the IND .TESTFILE directive returns only a syntax error and terminates IND. IND cannot determine the exact cause of the error.
- IND control files cannot include an indirect command file that calls another IND control file.
- Do not use CSI or DCL commands which require more than one line in IND control files. To execute multiline CSI commands and multiline DCL commands from an IND control file, create an indirect command file that contains the command and call the command file from the control file by using the $@ syntax.

  With single-line CSI commands, you can use CCL commands instead of creating and calling the indirect command file, which saves space in the IND control file.
- 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.

2.7.7 KED/KEX

- If you choose not to replace an existing journal (.JOU) file, you will receive an "*Unable to open journal file*" message.
- If there is sufficient memory on the system for KED or KEX to be loaded, but insufficient memory for it to run, you can get multiple and seemingly extraneous error messages as KED/KEX exit from memory. KED/KEX do not remove unprocessed elements from the command buffer when exiting and KMON treats the unprocessed elements as a KMON command and attempts to execute those elements. Typically, KMON displays an invalid command error. If the unprocessed command elements constitute a valid KMON command, KMON executes that command.

For example, assume you issue the command EDIT RESET to edit the file RESET.MAC. Assume there is sufficient memory for KED (in this case) to load but not run. After KED exits, KMON then treats the unprocessed command element RESET as a valid KMON command and executes it, initializing any background job tables.

- You must terminate (execute) all commands in LEARN mode by pressing the ENTER keypad key; do not use the RETURN key.

2.7.8 LD

- Do not rename the LD pseudohandler.
- When LD is generated for extended-unit support, logical disk unit numbers 40(octal) through 77(octal) are reserved for future use. Do not write any programs, command, or control files that use those logical disk unit numbers.
- When using nested logical disks, the primary disk must have a lower unit number than the nested disk. For example, if logical disk B is located within logical disk A, A must have a lower unit number than B. Logical disk A can be LD0 and logical disk B can be LD1, but not vice versa.

  Similarly, you cannot mount a logical device located within the same logical unit. For example, the following mount command attempts (but fails) to mount logical disk LD3:FOO.DSK that is located within logical unit LD3:

  ```
  MOUNT LD3: LD3:FOO.DSK
  ```

  However, the following command will work because it obeys the nested logical disk rule:

  ```
  MOUNT LD3: LD2:FOO.DSK
  ```

2.7.9 LIBR

- The LIBR /X option for creating multiple definition libraries is included in RT-11 only for compatibility with the RSTS operating system. It is recommended that you do not use the LIBR /X option under RT-11.

2.7.10 LINK

- Do not link programs using both the /R and /V:n options. Those options specify a REL format (foreground or system job environment) and an extended memory overlay structure. Such a program is not supported under RT-11 and can cause a system crash.
- The maximum address space of a program without virtual overlays is 65534_{10} bytes or 32767_{10} words, which is one word less than 32K words.

  The maximum virtual address space of a program with virtual overlays is 65472_{10} bytes or 32736_{10} words, which is 32_{10} words less than 32K words.

2.7.11 MACRO

- In MACRO-11 programs, all variable names of the form ...Vx are reserved. These names are used by macros in the system macro library (SYSMAC.SML).
- In MACRO-11 programs, all macro names of the form ...CMx are reserved. These names are used by macros in the system macro library (SYSMAC.SML).
- 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).

2.7.12 MDUP

- The distributed MDUP programs do not function with LSI–11 processors. (11/03 Quad and Dual Boards, LSI-11 and LSI-11/2) Use the procedure described in Appendix A of the *RT-11 Installation Guide* to build a monitor and an MDUP program that you can use with your LSI–11 processor.

2.7.13 QUEUE

- QUEUE protects files until it has placed them in the queue. A permanent feature of RT-11 is 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).

2.7.14 RESORC

- SHOW CONFIGURATION does not reliably display the existence of PMI memory. The reporting of PMI memory is determined by the placement of memory cards on the processor backplane and if the PMI memory is located on the backplane before Q-bus memory, PMI memory is not reported.
- 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.

- SHOW DEV: DU on SBC–11/21 processors reports the sysgened CSR address instead of the actual CSR address of 176150.

2.7.15 SETUP

- Set an LA210 or LA100 printer to AUTO mode before issuing it SETUP commands.
- Do not issue SETUP COLOR commands to monochrome monitors; such commands will overdrive your monitor, making the screen unreadable. Issue a SETUP RESET command to correct that condition. Do not include SETUP COLOR commands in command files that might be run on processors that have a monochrome monitor.

2.7.16 SIPP

- For RT-11 V5.7, SIPP does not understand the structure of I-D space programs and so you must use the SIPP /A option when patching I-D space programs. See the Extended Save Image File Format (.SAV) section in the *RT-11 Volume and File Formats Manual* for more information on the structure of I-D space programs. For more information pertaining to overlaid I-D space programs, see the *RT-11 System Utilities Manual, Part I*, Appendix A.
- When an overlaid program is specified to SIPP, SIPP reads in 259_{10} words of that program at a time. That is three words more than a single block—SIPP reads in the first three words of the next block too. If either block is a bad block, SIPP returns the error message, ?SIPP-F-Input error <dev:filenam.typ>. If you receive that error message, issue the command DIRECTORY/BAD for that device to find the bad block. SIPP reads in one block at a time when the program is overlaid and you use the /A option, or if the program is not overlaid.
- If SIPP is used to patch or examine an overlaid program without using the /A option and SIPP reads to the end of a disk, SIPP returns the error message, ?SIPP-F-Input error <dev:filenam.typ>, and does not read the last block on the disk.

2.7.17 SL

- The SET TT CRLF option causes the terminal driver to insert a CR LF pair whenever the number of characters displayed on a line exceeds the SET TT WIDTH=n value. However, escape sequences which are used for repositioning the cursor is not factored into the line count. Therefore, the CR LF may be inserted too early in the current line.

  If you wish to have the SET TT CRLF functionality, it is recommended the hardware setup features of the VTxxx family terminals be used. This can be done using the SETUP TERM WRAP command in RT-11 which will cause the terminal to track the position of the cursor and perform the equivalent of the CR LF pair at the proper positions.

- You should not include escape sequences in .GTLIN prompts and use SL in ON mode. Escape sequences embedded in .GTLIN prompts with SL set to ON cause cursor misplacement when the .GTLIN prompt is displayed.
- SL converts tab characters to blanks.
- You cannot use SL to edit the continuation line of a DCL KMON command.
- The single-line command editor (SL) cannot be used with foreground or system jobs.
- When using the single-line (SL) editor, if you SET TT CRLF you must also set your terminal to wrap. Set your terminal to wrap, using the SETUP TT WRAP command, or use the terminal hardware setup feature.
- The single-line editor (SL) does not support passing 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.
- Problems can occur with program prompting with the single-line command editor (SL) set ON. Although SL set ON prompts properly when used with programs that use .GTLIN for terminal input, SL may not prompt properly when used with various combinations of .PRINT and .TTYIN, or when prompts contain control characters such as line feeds or carriage returns. No correction is available for this problem; however, the recommended work-around is to SET SL KMON (rather than SET SL ON) or have your program set the EDIT$ bit (bit 4) in the JSW.
- SL and BATCH cannot be used together.

2.7.18 SPOOL

- You cannot run version (V2.1) of DECnet-RT and SPOOL at the same time. Together with probable low-memory constraints, a conflict occurs with the RMON fixed offset used by those two jobs.
- The transparent spooler for the FB monitor (SPOOL.REL) supports only a single device: SP0 or SP. Attempts to SET conditions for SPn, where n is other than 0 or omitted, produces an error message.
- You cannot use SPOOL (or SP handler) as the LOG device in BATCH under a mapped monitor.

2.7.19 TRANSFER

- TRANSF.EXE and the VMS Version 5.0 LAT were incompatible. Transferring files with this connection could cause premature LAT disconnection. A subsequent update to VMS Version 5.0 corrected the incompatibility and transfers are supported as documented.

- TRANSF.TSK (TRANSFER for RSX operating systems) requires the RMSRES and DAPRES libraries to be installed on the host PDP-11 processor.
- Run TRANSFER only from a host that you have logged onto using VTCOM. Attempting to run TRANSFER from a host you have logged onto using DECnet will hang the system. That is, do not use the SET HOST command to log onto a host and then attempt to run TRANSFER from that host (across the DECnet).

2.7.20 UCL

- If you define a UCL command as a save image (.SAV) file, the save image file returns an error message. KMON tries to execute the UCL command as a CCL command. If you define a UCL command as a valid DCL command, the keyboard monitor (KMON) returns an error message. KMON tries to execute the UCL command definition as a DCL command. See the *Introduction to RT-11* for information on defining your own commands.
- The CSI (command string interpreter) interface is not supported for UCL; do not issue the R UCL command.
- Only one command can be executed each time UCL is RUN, and all errors, including simple command line errors, are fatal and will return to the KMON prompt.

2.7.21 VBGEXE

- A completely virtual job (a job that is run under VBGEXE) has a fixed amount of kernel memory into which it can fetch handlers, create channels, and add additional queue elements. Kernel memory allocated on behalf of a completely virtual job cannot be reused during the life of that job. Thus, excessive use of .CDFN, .QSET, and .FETCH within a completely virtual job should be avoided.

  A common example of this is when MACRO is run as a completely virtual job in CSI prompting mode and several command lines are processed before exiting (or rerunning) MACRO. If these command lines contain references to devices whose handlers are not in memory, those handlers will be fetched and released multiple times. Since each fetch consumes kernel memory without allowing it to be reused when the handler is released, the job can run out of kernel memory and exit with an error. To avoid this problem, you should either load the handlers involved or limit the commands processed in any one invocation of a completely virtual program.
- You cannot run DUP with the /O option (BOOT command), IND, RESORC, or SPOOL from VBGEXE.
- The program cannot contain interrupt service routines.
- The program cannot use the following programmed requests: .FORK, .INTEN, .MFPS, and .MTPS.

- The program can use only the .PEEK, .POKE, .GVAL, and .PVAL programmed requests to access the kernel address space (for example, monitor data and the I/O page). An alternative is to map the KERNEL and/or IOPAGE regions to access kernel address space.
- Only a background program can issue a .FETCH request; VBGEXE fetches the handler into kernel memory below the US.
- VBGEXE places channels allocated by .CDFN in kernel memory below the USR and free of PAR1 space.
- VBGEXE places queue elements allocated by .QSET in kernel memory below the USR and free of PAR1 space.

2.7.22 VTCOM

- The distributed SB monitor does not contain timer support, therefore does not support VTCOM which requires monitor timer support. A SYSGEN including timer support must be performed to use VTCOM under the SB monitor.

2.8 Multiterminal Systems

The following restrictions apply to systems using the RT-11 V5.7 multiterminal special feature:

- Do not use ODT or the hard I/O versions of the DBG–11 pseudohandlers (SDH.SYS or SDHX.SYS) while running RT-11 under a multiterminal monitor that was generated with multiterminal timeout support. If you use them under such a multiterminal monitor, characters may be echoed twice on your console terminal screen. Characters can be echoed by both the debugging utilities and the monitor.

  Also, using ODT, SDH.SYS, or SDHX.SYS under that monitor causes characters you type as debugging commands to be placed in the KMON command buffer. When you exit from them, KMON attempts to execute those characters as a command, and KMON can return an 'invalid command' error message.

  Use instead VDT or the DBG–11 soft I/O pseudohandlers (SDS.SYS or SDSX.SYS) when running under a multiterminal monitor. (Also, you can run VDT, SDS.SYS, or SDSX.SYS from any terminal in a multiterminal configuration.)
- If you use VDT or the DBG–11 soft I/O version pseudohandlers (SDS.SYS or SDSX.SYS) with a multiterminal monitor, you must set your terminal to NOWRAP (SET TT NOWRAP).
- Support of the DL11–W interface requires a REV E or later module. Without a REV E module, ECO (Engineering Change Order) number DEC-O-LOG M7856-S0002 must be applied to the M7856 module.
- Support of the DLV11–J interface requires a REV E or later module. Without such a module, ECO M8043-MR002 must be applied to the M8043 module.

- The multiterminal handler can support remote terminals. Modem control is available for both DL11–E and DZ11 interfaces. The DL11 control answers ring interrupts, letting terminals dial in to the system. Dial-in is possible with the DZ11 interface, despite lack of a ring interrupt in the DZ11, if the modem is operated in auto-answer mode. Dial-in is achieved through a polling routine that periodically checks the status of each line on the multiplexer (see *RT-11 Device Handlers Manual*). Dial-up support for DZ interfaces requires BELL 103A-type modems with "common clear to send and carrier" jumpers installed. With this option installed, the modem operates in auto-answer mode.
- The hardware console interface must be a DL interface and it must be a local terminal. You can use the SET TT: CONSOL command to move the background console to any other local terminal in the system.
- The number of DL interfaces RT-11 supports, both local and remote, is limited to eight. This number includes the hardware console interface.
- The number of DZ11 controllers is limited to two, for a total of 16 lines. The total of DZV11 controllers is limited to four, for the same total of 16.
- The number of DHQ11, DHV11, or CXY08 controllers is limited to two, for a total of 16 lines. The total of DHU11, CXA16, CXB16, DHF11, and CXF32 controllers is limited to one, for a total of 16 lines (Note that although the DHF11 and CXF32 controllers have 32 lines, RT will only use a maximum of 16 of them).
- The maximum input data rate for a single terminal is 300 baud. The aggregate total input data rate for a system is 4800 baud.

  You can set the output baud rate to any speed; RT-11 sends output as fast as possible, depending on the capacity of the CPU and the nature of its load.
- If you plan to devote a serial line to the LS or XL handler, you can do so in RT–11 V5.6 through the SYSGEN dialog for a multiterminal system. In the new SYSGEN procedure, you can include the line and then use the new multi-terminal handler hooks support in the XL and LS handlers.
- Setting the baud rate, character length, number of stop bits, and parity by way of the .MTSET programmed request is supported only for DZ and DH interfaces.

3 Documentation Updates

This chapter describes the updates to the RT-11 Version 5.6 documentation.

3.1 RT-11 System Message Manual

This section contains an update to the System Message Manual

3.1.1 New Bootstrap Error Message

A new BSTRAP error message is:

```
?BOOT-U-Invalid processor for ZB/ZM monitor
```

This message is displayed if you try to boot a ZB/ZM monitor on a machine that does not support supervisor mode, separated I-D space. The crash code is octal 14.

If you get the preceding message, boot a non ZB/ZM monitor. For explanations of all other RT–11 system error messages, see the *RT–11 System Message Manual*.

3.2 RT-11 Commands Manual

This section contains updates to the RT-11 Commands Manual

3.2.1 Various

The COMPILE, DIBOL, and EXECUTE commands do not have a /BUFFERING option. The manual incorrectly lists this option.

3.2.2 DATE Command

The Date command has been modified to permit the entry of 4 digit years.

The Date command is of the form

DATE dd-mmm-yy

or

DATE dd-mmm-yyyy

where yy is a value 72 through 99 representing 1972 through 1999

where yyyy is a value representing the years 1972 through 2099

Note that the 2 digit values 00 through 72 are invalid, as are the 4 digit values less than or equal to 1971 or vales 2100 and greater.

3.3 System Macro Library Manual

Page 2-246 .SFPA

The description for the *addr* argument incorrectly refers to Data Space; *addr* always refers to Instruction Space Add a sixth and seventh bullet under *Notes*

- The monitor maintains two separate FPU exception routines for each job under a fully mapped monitor (ZB or ZM) - - one is for floating point exceptions taken in User Mode and the other is for those take in Supervisor Mode. When *addr* is even, .SFPA modifies the User Mode exception routine address. When *addr* is odd, .SFPA modifies the Supervisor mode exception routine address. There are three values of *addr* that are treated specifically:

#0	disables User Mode floating-point exception routines
#1	disables User Mode floating-point exception routines, but enables context switching of the FP registers under multi-job monitors. Under single job monitors #1 is treated identically to #0
#3	disables Supervisor mode floating-point exception routines

- Under a fully mapped monitor (ZB or ZM), when a user routine is activated, the monitor only disables the trap for the mode taken. The other mode's user trap routine is unmodified.

Page 2-271, .TRPSET The description for the *addr* argument incorrectly refers to Data Space; *addr* always refers to Instruction Space Add the following after the first paragraph under *Notes*

- Under the fully mapped monitors (ZM and ZB), the monitor maintains two separate user trap routines for each job - - one is for traps taken in User Mode and the other is for those take in Supervisor Mode. When *addr* is even, .TRPSET modifies the User Mode trap routine address. When *addr* is odd, .SFPA modifies the Supervisor mode trap routine address. There are two values of *addr* that are treated specifically:

#0	disables User Mode trap routines
#1	disables Supervisor mode trap routines

- Under a fully mapped monitor (ZB or ZM), when a user routine is activated, the monitor only disables the trap for the mode taken. The other mode's user trap routine is unmodified.

3.4 System Subroutine Library Manual

Replaced Examples

The following examples replace those found in the *RT–11 System Subroutine Library Manual* in the indicated sections.

Section 1.7.5 Character String Operations

```
        .TITLE  FGET2;2         ;calling SYSLIB from MACRO
        .GLOBL  CONCAT  CALL$F  ;SYSLIB routines
        .MCALL  .PRINT  .EXIT   ;macros
START:: MOV     #STRCON,R2      ;Point to final buffer
        MOV     #ARGBLK,R5      ;Point to argument block
        MOV     #CONCAT,R0      ;Point to routine to call
        CALL    CALL$F          ;Call it, saving registers R1-R4
        .PRINT  R2              ;Print resulting string
        .EXIT                   ;and away
ARGBLK: .WORD   3               ;3 arguments
        .WORD   STRNG1          ;first input string address
        .WORD   STRNG2          ;second input string address
        .WORD   STRCON          ;output string buffer address
STRNG1: .ASCIZ  "Research and"
STRNG2: .ASCIZ  " Development"
STRCON: .BLKB   31
        .END    START
```

Section 1.8.2 Passing String to Subprograms

```
        Byte R(21)      !20 char string variable
        Call Subr (R)
```

ABTIO

```
        Program FABTIO!2 !demo ABTIO
C
C       Pump out 9 buffers to LP using non-wait mode I/O.
C       Abort the I/O.  This should cause the printout to
C       be truncated.
C       NOTE: using LS7 as a trick to bypass SPOOLING, since
C       spooling is normally applied to LP, LP0, LS and LS0.
C
        Integer*2 DBLK(4)
        Data DBLK /3rLS7, 3rTES, 3rTXX, 3rTMP/ !LS7 to sneak around SP
        Integer*2 BUFFER(256,9), CHARS, CRLF
        CHARS = '11'                    !begin at 1
        CRLF = '015'o + ('012'o * '400'o) ! CR / LF pair in word
        ICHAN = IGETC ()
        Call IQSET (20)                 !get more queue elements
        Call ENTER (ICHAN, DBLK, 0)
        Do 300, J = 1, 9
            Do 200, I = 1, 256
                BUFFER(I,J) = CHARS
                If (IMOD (I, 60 * 2) .eq. 0) BUFFER(I,J) = CRLF
 200            Continue
            CHARS = CHARS + '400'o + 1  !then 2 ... 9
            Call WRITE (256, BUFFER(1,J), J - 1, ICHAN)
300         Continue
C
C       Comment out the call to ABTIO and observe the difference
C
        Call ABTIO (ICHAN)              !stop it in midstream
        Call CLOSEC (ICHAN)
        End
```

CALL$F

```
        .TITLE  FCALLF;2
        .GLOBL  GTLIN, CALL$F   ;routines from SYSLIB

        .PSECT  CODE,I          ;program code fragment

        MOV     #GTLIN,R0       ;routine to call (ultimately)
        MOV     #PARMS,R5       ;argument list to use
        CALL    CALL$F          ;call GTLIN, save R1-R4

        .PSECT  DATA,D          ;program data fragment

PARMS:  .WORD   2               ;number of arguments
        .WORD   BUF             ;response buffer
        .WORD   PROMPT          ;prompt string

BUF:    .BLKB   81.             ;buffer
PROMPT: .ASCII  "Enter username>" ;prompt
        .BYTE   200             ;without terminating cr/lf
```

CLOSZ

```
        Program FCLOSZ!2 !demo  ICLOSZ
C
C       Create DK.TEST.TMP with 100 blocks, and
C       make permanent at that size w/o I/O.
C
        Integer*2 DBLK(4)
        Data DBLK /3rDK , 3rTES, 3rT  , 3rTMP/
        Parameter SUCCS = '001'o
        Parameter FATAL = '010'o
C
        ICHAN = IGETC()
        If (ICHAN .lt. 0) Go To 100
        ISIZE = IENTER (ICHAN, DBLK, 100)
        IF (ISIZE .lt. 0) Go To (110, 120, 130) IABS(ISIZE)
        ISIZE = ICLOSZ (ICHAN, ISIZE)
        If (ISIZE .eq. -4) Go To 200
        Call IFREEC(ICHAN)
        Call Exit (SUCCS)
100     Type *, ' ?FCLOSZ-F-No channel available'
        Call Exit (FATAL)
110     Type *, ' ?FCLOSZ-F-Channel in use'
        Call Exit (FATAL)
120     Type *, ' ?FCLOSZ-F-Not enough room'
        Call Exit (FATAL)
130     Type *, ' ?FCLOSZ-F-Device in use'
        Call Exit (FATAL)
200     Type *, ' ?FCLOSZ-F-Protected file already exists'
        Call Exit (FATAL)
        End
```

CRAW

```
        Program FPLAS!2 !demo PLAS requests
C
C       This program has two behaviors depending on whether or not
C       the global region TSTREG exists.
C       If it does not exist, it creates it, get a line from the
C       terminal and stores it in the region.
C       If it does exist, it prints the line stored in the region
C       then eliminates the region.
C       In both cases it displays the mapping context of the region.
C
        Include 'SRC:RDBDF'     !RDB definitions
        Include 'SRC:WDBDF'     !WDB definitions
        Parameter SUCCS = '001'o, FATAL = '010'o
        Parameter BASADR = '160000'o !PAR 7 for gbl region
        Parameter REGSIZ = (81 + 63) / 64 !size in chunks
        Integer*2 REGNAM(0:1)   !global region name
        Data REGNAM /3rTST, 3rREG/
        Integer*2 WDB (0:WNLGH) !WDB block
        Integer*2 RDB (0:RGLGH) !RDB block
        Character*7 ERRCAL      !code for error call
        Integer*2 AREA(0:1)     !must disable subscript checking
        Integer*2 AREA0         !addr of AREA(0)
        Integer PAR7            !subscript for AREA
C
C       Find a way of referencing address 160000
C
        AREA0 = IADDR (AREA(0)) !find addr of AREA(0)
        PAR7 = ('160000'o - AREA0) / 2 !find "element" of AREA
C                               !that is at 160000
C
C       Create (or attach) the global regions
C
        RDB(RGSIZ) = REGSIZ     !region size
        RDB(RGSTS) = RSGBL + RSCGR !create or attach global
        RDB(RGNAM+0) = REGNAM(0)!region name
        RDB(RGNAM+1) = REGNAM(1)!region name
        IERR = ICRRG (RDB)      !can we find it?
        ERRCAL = 'CRRG'
        If (IERR .ne. 0) Go To 1000 !error
C
C       Create an address window
C
        WDB(WNAPR) = (BASADR / '20000'o) * '400'o
C                               !Put PAR number in high byte
        WDB(WNSIZ) = REGSIZ     !region size
        WDB(WNRID) = RDB(RGID)  !region ID
        WDB(WNOFF) = 0          !offset 0
        WDB(WNLEN) = 0          !full size
        WDB(WNSTS) = 0          !take all defaults
        IERR = ICRAW (WDB)      !create a window
        ERRCAL = 'CRAW'
        If (IERR .ne. 0) Go To 1000 !error
C
C       Map to it (could be done by ICRAW)
C
        IERR = MAP (WDB)        !map into it
        ERRCAL = 'MAP'
        If (IERR .ne. 0) Go To 1000 !error
C
C       display mapping context
C
        IERR = IGMCX (WDB)      !return mapping context
        ERRCAL = 'GMCX'
        If (IERR .ne. 0) Go To 1000 !error
```

```
        Type 1, '    Window ID = ', IAND (WDB(WNID), '377'o),
        1       '   Window APR = ', WDB(WNAPR) / '400'o,
        2       '  Window Addr = ', WDB(WNBAS),
        3       '  Window Size = ', WDB(WNSIZ),
        4       '  Window RgID = ', WDB(WNRID),
        5       'Window Offset = ', WDB(WNOFF),
        6       'Window Length = ', WDB(WNLEN),
        7       'Window Status = ', WDB(WNSTS)
1       Format (8(' ', a16, o7/))
C
C       Decide if this is the first or second run
C
        If (IAND (RDB(RGSTS), RSNEW) .eq. 0) Go To 100
C       else first pass
C
C       Collect a line, put it in the global region and exit
C       leaving the region for the next run of this program
C
        Call GTLIN (AREA(PAR7),,'p') !get a string and put in region
        Call Print ('!FPLAS-I-Pass 1 success')
        Call Exit (SUCCS)
C
C       second pass
C
C       Get the line from the region and display it,
C       then eliminate the region.
C
100     Continue
        Call PRINT (AREA(PAR7)) !print the string in the region
C
C       Unmap the window
C
        IERR = IUNMAP (WDB)
        ERRCAL = 'UNMAP'
        If (IERR .ne. 0) Go To 1000 !error
C
C       Delete the window
C
        IERR = IELAW (WDB)
        ERRCAL = 'ELAW'
        If (IERR .ne. 0) Go To 1000 !error
C
C       Eliminate the region
C
        RDB(RGSTS) = RSEGR      !eliminate region
        IERR = IELRG (RDB)
        ERRCAL = 'ELRG'
        If (IERR .ne. 0) Go To 1000 !error
        Call Print ('!FPLAS-I-Pass 2 success')
        Call Exit (SUCCS)
C
C       Error processing
C
1000    Continue
        Type *, '?FPLAS-F-', ERRCAL, 'Failed with code', IERR
        Call Exit (FATAL)
        End
```

DATE

```
        Program FDATE!2 !demo DATE & IWEEKD
C
C       Display current date and day of week
C
        Byte DAYSTR(9)
        Integer*4 DAYS(7)
        Data Days /'Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Dat' /
C
        Call IDATE (MONTH, IDAY, IYEAR)
        IWD = IWEEKD (MONTH, IDAY, IYEAR)
        Call DATE (DAYSTR)
        Type 100, DAYS(IWD), DAYSTR
100     Format (' ', 'Today''s date: ', 6x, a4, 1x, 9a1)
        End
```

GICLOS/GIOPEN/GIREAD/GIWRIT

```
        Program FGIDIS!2 !demo GIDIS interface routines
C
C       Declare storage
C
        Integer*2 BUFLEN, LUN, MSGLEN, OCLEN, OPCODE
        Integer*2 BUFFER(3), MESSAG(1), STATUS(2)
C
C       user program here
C             . . .
C
        LUN = 5         !assign logical unit number
        OPCODE = 55*256 !request current position
        OCLEN = 0       !opcode length is 0
C
C       Put OPCODE and OCLEN into MESSAG buffer
C
        MESSAG(1) = OPCODE + OCLEN
        MSGLEN = 1      !length of message
C
C       Send to GIDIS
C
        Call GIWRIT (STATUS, LUN, MESSAG, MSGLEN)
        If (STATUS(1) .le. 0) Go To 999 ! error
C
        BUFLEN = 3      !length of report
C
C       Get report from GIDIS
C
        Call GIREAD (STATUS, LUN, BUFFER, BUFLEN)
        If (STATUS(1) .le. 0) Go To 999 ! error
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
C       BUFFER(3) = Current 'Y' position
C
C       more user program
C             . . .
C
999     Continue                !diagnose errors here
        End
```

GTDIR

```
        Program FGTDIR!2 !demo GTDIR and GTENT
C
C       Display file(s) on SY:
C
        Parameter ERROR= '010'o
        Integer*2 WKAREA(64)    !area for IGTDIR/IGTENT
        Integer*2 ENTRY(7)      !single directory entry
        Integer*2 BUFFER(512)   !directory buffer
        Integer*2 DBLK(4)       !Device/file to search
        Integer*2 BLOCK2
        Integer*4 BLOCK         !really unsigned 16 bit
        Byte NAME(11)           !file name
        Byte DATSTR(11)         !date in ascii
        Byte STRING(81)         !filespec(s) w/o device
        Byte Prompt(8)          !command prompt
        Data Prompt /'F', 'i', 'l', 'e', 's', '?', ' ', '200'o/
        Data DBLK /3rSY , 3*0/  !system device
        Data BLOCK /0/
        Equivalence (BLOCK2, BLOCK)
C
        ICHAN = IGETC ()
1000        Continue
            Call RCTRLO         !reset ^O
            Type *, ' '         !blank line
            Call GTLIN (STRING, PROMPT) !get filespec(s)
            IERR = IGTDIR (64, WKAREA, ICHAN, BUFFER, , DBLK, STRING)
            If (IERR .ne. 0) Go To 2000
1100            IERR = IGTENT (WKAREA, ENTRY, , BLOCK2, NAME)
                If (IERR .lt. 0) Go To 1000
                Call DATE4Y (DATSTR, ENTRY(7))
                Type 1, (NAME(I), I = 1, 10), ENTRY(5), DATSTR, BLOCK
1               Format (' ', 10a1, ' ', i6, ' ', 11a1, ' ', i6)
                Go To 1100
2000    Call Print ('?FGTDIR-F-GTDIR failed')
        Call Exit (FATAL)
        End
```

HERR

```
        Program FHERR!2 !demo HERR and SERR
C
C       Demonstrate how to save, modify and restore the
C       HERR/SERR status in a subroutine
C
        Parameter SUCCS = '001'o, FATAL = '010'o
        Integer*2 DBLK(4)       !unknown device/file
        Integer*2 CHAN          !channel to use
        Integer*2 OLD1          !mainline SERR/HERR setting
        Data DBLK /3rZZZ, 3*3rYYY/ !non-existent device
C
        CHAN = IGETC ()
        Call IHERR ()           !force HERR status
        OLD1 = IHERR ()         !confirm it
        Call TEST (CHAN, DBLK)  !call routine that uses SERR
        If (OLD1 .ne. IHERR ()) Then
            Type *, '?FHERR-F-S/HERR status not saved'
            Call Exit (FATAL)
        Else
            Type *, '!FHERR-I-Success'
        End If
        Call ISERR ()           !force SERR status
        OLD1 = ISERR ()         !confirm it
        Call TEST (CHAN, DBLK)  !call routine that uses SERR
        If (OLD1 .ne. IHERR ()) Then
```

```
            Type *, '?FHERR-F-S/HERR status not saved'
            Call Exit (FATAL)
        Else
            Type *, '!FHERR-I-Success'
            Call Exit (SUCCS)
        End If
        End

        Subroutine TEST (CHAN, DBLK)
        Integer*2 DBLK(4)       !unknown device/file
        Integer*2 CHAN          !channel to use
        Integer*2 ERROR         !error code
        Integer*2 OLDERR        !previous SERR/HERR setting
C
        OLDERR = ISERR ()       !set SERR, save old setting
        ERROR = LOOKUP (CHAN , DBLK) !open that WILL fail
        Type *, 'LOOKUP returned', ERROR
        If (OLDERR .eq. 0) CALL HERR !restore setting
        RETURN
        END
```

IADDR

```
        Program FADDR!2 !demo IADDR function
C
C       This is example code of an actual use of IADDR.
C       Mapping requests often need addresses on PAR boundaries
C       and this code demonstrates how you can create such an address
C       dynamically.
C
        Implicit Integer*2 (A-Z)
        Parameter SUCCS = '001'o
        Parameter FATAL = '010'o
        Integer*2 AREA(0:1)     !must disable subscript checking
C
        AREA0 = IADDR (AREA(0)) !find addr of AREA(0)
        PAR1 = ('20000'o - AREA0) / 2 !find element of AREA
C                               ! that is base of Par1
        AREAN = IADDR (AREA(PAR1)) !verify it
        Type 100, AREA0, PAR1, AREAN
100     Format (' ', 'Base address of AREA() is ', o8 /
        1       ' ', 'Address of AREA(', i5, ') is ' o8)
        If (AREAN .ne. '20000'o) Go To 1000
        Type *, '!FADDR-I-Success'
        Call Exit (SUCCS)
C
1000    Continue
        Type *, '?FADDR-F-Failed'
        Call Exit (FATAL)
        End
```

JAFIX

```
        Program FJAFIX!2 !FORTRAN IV
C
C       Demo Real*4 to Integer*4 conversion
C
        Real*4 RTRY             !test value
        Real*4 RPTRY            !previous test value
        Real*4 RNEW             !reconverted value
        Integer*4 JTRY          !integer equivalent
        Integer*4 JA, JB, JC    !variables
        Logical*1 ASCII(9)      !variables
        Data ASCII(9) /0/       !terminate string with null
C
        RTRY = 1.               !start at the beginning
        RPTRY = RTRY            !remember last value
100     Continue
            IERR = JAFIX (RTRY, JTRY) ! convert to I*4
            If (IERR .eq. -2) Type *, '?FJAFIX-W-Overflow', RTRY
            RNEW = AJFLT (JTRY) !convert back
            If (RNEW .ne. RTRY) Go To 200 !lost some bits
            RPTRY = RTRY
            RTRY = RTRY * 2.
            Go To 100
C
200     Continue
        Type *, RTRY, RNEW
        End
```

JDFIX

```
        Program FJDFIX!2 !FORTRAN IV
C
C       Demo Real*4 to Integer*4 conversion
C
        Real*8 RTRY             !test value
        Real*8 RPTRY            !previous test value
        Real*8 RNEW             !reconverted value
        Integer*4 JTRY          !integer equivalent
        Integer*4 JA, JB, JC    !variables
        Logical*1 ASCII(9)      !variables
        Data ASCII(9) /0/       !terminate string with null
C
        RTRY = 1.               !start at the beginning
        RPTRY = RTRY            !remember last value
100     Continue
            IERR = JDFIX (RTRY, JTRY) ! convert to I*4
            If (IERR .eq. -2) Type *, '?FJDFIX-W-Overflow', RTRY
            RNEW = AJFLT (JTRY) !convert back
            If (RNEW .ne. RTRY) Go To 200 !lost some bits
            RPTRY = RTRY
            RTRY = RTRY * 2.
            Go To 100
C
200     Continue
        Type *, RTRY, RNEW
        End
```

JSUB

```
        Program FJSUB!2 !FORTRAN IV
C
C       using JJCVT and TIMASC to display results
C
        Integer*4 HOUR1         !value of 1 hour
        Integer*4 HOUR12        !value of 12 hours
        Integer*4 JA, JB, JC    !variables
        Logical*1 ASCII(9)      !variables
        Data ASCII(9) /0/       !terminate string with null
C
C       init "constants"
C
        Call JTIME (1, 0, 0, 0, HOUR1)
        Call JTIME (12, 0, 0, 0, HOUR12)
C
C       convert from RT-11 time format to I*4 format
C
        Call JJCVT (HOUR1)
        Call JJCVT (HOUR12)
        Call JMOV (HOUR1, JA)   !JA = 1hr
        Call JMOV (HOUR12, JB)  !JB = 12hr
        Call JSUB (JB, JA, JC)  !JC = JB - JA
        Call JJCVT (JC)         !back to time format
        Call TIMASC (JC, ASCII) !display results
        Call PRINT (ASCII)      !...
        End
```

KPEEK

```
        Program FKPEEK!2
C
C       Demo program to show using KPEEK and KPOKE accessing the PS
C       This is used to illustrate safely modifying shared memory by
C       temporarily raising processor priority to 7.
C
C       WARNING: invoking KPEEK uses an EMT which is executed at PR0!
C
        Integer*2 PS, PR7       !Proc status register, priority 7 mask
        Data PS /"177776/, PR7 /"000340/
        Integer*2 OLDPS, NEWPS  !copies of PS
        Integer*2 SHARED        !location shared with a
        Common /STATUS/ SHARED  !  completion routine
        Integer*2 OLDSHR        !local copy of old value
C
        OLDPS = KPOKE (PS, PR7, 'BIS') !Set PS to PR7 (BIS)
        NEWPS = KPEEK (PS)      !get modified PS
        OLDSHR = SHARED         !get current value
        SHARED = SHARED + 3     !change shared location, protected
C                               !  with PR7 from interference
        CALL KPOKE (PS, OLDPS, 'MOV') !Put old PS back
        Type 100, OLDPS, NEWPS, KPEEK (PS) !display PSs
100     Format (' ', 3o10)
        End
```

MRKT

```
        .TITLE  FAMRKT;2  macro completion routine for FMRKT
        .MCALL  .RSUM   .PRINT  .MRKT

FAMRKT::
        MOV     R0,ID           ;reissue with same ID
        .PRINT  #HERE           ;indicate entered
        INC COUNT               ;indicate entered
        .RSUM                   ;resume suspended mainline
        .MRKT   #AREA,#TIME,#FAMRKT ;reissue the request
        RETURN

AREA:   .BLKW   4
ID      =:      AREA+6          ;ID word in AREA

HERE:   .ASCIZ  "... in FAMRKT"

.PSECT  TIMING,RW,D,GBL,REL,OVR ;Common /TIMING/ ...
COUNT:  .BLKW   1
TIME:   .BLKW   2
        .END

        Program FMRKT!2 ! Demo MRKT and CMKT
C
C       Run two timers at 5 and 50 ticks each, shutdown
C       first timer after 5 times and wait for second.
C
        Integer*2 COUNT
        Integer*4 TIME, TIME2, RTIME
        Common /TIMING/ COUNT, TIME
        External FAMRKT         !macro completion routine
C
        Call IQSET (10)         !get some queue elements
        COUNT = 1               !init count
        TIME = 5                !init at 5 ticks
        Call JJCVT (TIME)       !and convert from int to time
        TIME2 = 50              !init at 50 ticks
        Call JJCVT (TIME2)      !and convert from int to time
        Call MRKT (12345, FAMRKT, TIME) !timer 1
        Call MRKT (02222, FAMRKT, TIME2)!timer 2
100     Continue
        Call SUSPND             !wait for completion
        If (COUNT .eq. 5) Then
            Call CMKT (12345, RTIME) !on 5th, shutdown timer 1
            Go To 200
        Else
            Go To 100
        End If
C
200     Continue
        Call SUSPND             !wait for final completion
        Call PRINT ('!FMRKT-I-Normal Termination')
        End
```

RAD50

```
        Program FRAD50 !demo RAD50 routine
        Real*8 FSPEC
        Real*4 FSPECR(2)
C
        FSPECR(1) = RAD50 ('SY SWA')
        FSPECR(2) = RAD50 ('P  SYS')
C
C       The equivalent for IRAD50 is
C
        Call IRAD50 (12, 'SY SWAP  SYS', FSPEC)
        End
```

READF

```
        Program FREADF!2
C
C       demonstrate READF and WRITF routines
C
        Integer*2 ICHAN, OCHAN          !channel numbers
        Integer*2 BLK                   !current block number
        Integer*2 SIZE                  !file size (hi BLK+1)
        Parameter WCNT = 256            !word cound
        Integer*2 BUFFER(WCNT)          !buffer
        Integer*2 ERROR                 !error indicator
        Common /JFWCCW/ ICHAN, OCHAN, BLK, SIZE, BUFFER, ERROR
        Integer*2 DBLK1(4), DBLK2(4)    !file names
        Data DBLK1 /3rSY , 3rRT1, 3r1XM, 3rSYS/
        Data DBLK2 /3rDK , 3rRT1, 3r1XM, 3rTMP/
C
        ICHAN = IGETC()
        OCHAN = IGETC()
        SIZE = LOOKUP (ICHAN, DBLK1)    !open input
        If (SIZE .lt. 0) Go To 1000
        IERR = IENTER (OCHAN, DBLK2, SIZE) !open output
        If (IERR .lt. 0) Go To 1100
        SIZE = SIZE - 1                 !highest block number
        BLK = -1                        !since we preincrement in FREADG
        Call FREADG (0)                 !start the I/O
C       here we could do other stuff while I/O is happening
        Call SUSPND                     !wait for I/O to finish
        If (ERROR .eq. 0) Go To 900     !success
        Type *, '?FREADF-F-A completion routine reported code = ', ERROR
        Call EXIT (FATAL)
C
900     Type *, '!FREADF-I-Success'
        Call EXIT (SUCCS)
1000    Type *, '?FREADF-F-LOOKUP failed, code = ', SIZE
        Call EXIT (FATAL)
1100    Type *, '?FREADF-F-ENTER failed, code = ', IERR
        Call EXIT (FATAL)
        End
```

SDAT

```
        Program FSDAT!2 !demo SDAT (use with FRCVDx)
C
C       Try to send a message to the other job
C       send a null message for EOF and exit
C
        Parameter SUCCS = '001'o, FATAL = '010'o
        Parameter WCNT = 42
        Integer*2 SNDCNT
        Integer*2 BUFFER(WCNT)
        Integer*2 CTRLZ
        Data CTRLZ /'032'o/             !^Z null string equiv
        External LEN                    !use RT-11 SYSLIB LEN func
C
100     Continue
            Call GTLIN (BUFFER)         !get a line
            SNDCNT = LEN (BUFFER)       !get byte count (-1)
            If (SNDCNT .eq. 0) Then     !is it a null string
                BUFFER(1) = CTRLZ       !yes, send ^Z
                SNDCNT = 1              !...
            End If
            SNDCNT = (SNDCNT + 2) / 2   ! get word count
200             Continue
                IERR = ISDAT (BUFFER, SNDCNT) !try sending a message
                If (IERR .eq. 1) Go To 200 !wait for the job
            If (IERR .ne. 0) Go To 300  !unknown error
            Call MWAIT          !wait for a message to be received
            If (BUFFER(1) .ne. CTRLZ) Go To 100
        Call PRINT ('!FSDAT-I-Normal termination')
        Call EXIT (SUCCS)               !and done
C
300     Call PRINT ('?FSDAT-F-Unknown error code from ISDAT')
        Call EXIT (FATAL)
        End
```

TIMEA

```
        Subroutine FTIMEA (ID) !2
C
C       demonstrate the FORTRAN timer completion routine
C
        Integer*2 HRS, MIN, SEC, TIC
        Integer*2 AREAA(4)
        External FTIMEC         !fast timer completion
C
        Call TIMER (0, 0, 1, 0, AREAA, ID, FTIMEC)
        Call PRINT ('!FTIMEA-I-Entered')
        End
```

TRIM

```
        Program FTRIM!2
C
C       This demonstrates the TRIM function
C
        External LEN            !use the RT-11 LEN function
                                !not the F77 built-in LEN
        Byte INPUT(81)          !input buffer
C
        Accept 100, (INPUT(I), I=1,80)
100     Format (80a1)
        Call SCOPY (INPUT, INPUT, 80)   !punch in a null
        Type *, LEN (INPUT)             !length before trimming
        Call TRIM (INPUT)               !trim trailing blanks
        Type *, LEN (INPUT)             !length after trimming
        End
```

New Information

The following system subroutine is not in the *RT-11 System Subroutine Library Manual* but is in SYSLIB. See the *RT-11 System Subroutine Library Manual* for all other subroutines.

MGETR (Requires Mapped Monitor)

The MGETR subroutine attached to a specified global region (or creates a local region). MGETR can optionally initialize a region by reading a portion of a file into the region or by calling a specified subroutine.

MGETR does not fetch handlers. Any handler required by I/O in MGETR must be loaded or fetched by the program. MGETR is a replacement for IGETR which is retained for backwards compatibility only.

In the following, "chunk" is defined as 64-byte units.

```
I = MGETR
   -or-
CALL MGETR

     (WORK,WRKSIZ,{'Private'},[NAME],ADDR[,MEMADR][,CSIZE][,MSIZE][,OFFSET]
    {'Shared' }
    {'Age'    }
    {'Local'  }

          [{,CHAN[,BLK]]}][{,'UI'}][{,'ReadOnly'}][{,  'Bypass']})
          [{,FILE[,BLK]]}][{,'UD'}][{,'Writable'}][{,'NoBypass']})
          [{,SBRTN,-1  ]}][{,'SI'}]
                          [{,'SD'}]
                          [{,'CI'}]
                          [{,'CD'}]
```

where:

work	is a 15_{10}-word work area array. The first five words contain the region definition block (RDB) which has the following fields: • A unique region identifier (R.GID) • The size of the region (R.GSIZ) • The region status word (R.GSTS) • The region name in two RAD50 words (R.NAME+0 and +2) The *work* array specified in the MGETR call is also specified in a corresponding IFREER call to remove the region. The contents of the rest of the work area array is reserved.
wrksiz	is an integer constant containing the number of words in the *work* area. A value of 15(10) or greater is valid.
char	is a character constant specifying the region attributes. Only the first letter of the constant need be specified, and it must be enclosed in single quotes (’). Specify one of the following: ’private’ — Program sole owner of the global region ’shared’ — Global region available to other programs ’age’ — Enable automatic global elimination (implies ’shared’) ’local’ — Local region, not global.
name	is the two word name of the global region, expressed in RAD50. This argument is not used if ’local’ has been specified, but the position of the argument must be accounted for: `CALL MGETR (WORK, 15, ’local’, , addr ...)`
addr	is an array which is mapped to the region’s base address. It must be on a PAR boundary (a number evenly divisible by 20000_8). See the IADDR example for a technique to do this.
memadr	is the physical memory address for the base of a newly created region. This value is expressed in chunks. If omitted, the monitor uses a physical address of its own choosing.
csize	is the size of the region expressed in words. If the region exists *csize* may be omitted or have the value 0. If the region does not exist, *csize* must be specified since that information is required for creation.
msize	is the size of the region to map, expressed in words. If you specify *msize* as 0 or omit it, the entire region will be mapped.
offset	is the offset from the beginning of the region, expressed in chunks. The *offset* is the number of chunks to skip in the region before beginning mapping to your job. If *offset* is specified as 0 or omitted, you begin mapping at the beginning of the region. The next argument position can have three interpretations, if the value of *blk* is -1, this argument is *sbrtn*. If the value of the argument is between 0 and 255(10) and *blk* is not -1, it is chan. If the value of the argument is greater than 255_{10} and *blk* is not -1, it is file.
chan	is a channel opened (by the calling program) on a file from which to read initialization data.

`file` is a pointer to a four-word block containing a RAD50 device and file descriptor (DBLK). This file is one from which to read initialization data.

`sbrtn` is the name of a subroutine that is to be called to initialize the region. *sbrtn* must be declared in an EXTERNAL statement. When a new region has been created, the following call is performed:

```
CALL sbrtn (addr, msize)
```

`blk` is the first block in the referenced file to use if *chan* or *file* is used or a *constant* -1 if sbrtn is used. If *blk* is omitted, it is treated as though it were specified as 0.

`char` is a character constant specifying the mapping attributes. Both letters of the string need to be specified, and it must be enclosed in single quotes ('). Specify one of the following:

'UI' — user instruction space
'UD' — user data space
'SI' — supervisor instruction space
'SD' — supervisor data space
'CI' — current instruction space
'CD' — current data space

If omitted, defaults to an old style (pre V5.6) mapping request.

`char` is a character constant specifying the region read/write attributes. Only the first letter of the constant need be specified, and it must be enclosed in single quotes ('). Specify one of the following:

'ReadOnly' — references to the region must be reads
'Writable' — references may be reads or writes

If omitted, defaults to 'Writable'.

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

'Bypass' — bypass hardware memory caching
'NoBypass' — do not bypass hardware memory caching

If omitted, defaults to 'NoBypass'.

MGETR can be called from MACRO-11 programs using the standard FORTRAN calling convention. All registers contents are destroyed across the call (see CALL$F). MGETR calls IGETC and IFREEC, which are FORTRAN dependent routines.

To use MGETR in a MACRO-only program, use the following IGETC and IFREEC substitute routines:

```
IGETC::
 MOV #channo,R0 ;supply available chan number
IFREEC::   ;ignore return of channel
 RETURN   ;and done
```

Errors:

```
-1 Invalid ADDR alignment
-2 No WDB for .CRAW
-3 Any .CRAW error except No WDB
-4 EOF on .READW
-5 I/O Error on .READW
-6 Channel closed when .READW attempted
-6 No Chan available from IGETC
-7 No RCB for .CRRG
-8 Insufficient memory for .CRRG
-9 <RESERVED>
-10 Memory too fragmented to return at .ELRG
-11 Global region not found (and no non-zero SIZE specified)
-12 No GRCB 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 sharable
-18 Any .ELRG error except Mem Frag
-19 Unknown char argument value
-20 Required argument is missing
-21 Work area too small
```

3.5 RT-11 Utilities Manual, Part I

Appendix A of the *RT–11 Utilities Manual, Part I* is incorrect in saying that SHANDL contains the source code for all the overlay handlers. UHANDL is provided on the RT–11 distribution source kit and contains the source code for the OHANDL, VHANDL, XHANDL, and ZHANDL overlay handlers. Conditionals in UHANDL build the different variants of the overlay handler with the exception of SHANDL. SHANDL is built from the SHANDL source file.

3.6 Device Handlers Manual

The RT-11 Version 5.7 Ethernet handlers (NI, NQ, NC, NU) now have built in support for some .SPFUN functions for generalized Ethernet applications including, in particular, TCP/IP implementations.

.SPFUN 206 enables / disables internal device handler queueing of received pachets (protocols enabled with .SPFUN 202).

.SPFUN 207 returns the 6 byte physical station address of the network card without requiring a unit allocation.

.SPFUN 210 returns internal handler status information containing total receive interrupts, total transmitter interrupts, total receive timeouts, total transmitter timeouts, number of received packets for each logical unit, and number of transmitted packets for each logical unit.

3.6.1 .SPFUN 206 Enable Frame Queueing

The Enable Frame Queueing special function allows the Ethernet handler to internally queue received packets that match any enabled address / protocol even though a reveive request has not been issued for that address / protocol or a previous packet has not been processed. The NQ / NU handlers allocate space for internally queueing up to 6 packets. The NC handle allocates space for internally queueing up to 10 packets.

The following is the form of the special function Enable Frame Queueing (.SPFUN 206)

```
Macro Call:

 .SPFUN area, chan, func, buf, wcnt, blkn, [crtn]

where:

  area is the address of a 6 word EMT argument block

  chan is a channel number in the range 0 to 376 (octal)

  func is 206

  buf is the address of a 1 word buffer for the
   returned status word which is always 0

  wcnt is 0

  blkn is 1

  crtn is the entry point of a completion routine
```

3.6.2 .SPFUN 206 Disable Frame Queueing

The Disable Frame Queueing special function disables the internal queueing of packets. This function may only be issued if all address / protocol pairs have been disabled.

The following is the form of the special function Disable Frame Queueing (.SPFUN 206)

```
Macro Call:

 .SPFUN area, chan, func, buf, wcnt, blkn, [crtn]

where:

  area is the address of a 6 word EMT argument block

  chan is a channel number in the range 0 to 376 (octal)

  func is 206

  buf is the address of a 1 word buffer for the
   returned status word

   The high byte of the status word contains a 0.
   Disable Frame Queueing  returns one of the
   following octal status codes in the low byte
   of the status word:

   0 Success

   6 Address / protocol active

  wcnt is 0

  blkn is 0
```

```
  crtn is the entry point of a completion routine
```

3.6.3 .SPFUN 207 Get Physical Status

The Get Ethernet Address special function returns the physical address of the Ethernet controller card. Unit allocation is not required for this SPFUN

The following is the form of the special function Get Ethernet Address (.SPFUN 207)

```
Macro Call:

 .SPFUN area, chan, func, buf, wcnt, blkn, [crtn]

where:

  area is the address of a 6 word EMT argument block

  chan is a channel number in the range 0 to 376 (octal)

  func is 207

  buf is the address of a 4 word buffer which returns
   the status word (always 0) and the 6 byte
   physical ethernet address

  wcnt is 0

  blkn is 0

  crtn is the entry point of a completion routine
```

3.6.4 .SPFUN 210 Get Handler Status Block

The Get Handler Status Block special function returns the Ethernet handler's internal parameter block. The information contained in the parameter block includes the number of packets received and transmitted for each logical unit as well as timeout information.

The following is the form of the special function Get Handler Status Block (.SPFUN 210)

```
Macro Call:

 .SPFUN area, chan, func, buf, wcnt, blkn, [crtn]

where:

  area is the address of a 6 word EMT argument block

  chan is a channel number in the range 0 to 376 (octal)

  func is 210

  buf is the address of a buffer into which
   wcnt words are written from the ?Q / ?U
   handlers status block.

  wcnt is the number of words to return to the
   buffer.  The status word must be included
   in the word count.

  blkn is 0

  crtn is the entry point of a completion routine

 The first word returned is always a status word of value 0.
 The following 4 byte (long integer) parameters are:
```

```
eiipkt  total receiver interrupts
eiopkt  total transmitter interrupts
eirtmo  total receive timeouts
eixtmo  total transmit timeouts
eirun[8]  received packets for each unit
eixun[8]  transmitted packets for each unit
```

3.7 System Utilities Manual Part 1

Page 15-21

The second bullet under the Optional Type Argument section should read:

/b:value:INS specifies the lowest address to be used by the I-space code in the load module.

3.8 System Internals Manual and Quick Reference Manual

System Internals Manual Page 2-61, Table 2-29
Quick Reference Manual Page 7-31, Table 7-29

Just before the I.SHCP entry add:

- - -	I.TRPS	2	Fully mapped (ZB or ZM) only; Address of supervisor mode trap to 4 and 10 routine defined via .TRPSET
- - -	I.FPPS	2	Fully mapped (ZB or ZM) and FPU only; Address of supervisor mode FPP exception routine defined via .SFPA

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.

The *RT-11 Installation Guide* describes software customizations. In the customizations, symbols are used in place of values and addresses. Use the values and addresses provided in CUSTOM.TXT in place of each symbol shown in the customization. (Monitor values are provided with .MAP files for each monitor.)

4.1 Specific Installation Information

Some installations and upgrades require some specific additional instructions. Please read the sections below if required for your installation.

- Read Section 1 if the system includes a TMSCP (MU) tape drive
- Read Section 2 if installing RT-11 for the first time on a CTI Bus based processor (PRO 300 systems)
- Read Section 3 if installing RT-11 on a CTI Bus based processor currently running one of the versions 5.3 through 5.6
- Read Section 4 if installing RT-11 on a CTI Bus based processor currently running Version 5.1 or 5.2 (or any update of that version)
- Read Section 5 if installing RT-11 on a Q-Bus or UNIBUS processor currently running Version 5.2 (or any update of that version)

Section 1 - System Includes a TMSCP (MU) Tape Drive

MSCPCK.SAV checks and reports TMSCP controller and drive firmware levels. For example:-

```
?MSCPCK-I-MU0-dev FW Rev Level is nnn/HW Rev Level is nnn
```

dev is the device type and *nnn* is the firmware or hardware revision level.

Reliable operation of TK50 tapes on Q-bus processors require a TQK50 controller at Firmware revision 4 and hardware revision 6 or later. If your devices are below these revision levels, contact Digital Services to enquire about an upgrade.

Section 2 - First Installation on a CTI Bus-based processor

RT-11 Version 5.7 does not perform an automatic installation on CTI Bus based processors. Use the following procedures to manually install RT-11

1. Remove diskette 10 from your distribution kit and refer to the *RT-11 Installation Guide* Chapter 9.
2. Follow the steps described in Chapter 9 through Section 9.3. Use diskette 10 as your system disk rather than Diskette 1. To hardware bootstrap your system, turn your processor off then on.

3 Copy the contents of the rest of your software distribution as described in Section 9.4. Disregard the information in Chapter 9 after Section 9.4

4 After copying all the kit to your hard drive, verify the installation by running the VERIFY program.

```
.R IND VERIFY
```

5 After verifying the installation, optionally, run the configuration program, CONFIG, to remove unusable files from your hard disk. CONFIG is optional, but recommended.

```
.R IND CONFIG
```

6 You have then successfully installed RT-11 Version 5.7. Next read the *Introduction to RT-11* to learn about using RT-11

Section 3 - Upgrading a CTI Bus-based processor from Version 5.3, 5.4, 5.5 or 5.6

Use the following procedure to install RT-11 Version 5.7

1 From the software distribution kit, remove diskette 10 and place the diskette in drive DZ0. Diskette 10 is bootable on CTI Bus bead processors. Boot DZ0 and make DW the default data (DK) device

```
.BOOT DZ0:RT11XM.SYS
.ASSIGN DW DK
```

2 Installation of all the files in RT-11 Version 5.7 requires at least 6600 free blocks of disk space. If you do not have sufficient free blocks or your disk space is fragmented into many small sections,

   a. Delete all unwanted files

   b. If you still have insufficient space, back up files from your disk and delete them, for later restoration. See *Introduction to RT-11*

   c. Squeeze your disk

3 If you want to preserve any RT-11 system files, rename them to an arbitrary and unused file type. For example, you may wish to rename STARTX.COM to STARTX.COO

4 Remove protection from all the files on your hard disk

```
.UNPROTECT/SYSTEM DW:
```

5 Use the following command to begin replacing the operating system files on DW: with the Version 5.7 files on DZ and copy the Version 5.7 monitor bootstrap to DW

```
.COPY/PREDELETE/SYSTEM DZ0:*.* DW:*.*
.COPY/BOOT DW:RT11XM.SYS DW:
```

6 When that operation completes, boot the new version on DW:

```
.BOOT DW:
```

7 Remove diskette 10 and insert diskette 1 in DZ0. Beginning with diskette 1 copy the files from each distribution diskette to DW:

```
.COPY/PREDELETE/SYSTEM DZ0:*.* DW:*.*
```

8 When complete run the installation verification procedure

```
.R IND VERIFY
```

9 After you have verified your installation, you can optionally run the configuration program, CONFIG, to remove unusable files from your hard disk. CONFIG is optional, but recommended.

```
.R IND CONFIG
```

10 If you backed up files to make space on your disk, you should restore them now.

11 You have then successfully installed RT-11 Version 5.7. Next read the *Introduction to RT-11* to learn about using RT-11 Version 5.7

Section 4 - Upgrading a CTI Bus-based processor from Version 5.1 or 5.2

Changes were made to the DW handler that require reformatting your hard disk. This process will destroy all data remaining on that disk. Use the following procedure to preserve your data and install RT-11 Version 5.7

1 Boot your current version of RT-11

2 From the software distribution kit, remove diskette 10 and place the diskette in drive DZ0.

3 Issue the following commands

```
.UNPROTECT DW:BUP.SAV
.COPY/PREDELETE DZ0:BUP.SAV DW:BUP.SAV
```

4 Remove Diskette 10 from DZ0 .

5 Delete any files on your hard disk you do not want to preserve

6 You must use BUP to backup all remaining files on your disk, because the DCL BACKUP command is not supported on these versions.

```
.BUP DZ0:=DW:*.*/Z/V
```

This backs up all files on your disk to diskettes you place in DZ0. You will be prompted through the operation and the backup is verified as it proceeds. When complete, remove the final diskette from DZ0 and store the set of backup diskettes in a safe place for later use.

7 Now go to Section 2 of these instructions and continue from there.

8 After installing, verifying and configuring Version 5.7, restore the files needed from the backup created earlier.

If your hardware configuration included an RD52 hard disk, note the following:

Because DW mapping was changed for version 5.3, all RD5x disks that were used by prior versions to 5.3 must be manually reconfigured. Copy any files you want to preserve to RX50 diskettes, using the version of RT-11 that created them. Then initialize the RD5x disk, using RT-11 V5.7. Reconfiguration of the RD52 disks may change their usable size in blocks.

If the disk was manufactured by CDC, or Atasi, the size remains unchanged. If the disk was manufactured by Quantum, the size decreases from 65535. blocks to 65407. blocks. This is due to the correction of an error in previous versions of RT-11.

As a side effect of that reduction in size, any disk image backups must be restored as file images with Version 5.7

9 You have then successfully installed RT-11 Version 5.7. Next read the *Introduction to RT-11* to learn about using RT-11 Version 5.7

Section 5 - Upgrading a Q-bus or UNIBUS processor from Version 5.2 or earlier

Do NOT use any BUP utility or BACKUP command to back up any of your files prior to installing version 5.7 due to incompatibilities between the versions of BUP and backup. Instead, follow this procedure

1 Boot your current (pre Version 5.7) system

2 Locate the volume from the Version 5.7 distribution containing the file BUP.SAV, mount it, and issue the following commands where *ddn* is the device where you mounted that volume

```
.UNPROTECT SY:BUP.SAV
.COPY/PREDELETE ddn:BUP.SAV SY:BUP.SAV
```

3 Remove the volume containing the file BUP.SAV

4 You can now make backup copies of any files before installing version 5.7 but note that you cannot use BACKUP commands, but must instead use BUP commands because this version of RT-11 did not support the DCL command structures.

5 Proceed with the installation as if a new installation

6 When complete, restore any files you want from these backups

4.2 Procedures for Distribution Kits

Automatic installation and verification is supported only with certain distribution kits. If you attempt to bootstrap the automatic installation monitor, RT11AI.SYS, on a distribution kit which does not support it, you will be informed that automatic installation is not supported, and that you must install your system manually by following the procedures in the *RT-11 Installation Guide*. If your configuration does not include a line-time clock or 48K-bytes of memory (FB monitor requirements), you must first copy the bootstrap for the SB monitor from RT11SB.SYS to your system volume while still operating under the automatic installation monitor. Volume 10 of the RX50 distribution kit is bootable on CTI bus-based processors, however, CTI bus-based processors are unsupported for Version 5.7.

4.3 Procedure for Printer Handlers

This section describes restrictions and workarounds for printers on RT-11.

- When output to some printers is terminated prematurely, printing may resume in the middle of the line when restarted. To avoid this, manually reset the printer.
- Under a MTTY monitor with hooks support, it is possible to build an LS handler to use hooks support. When doing so, SET TT CONSOL to another line and install the LS handler using line (LUN) 0.

- The SET LS NOHANG option is valid only if device timeout support is included in a monitor created through system generation. However, It is recommended that you do not use the command SET LS NOHANG. That command can cause printers with very large buffers to cause the LS handler to abort an I/O request before they are through printing.
- If the LP or LS handler NOFORM or SKIP condition is used, load the handler by including a LOAD command in your STRTxx.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 manually move the paper. Instead, the handler should perform all movement of the paper.

  For example, to remove the last printed file from the printer, you should send two form feeds to the printer. Include the command SET LP/LS ENDPAG=2 in your start-up command file.

4.4 Recommended Terminal/Printer SET Options

These are the nondefault SET options recommended for use with the indicated terminals and printers:

LA50, LA75, LA210, LQP02

```
SET LS TAB,CR,CTRL,FORM,NOFORM0,LC,WIDTH=n
```

LA100

```
SET TT NOSCOPE,TAB,FORM,WIDTH=n
SET LS TAB,CR,CTRL,FORM,NOFORM0,LC,WIDTH=n
```

LA30, LA35, LA36, and LS120

```
SET TT NOSCOPE,WIDTH=n
SET LS CR,CTRL,NOFORM,NOFORM0,LC,WIDTH=n
```

LA34 and LA38

```
SET TT NOSCOPE,TAB,WIDTH=n
SET LS CR,CTRL,NOFORM,NOFORM0,LC,TAB,WIDTH=n
```

LA120

```
SET TT NOSCOPE,FORM,TAB,WIDTH=n
SET LS CR,CTRL,NOFORM0,LC,TAB,WIDTH=n
```

LA180

```
SET LP CR,CTRL,NOFORM0,LC,WIDTH=n
SET LS CR,CTRL,NOFORM0,LC,WIDTH=n
```

LG01, LG02

```
SET LP CR,LC,CTRL,FORM,NOFORM0,TAB
SET LS CR,LC,CTRL,FORM,NOFORM0,TAB
```

LJ250

```
SET LS CR,LC,CTRL,FORM,NOFORM0,TAB
```

VT05

```
SET TT TAB,WIDTH=72
```

VT50, VT52, VT55, and VT100 Series

```
SET TT NOCRLF,TAB
```

SET TT commands are not permanent and must be issued every time the monitor is bootstrapped. Therefore, you should include the commands in the appropriate STRTxx.COM file(s).

SET LP and SET LS commands are permanent and should be issued 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 STRTxx.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 STRTxx.COM file to return control to the boot-time console.

A Reporting Problems

This appendix describes submitting problem reports.

Note: If you have a software support contract with Digital Equipment Corporation, you should submit Software Performance Reports (SPRs) to Digital.

All other customers may submit Software Problem Reports (SPR) to Mentec, Inc.

An SPR can be used for:

- Software errors
- Documentation errors
- Follow-up on a previous SPR
- Questions
- Suggestions

An SPR cannot be used for:

- Software license and price policies
- Obvious hardware problems
- Logistical or clerical problems with kits, such as blank media
- Problems with user-written software

In general, when you complete an SPR, use the following guidelines:

- Describe only one problem per SPR.
- Define as accurately as possible the state of the system and circumstances when the problem occurred.
- Illustrate the problem with specific examples.
- If you report a documentation error, specify the title of the manual, and include the section and page number where the error occurred. Include a table or figure number if appropriate.

Categories of SPRs:

- Problem/Error SPR

  This type of SPR contains a software problem. It is assigned a priority of 1 to 5. You receive an answer to this report.

- Suggested Enhancements/Other SPR

  This type of SPR contains a question or suggestion. It is assigned a priority of 5. You may or may not receive an answer.

Priorities Assign a priority of 1 to 5 to your SPR, 1 being the highest using these guidelines:

1 Most production work cannot be run.

— Major system functions are unusable.

— You cannot boot the system.

— Necessary peripherals cannot be used.

2 Some production work cannot be run.

— Certain functions are unusable.

— System performance has declined.

— Installation does not have excess capacity.

3 All production work can be run with some user impact.

— Significant manual intervention is required.

— System performance has declined.

— Installation has excess capacity.

4 All production work can be run with no significant impact on user.

— Problem can be patched or easily bypassed.

5 No system modifications are needed to return to normal production.

— Suggestions are supplied.

— Errors in documentation are noted.

Please supply the following information (in machine-readable form where applicable) when you report a problem:

RESORC

The RESORC/V option displays identifying information for a .SYS, .SAV, or .REL file distributed as part of RT–11. This information includes the RT-11 Release under which the program was compiled, and some identifying information for the program.

The RT-11 Release is labeled as such, and the identifying data is displayed as a series of decimal integers. For example:

```
.RESORC SY:SL.SYS/V
Release = V05, ID(s): 68 -32522 8352

.RESORC SY:LD.SYS/V
Release = V05, ID(s): 63 1282 7171
```

In the event of a problem with a module of RT–11, all the information supplied by RESORC/V should be reported, exactly as it is displayed, along with a detailed description of the problem, in an SPR.

If a failure occurs when you are running privileged, add-on software (for example, the DECnet package), try to reproduce the failure without the additional software. Then, when you write the SPR, indicate how the system operated with and without the add-on software.

B Features New and Changed in RT–11 Version 5.6

B.1 Main V5.6 Features

These were the main features new to RT-11 in Version 5.6:

- Limited Instruction and Data (I and D) space and supervisor mode support. See the *RT-11 File and Formats Manual* for how I and D space is separated in an extended save image.

  Two new extended memory monitors are now available that will support supervisor mode and I-D space. They are RT11ZM.SYS (multiple jobs) and RT11ZB.SYS (single job). See the *RT-11 System Internals Manual*.

  Support for separated I and D space is also included in LINK. See the *RT-11 System Utilities Manual, Part I* for the I and D space options and for how I and D space is overlaid.

- Two new overlay handlers (ZHANDL and SHANDL) to map separated I (instruction) and D (data) space.

  UHANDL is provided on the RT–11 distribution source kit and contains the source code for the OHANDL, VHANDL, XHANDL, and ZHANDL overlay handlers. Conditionals in UHANDL build the different variants of the overlay handler with the exception of SHANDL. SHANDL is built from the SHANDL source file. See the *RT-11 System Utilities Manual, Part I*.

- Creation of the *completely virtual environment* by running virtual background programs either manually or automatically through the rewritten VBGEXE utility. See the *RT-11 System Utilities Manual, Part II*.

  – Inclusion of the SET RUN VBGEXE command. See the *RT-11 Commands Manual*.

  – Inclusion of the new V and VRUN commands to manually run virtual programs in the completely virtual environment. See the *RT-11 Commands Manual*.

  – Inclusion of new data structures to support automatic virtual running.

  – Support for maintaining the completely virtual environment when chaining from one program to another.

- Inclusion of the SB single-job (RT11SB.SYS) version of the FB monitor (replaces the SJ monitor).

- Inclusion of the XB single-job (RT11XB.SYS) version of the XM monitor.

- Common Monitor Sources: RT–11 monitors are now built from one set of common sources. The traditional Single Job Monitor (SJ) has been replaced with a "Single Background" (SB) monitor that is conditionally assembled from these common sources. This will provide a more reliable monitor, that is more consistent as well as more maintainable with the others.

- Support for DH interfaces as a multiterminal option. See the *RT-11 System Generation Guide*.
- Support for building the distributed handler sources LS.MAC and XL.MAC, using SYSGEN, for connection through multiterminal support. See the *RT-11 System Generation Guide*. Documentation is included for coding user-written handlers for this support. See the *RT–11 Device Handlers Manual*.
- Support for virtual to physical address translation in the resident monitor (RMON), using the $JBREL subroutine. (Replaces the ATX.SYS pseudohandler.) See the *RT-11 System Subroutine Library Manual*.
- Support for MQ (message-handling code built into a monitor) is included with all distributed multi-job monitors, but can be removed at SYSGEN, if desired. This allows users of multiple-job systems who do not use, or who infrequently use the message communication features of RT-11 the ability to remove this code from low memory when it is not necessary. In RT-11 V5.5, the MQ handler was always part of the RT-11 resident multi-job monitor.

  MQ is used for communications between foreground and background jobs. MQ is never available in a single job monitor. Nor is it available as the handler MQ.SYS. See the *RT-11 System Generation Guide*.
- Selection of command line interpreter (CLI) support through the SET CLI command. (Eliminates need for a customization patch to support UCF.)

  The CLI support is displayed by the SHOW CONFIGURATION command. See the *RT-11 Commands Manual*.
- Support for the time-of-year (TOY) clock on the 11/93 and 11/94 (KDJ11–E) processors. See the SET NL [NO]TOY command in the *RT-11 Commands Manual*.
- Mapping context changes:
  - Mapping context removed from a job's impure area and placed in a region in extended memory (MCA region).
  - Preservation of the mapping context across most context switches (context no longer automatically recomputed from the WCBs).
  - Mapping context specified by new .CMAP, .GCMAP, and .MSDS programmed requests. See the *RT-11 System Macro Library Manual*.
- New run-time debugging message macros: .DEBUG and .DPRINT. See the *RT-11 System Macro Library Manual* and the *DBG–11 Symbolic Debugger User's Guide*.
- New and changed system subroutines:

```
ICMAP
ICRAW
ICRRG
IELRG
IGCMAP
IGMCX
IUNMAP
ISCCA/MSCCA
JREAD/JREADC/JREADW
JWRITE/JWRITC/JWRITW
MAP
MRCVD/MRCVDC/MRCVDW
MREAD/MREADC/MREADW
MWRITE/MWRITC/MWRITW
MSDAT/MSDATC/MSDATW
MSDS
MSPFN/MSPFNC/MSPFNW
MTATCH
```

See the *RT–11 System Subroutine Library Manual*.

- New programmed requests:

```
.CMAP
.GCMAP
.MSDS
```

See the *RT-11 System Macro Library Manual*.

B.2 Distribution Kit

The RT–11 software distribution kit has changed. See your binary distribution kit and the cover letter sent with it for the contents.

See the *RT–11 Installation Guide* for a brief description of each file that is distributed.

- New Monitor, Handler, and Start-Up Command File Names
  - — Each distributed monitor file has the name RT11mn.SYS, where mn identifies the type of monitor.
  - — Each handler file has the name xx.SYS, where xx identifies the handler. The uppercase X ending some handler names is an essential part of the name and indicates that those handlers are mapped ones.
  - — Each start-up file has the name STRTxx.COM, where xx identifies the monitor.

MONITOR	HANDLERS	START-UP COMMAND FILE
RT11SB.SYS	xx.SYS	strtSB.com
RT11FB.SYS	xx.SYS	strtFB.com
RT11XB.SYS	xxX.SYS	strtXB.com
RT11XM.SYS	xxX.SYS	strtXM.com
RT11ZB.SYS	xxX.SYS	strtZB.com
RT11ZM.SYS	xxX.SYS	strtZM.com

B.3 Hardware

- Support was added for new RA series Disk (DU) storage devices:
  - — RA90 and RA92

    The RA90 is a 1.2G-byte fixed hard disk and the RA92 is a 1.5G-byte fixed hard disk, both supported by the MSCP disk handler DU. They can be divided into multiple 64K-block partitions which operates as a separate disk.
- Support was added for the following printers
  - — LA210 Printer
  - — LA324 Multiprinter
  - — Turbo LA75 Companion Printer
    (the LA75 replacement for Desktop application)
- New Interface Support

  The following lists the new serial-line multiplexors supported in Version 5.6 This new hardware is supported on Q-bus and UNIBUS systems, where appropriate:
  - — *DHQ11* 8-Line Serial Interface for Q-bus, Dual
  - — *DHV11* 8-Line Serial Interface for Q-bus, Quad
  - — *DHU11* 16-Line Serial Interface for UNIBUS, Hex
  - — *CXY08* 8-Line Serial Interface for BA200 (RS-232C)
  - — *CXA16* 16-Line Serial Interface for BA200 (DEC423)
  - — *CXB16* 16-Line Serial Interface for BA200 (RS-422)
  - — *DHF11* 16/32* line interface for Q-bus
  - — *CXF32* 16/32* line interface for Q-bus

  * RT-11 will only support 16 lines. See the *RT–11 System Generation Guide*.

B.4 KED/KEX Editor

The KED editor (KED on an unmapped monitor and KEX on a mapped monitor) has the following two new commands not documented in the *PDP–11 Keypad Editor User's Guide*:

- SET [NO]WORKING

  This command controls the display of the "working" message. The default setting (SET WORKING) causes a message to be displayed every two seconds while KED is doing some time consuming operation. The other setting (SET NOWORKING) suppresses that message.

  The main application for SET NOWORKING is when KED is run on a dual session terminal (for example, a VT330). If KED is set on some long task and you switch to another KED session, the "other session" changed indicator in line 25 changes only at the end of the long KED

operation, rather than once a second (in concert with the "working" message appearance and disappearance).

This command may also be applicable to running KED on "multi-user" systems to reduce extraneous swapping.

- SET ERROR [CONTINUE / STOP]

  This command allows a command file or macro to suppress error termination (that is, to be treated the same as a journal recovery file). Because it uses the same mechanism as the journal recovery file, it should not be used in interactive editing sessions that use journaling, as recovery of such a session is not guaranteed.

  The default setting (unless KED is being run with /RECOVERY) is SET ERROR STOP. This setting causes an error to terminate a command file or macro.

  The default setting when KED is being run with /RECOVERY is SET ERROR CONTINUE. This setting causes an error not to terminate a command file or macro. In all cases, a repeat is terminated by an error.

B.5 RT–11 (DCL) Commands

See the *RT–11 Commands Manual* for current descriptions of all the RT–11 DCL commands.

New Commands

LINK/IDSPACE
SET CLI [NO]DCL, [NO]CCL, [NO]UCL, [NO]UCF
SET LS LINE=n
SET LS [NO]MTTY
SET MODE [NO]SJ
SET NL [NO]TOY
SET RUN [NO]VBGEXE
SET XL LINE=n
SET XL [NO]MTTY
V/VRUN

B.6 Monitors

The monitors in RT–11 V5.6 were slightly different from those in prior releases. The same group of 6 monitors are used in RT-11 V5.7. See Chapter 1 of these Release Notes for a brief Description of these monitors See the *Introduction to RT–11* and the *RT–11 System Internals Manual* for full descriptions of the monitors.

Multiterminal Monitor Enhancement

The Multiterminal Monitor option is enhanced with significant new functionality.

— DH Serial Line Support

— Multiterminal Hooks to handlers

See the *RT–11 System Generation Guide*.

B.7 Utilities and Handlers

See the *RT–11 Utilities Manual, Part I* and *Part II* for the current descriptions of RT–11 utilities. See the *RT–11 Device Handlers Manual* for the current descriptions of RT–11 device handlers.

The following list describes some of the enhancements made to the utilities and handlers for RT-11 V5.6:

Error Logging

RT-11 error logging is modified to run under the new single-job monitors—RT11SB, RT11XB, and RT11ZB.

LINK Utility

The LINK utility is modified to allow multiple passes for linking SAV image programs with separate Instruction and Data Space.

MACRO-11

The file MACRO.SAV distributed on the RT-11 V5.6 kit has increased the default MACRO work file size to 256_{10} from 128_{10}.

SD Handler

The Symbolic Debugger handler (SD) is modified to operate in supervisor mode and separated I-D space. See the *DBG–11 Symbolic Debugger User's Guide*.

VBGEXE

VBGEXE loads most programs into a completely virtual environment. See *RT–11 Utilities Manual, Part II* and the descriptions of the run commands in the *RT-11 Commands Manual*.

B.7.1 System Generation Procedure (SYSGEN)

See the *RT-11 System Generation Guide* for the current description of the SYSGEN procedure, sample answer files, a worksheet to help create a system, and a summary description of the RT–11 conditionals used in creating a system.

New Features

- Three new distributed answer files (SBFB.ANS, XBXM.ANS, and ZBZM.ANS).
- One updated answer file (XMEL.ANS).
- Support for removing the message handler (MQ) from the resident portion of the multi-job monitors.
- Support for the DH family interfaces in the multiterminal environment.
- New DZ operation with multiterminal support.

  Only those remote lines that have been attached by a job (or by a handler using the new handler hooks facility) will have DTR set. Previously all remote lines would always have DTR set.
- Support for terminal hooks in the multiterminal environment, allowing any serial line to function as a printer port under LS, or as a communication port using XL. User handlers may also be written to use the terminal hooks in the multiterminal environment.
- Support for compiling and linking the build files in a completely virtual environment.
- New system conditionals - check the emphasis *System Generation Manual* for details.

B.7.2 System Subroutine Library (SYSLIB)

See the *RT–11 System Subroutine Library Manual* for the current descriptions of all the RT–11 subroutines.

Support for I-D Space

Mapped versions of SYSLIB routines have been added. The name of the routine begins with the letter M and is called in the same manner as an unmapped routine, but with an additional argument which specifies the type of mapping required. In these cases, mapping is shown as an optional parameter in the generic command string. For example, ISDATW/MSDATW is shown as:

```
I = ISDTW (buff,wcnt)
I = MSDATW (buff,wcnt[,bmode])
```

The following routines were added or changed to support separated instruction and data space:

ICMAP	IUNMAP	JREADW	MREAD	MREADC	MSPFNC
ICRAW	ISCCA	JWRITE	MREADW	MRVCDW	MTATCH
ICRRG	JREAD	JWRITW	MRKT	MSDS	MWRITC
IELRG	JREADC	MAP	MSDAT	MSPFN	MWRITE
IGCMAP	JREADW	MRCVD	MSDAT	MSDATC	MWRITW
IGCMX	JWRITC	MRCVDC	MSPFNW	MSDATW	

Other Changed SYSLIB Subroutines

DEVICE	IFPROT	ISPY
GIWRIT	IGTDUS	IUNTIL
GTLIN	ISFDAT	LOOKUP
ICSTAT	ISPFN	SCCA

B.7.3 System Macro Library (SYSMAC)

See the *RT–11 System Macro Library Manual* for the current descriptions of all the macros in the system macro library.

- New Run-Time Debug Message Macros

  These macros provide a simple, uniform way of inserting debugging messages into programs and other system components:

  — .DEBUG — Sets up the environment for the .DPRINT macro.

  — .DPRINT — Conditionally generates code to print a string.

- Modifications to .MTGET and .MTSET for DH Support
- New and Changed Requests to Support I and D Space

  — *New Requests*

.CMAP	.GCMAP	.MSDS

  — Changed Requests

.MRKT		
.MTATCH		
.RCVD	.RCVDC	.RCVDW
.READ	.READC	.READW
.SCCA		
.SDAT	.SDATC	.SDATW
.SFPA		
.SPFUN		
.TRPSET		
.WRITE	.WRITC	.WRITW

C Changes in RT–11 Between V5.0 and V5.5

This appendix is provided as a supplement to the Version 5.7 System release notes extracted from the Version 5.6 System release notes. It summarizes the hardware, software, and documentation components added to and changed in RT–11 between Versions 5.0 and 5.5, inclusive.

C.1 Processors

Support was added for the following processors:

- T–11 Falcon (SBC–11/21 and SBC–11/21 PLUS)
- F–11 Microprocessor Based Boards and Systems
- J–11 Microprocessor Based Boards and Systems

C.2 Storage Devices

Support was added for the following mass storage devices:

C.2.1 DSA Disks

RA60, RA80, RA81, and RA82 Disks

The RA60 is a 204M-byte (formatted) removable disk, the RA80 is a 124M-byte fixed hard disk, the RA81 is a 406M-byte fixed hard disk, and the RA82 is a 600M-byte fixed hard disk, all operating under the MSCP (DU) handler.

Because each of these disks contain more than 64K blocks, they can be divided by RT-11 into multiple 64K-block partitions, each operating as a separate disk.

RC25 Disk

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

C.2.2 MSCP Disks

RD31/RD32/RD50/RD51/RD52/RD53/RD54 Disks

The MicroPDP-11 can include a 10M-byte RD51 disk, a 20M-byte RD31 disk, a 42M-byte RD32 disk, a 33M-byte RD52 disk, a 71M-byte RD53 disk, or a 159M-byte RD54 disk. Each disk is an MSCP device supported by the MSCP disk class handler, DU.

Because the RD32, RD53, and RD54 disks contain more than 64K blocks, they be divided by RT-11 into multiple 64K-block partitions, each operating as a separate disk.

RX33 Diskette Drive

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

RX33 diskettes can be formatted on an RX33 diskette drive with the FORMAT command on systems with an RQDX3 controller (at firmware revision 2 or later)

CAUTION: Do not format RX50 diskettes in the RX33 drive

RX50 diskettes do not use the same oxide (storage medium) as RX33 diskettes and cannot safely store 1.2M bytes of data. RX50 diskettes therefore can randomly lose data if they are formatted as RX33 diskettes.

RX50 Diskette

The RX50 is a dual diskette subsystem for 5 1/5-inch diskettes. Each diskette provides a storage capacity of 400K bytes (800 blocks).

RX50 diskettes are MSCP (mass storage communication protocol) devices using the MSCP disk class handler, DU.

C.2.3 Tape Devices

TK25 Magtape

The TK25 is a streaming TS11-compatible cartridge magtape drive, using a 600-foot, 10-track (1/4-inch format) magtape capable of storing up to 60M bytes of data in 8K-byte blocks, with a record size of up to 16K bytes.

TK50 Magtape

The TK50 is a TMSCP cartridge tape drive which stores approximately 95M bytes of data (unformatted) on 600 feet of 1/2-inch magtape at a density of 6667 bits/in.

TS05 (TSU05/TSV05) Magtapes

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

Data is recorded on an industry-standard 1600 bpi phase-encoded 9-track 1/2-in format magtape.

TU81+ Magtape

RT-11 supports the TU81+ magtape device as a TMSCP data and backup device. The TU81+ is supported in 2 industry-standard modes: 1600 bpi phase-encoded and 6250 BPI GCR (group code recording).

C.2.4 Other Hardware

Support was added for the following hardware:

VT300 and 400 Series Terminals

RT-11 support for the VT300 and 400 series terminals in VT102 mode.

Processor and Memory

Global region base address support and cache-bypass support (See the *RT–11 System Internals Manual*.)

Printers

LA75 printer, the LG01 and LG02 printers, and the LJ250 printer

(See Appendix B of the *RT-11 Commands Manual* for information on printer support.)

C.3 Automatic Installation Procedure

Support was added for an automatic installation and verification procedure. See the *RT-11 Automatic Installation Guide* for details of the procedure.

C.4 System Generation Procedure (SYSGEN)

The system generation procedure and options were improved. See the *RT-11 System Generation Guide* for the current information on the system generation process. Also see the *RT-11 Installation Guide* for details of customizations to be done after SYSGEN.

- Support for the VT11 and VS60 has been removed from SYSGEN.
- SYSGEN includes a new question in the dialog that pertains only to user-supplied device handlers. The SYSGEN dialog asks the following question only after you have indicated you want support for a nondistributed device handler and after you have given the name for that device handler:

  Does your device contain an RT-11 style set option overlay (N)?

  If you override the default and respond YES, RT-11 links that device handler to place the SETOVR program section at an address that is evenly divisible by 512.
- Multiport booting. You can select support for multiport booting for DU, which lets you soft boot RT-11 from any port. Without multiport booting support, you can bootstrap RT-11 only from port 0. As before, you must hard boot DU from only port 0.
- DU initiated bad-block replacement is supported in the distributed XM monitor, but must be generated, if required, in the unmapped monitors.

  It is recommended that you request DU bad-block replacement support if you are using an RC25 or the RA series MSCP disks. Failing to provide support for bad-block replacement for those devices can cause the loss of data or excessive bad blocks on those devices.

- Default values for the CSR and vectors are displayed in parentheses when SYSGEN questions pertaining to a device or MSCP port are asked. Those default values are determined by the CSR and vector addresses that you entered in response to previous questions.

C.5 File Structure

See the *RT-11 Volume and File Formats Manual* for the current information on RT-11 file structure.

Random-Access Devices

— *Home Block Checksum*

For RT-11 V5.5, the checksum value located in the final word of the home block, conforms to the Files-11 On-Disk Structure Specification for calculating the checksum.

— *Prefix Block Indicator*

The first word of the directory entry for each file on a random-access device is the directory entry status word, which describes the status condition of each file. Previously, the low-byte of that word was reserved and always contained the value 000. RT-11 V5.5 defines the following bit in the low byte of the directory entry status word:

Bit	Symbol Meaning when set
020	E.PRE Indicates presence of at least one prefix block for this file.

File prefix blocks are defined as optional information blocks that precede file blocks and begin at logical block 0 of the file. It is the responsibility of the utility or application using the prefix blocks to set bit E.PRE in the status word, create the prefix blocks, and then manipulate them. The RT-11 USR and RMON monitors do not provide support for prefix blocks.

Prefix blocks are available to any utility or application that needs to store information about a file with that file. All previous and current utilities and applications can continue to write and read files without regard for prefix block functionality unless and until that functionality is implemented by a utility or application. Prefix block functionality is of no concern to you unless you implement it.

— *Read-only Indicator*

RT-11 V5.5 defines the following bit in the high byte of the directory entry status word:

Bit	Symbol Meaning when set
040000	E.READ Indicates this file is protected by the monitor against write operations from a .WRITE request. Setting of E.READ does not protect the file from special function write operations or from deletion.

— *Creation Date Word in Directory Entries*

RT-11 V5.5 defines bits 14 and 15 of the Creation Date word in each directory entry as *age bits*. Age bits, when implemented by the user, can extend the directory date by 32(decimal) year increments to the year 2099.

Sequential-Access Devices

Changes have been made to the FSM operations when a tentative file on magtape is not closed, due to a .PURGE request being issued to the magtape device. The tentative file entry is deleted by a series of BACKUP and WRITE_TAPE_MARK operations, leaving the magtape in a known position.

RT-11/VMS Magtape File Interchange

Because of changes made to how the FSM represents file names in the HDR1 and EOF1 label name fields, RT-11 magtapes are more compatible with the VMS operating system ANSI magtape implementation. The format change is backwards compatible with RT-11 utilities and allows text files to be transferred between the RT-11 and VMS operating systems.

C.6 RT–11 Commands

The RT-11 Monitor Commands became known as DCL commands. Many of these commands changed and many new commands were added. A few command options were deleted. See the *RT-11 Commands Manual* for an explanation of all current DCL commands.

Support was added for the following:

- *User Commands First Feature* (UCF)

  UCF support allows you to write your own command preprocessing utility. This is a mechanism to enable you to match a command from a PRELIST.

- *User Command Linkage Feature* (UCL)

  User command linkage (UCL) lets you create your own commands, which allows you to match a last LIST.

- *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.

Most commands have had changes made between Versions 5.0 through 5.7 It is strongly recommended that you read the *RT-11 Commands Manual* for details of the current command operation and syntax. The following list contains a very brief summary of the changed commands implemented for version 5.6

ABORT

New command to abort certain jobs.

BACKUP

New and subsequently updated command to backup information in savesets and provides a DCL interface to BUP.

BASIC

BASIC was removed as a DCL command. To run BASIC, enter BASIC RET.

COMPILE

The /BUFFERING and /PASS:1 options were eliminated.

COPY

— The /RETAIN option (COPY/DEVICE/RETAIN) was eliminated.

— The /DELETE option operation has been modified

— The /PROTECTION and /NOPROTECTION operation has been modified

— The /SYSTEM option to copy SYS files is no longer required unless you use wildcards in the input file type.

— The /VERIFY option can be used for files as well as for entire volumes.

DELETE

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

— The /SYSTEM option is no longer required unless you use wildcards in the input file type.

DIBOL

The /BUFFERING option was eliminated.

DIFFERENCES

— The DIFFERENCES command accepts wildcards to let you compare several files with one command.

— The /SLP option accepts a file specification argument.

DIRECTORY

— The /BACKUP option was eliminated and replaced with the /DIRECTORY option for the BACKUP command.

— The /VOLUMEID[:ONLY] option with /INTERCHANGE can be used to print the volume ID of an interchange diskette.

DISMOUNT

DISMOUNT is now a general DCL command, applicable to any runnable handler that supports the DISMOUNT command.

EXECUTE

Both the /BUFFERING and /PASS:1 options were eliminated.

FORMAT

— You can now format volumes while a foreground job is loaded or when the volume to be formatted contains protected files.

— The number of verification patterns valid for the /PATTERN option has been increased to 16.

INDEX

Support was added for an On-line Index Program (INDEX/INDEXX).

An on-line index program, INDEX, displays on your terminal screen most of the entries contained in the *RT-11 Master Index*. INDEX entries are taken directly from the index files used to create the *RT-11 Master Index*.

INITIALIZE

— The /BACKUP option is eliminated and replaced with the /INITIALIZE option for the BACKUP command.

— The /VOLUMEID[:ONLY] option can be used with the /INTERCHANGE option to write a volume identification on an interchange diskette.

LINK

If you use LINK/BOUNDARY[:value] check the *RT-11 Commands Manual* for its updated usage

MACRO

The /PASS:1 option was eliminated.

MOUNT

MOUNT is now a general DCL command, applicable to any runnable handler that supports the MOUNT command.

PRINT

KMON now assigns precedence to SPOOL for any PRINT command so options specified may not be interpreted correctly when QUEUE and SPOOL are both running.

PROTECT

Assigns a protection status that prevents deletion of a file until you remove the protection.

R

The R command is now identical to the RUN command, exception that the default location for a program executed by the R command is the SY device; and the default location for the utility executed by the RUN command is the DK device.

RECALL

RECALL displays and retrieves commands previously issued, and requires the Single-Line Command Editor be running under the XM monitor.

RENAME

The /SYSTEM option is no longer required to rename .SYS files unless you use wildcards in the input file type.

REMOVE

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

RESET

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

RUN

You can execute virtual jobs from devices other than the system device.

SET

— The SET SL conditions INSERT and REPLACE are now performed by using CTRL/A to toggle between INSERT mode (the default) and REPLACE mode. SL continues to return to INSERT mode after a command is issued.

— The SET SL condition VT52 is no longer supported.

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

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

— The DUn [NO]WRITE options are eliminated.

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

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

SHOW

— SHOW and SHOW SUBSET commands will indicate the default device.

— SHOW MEMORY command displays cache-bypass status for a global region, using the symbol BYP rather than the symbol GBL. Cache-bypass support is described in the *RT-11 System Internals Manual*.

— The SHOW DEVICES command displays installability information for any device handler which cannot currently be installed.

— SHOW SUBSET displays any logical name assigned to a logical disk.

— SHOW DEVICES:DU displays additional DU status information.

— SHOW DEVICES accepts the argument MU (SHOW DEVICES:MU) to display MU status information for the PORT and UNIT of each installed TMSCP controller.

— SHOW COMMANDS command can list the contents of the UCL data file

— The SHOW ALL command now displays the organization of physical memory, logical disk subsetting assignments, and the region type in the extended memory map.

— The SHOW CONFIGURATION command displays a list of system attributes.

— The SHOW DEVICES command displays information about a specific device.

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

— The SHOW QUEUE command shows the contents of the queue for SPOOL or QUEUE, or for both if both are running.

— The SHOW QUEUE command is performed by the RESORC /Q option, rather than by the QUEMAN /L option. However, the QUEMAN /L option is still valid for compatibility.

SRUN

The default file type for the SRUN command is a .REL on the system device (SY:).

TYPE

CTRL/O stops discards the output when multiple files are output to a terminal until another CTRL/O is executed, or until the beginning of the next file.

UNPROTECT

Removes protection from a file so you can delete it.

C.7 Utilities

See the *RT-11 System Utilities Manual, Part I* and *Part II* for the current descriptions of the RT-11 utility programs, unless another manual is referenced in the utility change description.

The following is a brief list of changes made:

BATCH

BATCH cannot be run or require jobs to be run in a virtual environment, and cannot run I-D space programs.

BINCOM

— Can use wildcards with BINCOM to compare multiple binary files.

— The /D option compares two entire volumes starting with block 0.

BUP

Support was added for BUP (Backup Utility Program), which quickly stores information in savesets or logical disks. Numerous changes have been made to BUP between versions of RT-11. The *Introduction to RT-11* contains tutorial information about backing up data, using BUP. The *System Utilities Manual* contains detailed information on using BUP.

CONFIG

The /D, /T, /F options were added. See the *Systems Utilities Manual* for details.

DBG–11

Support was added for DBG–11, an interactive symbolic debugging package for MACRO–11 programs. See the *DBG–11 Symbolic Debugger User's Guide*.

DIR

Additional options for display of protected and unprotected files have been added.

DUMP

When DUMPing a magtape record shorter than 256 words, only those words actually read are displayed. When dumping records longer than 256 words, only the first 256 words are displayed with TRUNCATED in the status line that precedes the displayed data.

DUP

— Bad block handling options have been changed.

— The /R:RET option is no longer supported with the /I option; COPY/DEVICE/RETAIN is not supported. The /R:RET option is ignored.

— Warning and error messages for some operations have been changed. See the *RT-11 System Message Manual*.

— Informational system prompts displayed for some operations have been changed.

— The /H option modifies the behaviour when dealing with bad blocks and should be only used when dealing with MSCP devices.

— The /I option to copy a larger volume to a smaller one requires confirmation before proceeding.

— You can abort a /W (WAIT) operation.

— It is no longer necessary to customize DUP to use variable-size volumes.

EDIT

EDIT is no longer the distributed default editor. To make EDIT your default editor, include the command SET EDIT EDIT in your start-up command file.

ERROR LOGGER

— The maximum number of devices for which error logging support can be generated is now to 22, and each statistical entry requires 11) words. System generation generate support for 10 device units by default

— (T)MSCP error logging is now available for MSCP devices. See the *RT-11 System Utilities Manual* and the *RT-11 Device Handlers Manual* for information.

— The error logger is now reentrant so calls to the error logger no longer need to be serialized (issued only after a .FORK call).

— ELINIT must be run before examining the error log otherwise an error message will be output.

— The block number returned in the device error report is in now decimal radix.

— The Error logger now 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.

FILEX

— The default device for all FILEX operations is DK:.

— A /W (wait) option is added to permit changing volumes

— A /V:[only] option is added when dealing with Interchange diskettes

FORMAT

— FORMAT now supports extended device units

— FORMAT pattern #12 defined as 16-bit pattern 162745.

— FORMAT supports formatting the RX33 diskette.

— The error message, ?FORMAT-U-Channel in use, indicates an internal FORMAT error. If you receive that message, reboot your system and try the operation again. If the error persists, submit an SPR.

— Attempting to format a disk that is not mounted returns the error message, ?FORMAT-F-Device not ready.

— The /VERIFY:only option can be used to perform the write/read verify operation on the following devices: DL, DM, DU, DW, DX, DY, DZ, and RK.

— FORMAT will confirm a request to format a volume that contains protected files or format a volume while a foreground job is loaded.

— You can abort a /W (WAIT) operation.

— Formatting of devices at nonstandard addresses is supported

See the *RT-11 System Utilities Manual, Part I* for more information on those changes to FORMAT. See the *RT-11 Device Handlers Manual* for device-specific formatting information.

HELP

The files HELP.TXT and HELP.EXE, which together make up the program HELP.SAV, are no longer provided on the distribution kit. See the *RT-11 Installation Guide* for information.

IND

Support was added for IND (Indirect Control-File Processor) which will process control files in a similar manner to indirect command files. Numerous changes and updates have been made to IND. See the *RT–11 IND Control Files Manual* for details.

KED/KEX

The PDP–11 keypad editor was improved in several ways and became the default RT–11 editor. On a mapped monitor this editor is called KEX, while on an unmapped monitor the editor is called KED. See the *PDP–11 Keypad Editor User's Guide*.

LET

— LET supports substitution strings of up to 32(decimal) characters. As before, you can define up to 5 symbols concurrently.

— LET supports defining F6 through F10, F14, and F17 through F20, as symbols for string substitutions on keyboards with function keys.

LIBR

— 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.

LINK

— The *n* parameter for the /Y option is optional (/Y[:n]). Not specifying the *n* parameter causes LINK to prompt for up to eight separate PSECT boundary addresses.

— LINK cannot locate and resolve any absolute global symbol, in a library unless the symbol was included in the module using the LIBR /A option or the symbol was associated with at least one relative global symbol.

— RT-11 V5 includes two additional 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.

MACRO–11

— The default size of the MACRO–11 work file can be changed by using a customization patch in the *RT-11 Installation Guide*.

MDUP

— The MDUP programs no longer support the following devices: RP02/RP03, RS03/RS04, and RF11. To create MDUP support for a device not supported, see the *RT-11 Installation Guide*.

— The /V:n option checks that extended memory contains at least the number (n) blocks specified. If available, MDUP executes commands until another /V is encountered. If not available, MDUP does not execute commands until another /V is encountered.

ODT

A customization patch in the *RT-11 Installation Guide* lets you create the ODT debugger which does not require the operating system be loaded in memory.

PIP

Numerous PIP options have been changed, and several new options have been added. Consult the *System Utilities Manual* for comprehensive details.

Queue Package

— The PRINT command is affected when you run SPOOL and QUEUE together.

— QUEUE always appends a form-feed character <FF> to the end of each copy of a queued file.

— The SHOW QUEUE command is now processed by the RESORC /Q option.

— QUEUE no longer concatenates multiple print files into one file.

— Input files are protected from deletion while QUEUE is copying them to the output device.

— The default number of banner pages is set with the /P option.

— Magtape input devices for QUEUE operations no longer rewind between files.

— QUEUE's work file is now SY:QUFILE.WRK.

— See the *System Utilities Manual* for information on new QUEMAN options.

RESORC

RESORC has been significantly modified and enhanced. For details of the options now available, consult the *RT-11 System Utilities Manual*.

SETUP

SETUP (Hardware Setup Utility) was added, which permits setting the operating characteristics for video terminals, printers, and some system clocks. Consult the *RT-11 System Utilities Manual* for details.

SIPP

— Always use the /A option to perform a customization on a distributed or generated monitor.

— When used to patch a file, SIPP changes the creation date of the patched file 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 RUN SIPP rather than R SIPP to permit the command file to be run from other than the system volume.

SL

Significant enhancements have been made to (SL) the Single-Line Command Editor program. See the *Introduction to RT-11* and the *RT-11 Commands Manual* for the current information on SL.

SLMIN

SLMIN (the minimum functionality Single-Line Command Editor) has no new support, but has been modified to make its limited functionality consistent with the current SL.

SLP

— SLP ignores any characters that precede the start-of-update character (-) in SLP command files.

— You can now update more than one file in a single SLP command file.

— SLP /C:[n] checksum option has been added

— SLP /N suppresses creation of a backup file

SPOOL

Support was added for SPOOL (Transparent Spooling Package). Consult the *System Utilities Manual* for more details.

SRCCOM

Consult the *System Utilities Manual* for details on the current operation of SRCCOM. Changes were made to the command line syntax and new options were added.

Native RSX and VMS Transfer Utility

Support was added for TRANSF.TSK and TRANSF.EXE, which let you copy files between your RT-11 stand-alone system and a PDP-11 processor running RSX or a VAX processor running VMS.

UCL

See the *Introduction to RT-11* for information on using UCL.

— You can create and issue UCL commands in upper-, lower-, or mixed case. The case you use to create the UCL command is stored (as entered) in the UCL data file, and the command's case is passed to KMON in the chain area as stored.

— If you change the name of the UCL data file, you can use a customization patch to match the name displayed by error messages with the name you chose for the data file.

  As distributed, the name displayed with UCL error messages is UCL (?UCL-). You can change that 3-character name to any three characters you want, using the following customization patch. This is particularly useful if you have changed the name of the UCL data file. See the *Introduction to RT-11* for information.

— The data structure for the data file UCL.DAT has been redesigned. Data files created prior to RT-11 V5.2 must be manually recreated.

— UCL supports the SHOW COMMANDS command, which displays command definitions on the console terminal. SHOW COMMANDS also supports the /PRINTER and /OUTPUT:filespec options.

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

— The maximum number of user-defined UCL commands in the command table in the UCL data file can be changed (increased or decreased) from 31(decimal) to a maximum you specify.

— You can optimize the UCL utility by combining the save image program (.SAV) file and the data (.DAT) file.

— Informational messages, in the form ?UCL-I-, have been removed from UCL. Warning prompts, in the form Are you sure?, have also been removed. You no longer need to verify your wish to replace or delete a command. Execution of a command to define, redefine, or delete a UCL command is indicated by the appearance of the monitor prompt (.).

VBGEXE

Support was added for VBGEXE (Virtual RUN Utility). VBGEXE lets you run most programs under the XM monitor when not enough low memory is available to run the program by using the R or RUN command.

VTCOM

Support was added for VTCOM (Virtual Terminal Communication Package). VTCOM lets you use your local computer as a terminal when you connect your local computer to a host computer. The *Introduction to RT-11* contains information on using RT-11 communication facilities, including VTCOM. See the *RT-11 System Utilities Manual, Part II* for a complete description of using VTCOM.

— The VTCOM autodialing feature uses the prefix and suffix characters that are appropriate for the DF224 modem. The default prefix is now CTRL/A. The default suffix is now an exclamation mark (!). The customization patches that modify the default prefix and suffix are unchanged.

— VTCOM and TRANSF now progressively restore packet size during continued success from transmitting at a reduced packet size. This process is dynamic; errors halve packet size, while success progressively restores packet size. The maximum and minimum packet size is from 512 bytes to 16 bytes.

— To run VTCOM.REL under XM is inefficient in memory allocation; you should run VTCOM.SAV. If you run VTCOM.REL under the XM monitor, VTCOM returns the message, ?VTCOM-W-Use XM version, VTCOM.SAV.

— Previously, the modem's Clear To Send signal controlled whether VTCOM reported a connection as established or lost. The Carrier Detect signal controls the status of the connection report. The connection report is the same and remains valid for all supported modems.

— Transfer speed enhancements were made to VTCOM for a previous release of RT-11. VTCOM sends data at the interface interrupt speed. That transfer speed can be too fast for the host terminal service to process. A symptom is a beeping terminal. To compensate, slow down the baud rate or let VTCOM adjust the transfer rate, using retries and reduced packet size.

— VTCOM for RT-11 V5.2 cannot be used with earlier versions of the XC and XL device handlers. The XC and XL device handlers for V5.2 cannot be used with earlier versions of VTCOM. Attempts to do so return an error message.

— VTCOM.REL and VTCOM.SAV support the Mini-Exchange, using the VTCOM command SELECT.

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

— VTCOM includes a HANGUP command.

— The VTCOM SEND command sends ASCII files at two speeds: SLOW or FAST. The distributed default speed is SLOW. Use SLOW if the host terminal service does not support XON/XOFF and FAST if it does support XON/XOFF. A customization in the *RT-11 Installation Guide* lets you set the VTCOM SEND command speed.

— The customization patch used to set a default dial string for the VTCOM DIAL command lets you specify the actual character itself, rather than the octal ASCII value for the character.

C.7.1 Monitors

These were the changes to the monitors. See the *RT-11 System Internals Manual* for a description of the current RT–11 monitors.

General

– *BSTRAP* (the monitor bootstrap)

  BSTRAP installs handlers on the system in their $PNAME table order. BSTRAP must perform a restart operation on UNIBUS processors that cannot load UB. BSTRAP no longer ignores set carry bit from system device handler load routine.

– The XM monitor supports 22-bit DMA addresses during I/O operations on supported UNIBUS processors, using the UB pseudohandler.

– The FB monitor is the default monitor for a working system installed through RT-11 automatic installation procedures for all UNIBUS and Q-bus processors.

– The XM monitor becomes a distributed feature rather than a system generation feature.

– BATCH becomes available only through system generation.

– FPU support is included in distributed monitors.

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

  Creating, attaching to, detaching from, and eliminating global regions are controlled by programmed requests in the same manner as regions local to a program.

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

– All monitors 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.

KMON

— KMON accepts numeric job names; that is, job names specified using the /NAME:name option can be numeric. For example:

```
SRUN SY:VTCOM.SAV/NAME:6/PAUSE
LOAD XC=6
RESUME 6
```

Specify 6 in response to the system job prompt (Job?) to call VTCOM. The SHOW JOBS command displays the job 6.

— KMON by default passes unaltered commands by means of the chain area. This procedure is useful for programs that require command input that is not a file specification (such as SETUP commands).

— You can create a KMON preprocessor (UCF) that intercepts all command line input after KMON has tried IND command syntax.

— KMON no longer automatically converts lowercase input to uppercase.

RMON - Resident Monitor

— The XM System Job Prompt changed from Job> to Job?

— *KERNEL Global Region*

RT-11 V5.5 provides a global region named KERNEL, which includes low (kernel) memory from address 00000000 to 00157777. As the I/O page (PAR7) is already a global region (IOPAGE), all kernel memory can be accessed by virtual jobs, using the .CRRG and .CRAW requests. This feature is especially useful for virtual jobs when they access kernel-mapped data structures.

— *IOPAGE Global Region*

The IOPAGE global region maps the I/O region of the monitor and resides in the top 4K words of physical memory. IOPAGE has been assigned the GR.CAC attribute, indicating that cache is bypassed when this region is accessed. This attribute is assigned for consistency only, as the hardware enforces cache bypass on access to the I/O page.

— *$PNAM2 Table*

Used when extended device units are generated.

— *$OWNER Table Support*

Support for the $OWNER table is a system generation feature for the FB and XM monitors. By default, the table is supported and lets a job 'own' a device handler unit (LOAD device=jobnam).

— *Default Device in $PNAM2 Table*

For monitors generated with extended device-unit support, the word following the $PNAM2 table contains the device name for the device, if any, assigned as the default device by the ASSIGN dev * command.

— *Synchronous system traps (SST)*

The SST are context-switched in a more uniform manner.

When a job is first loaded in memory, SST vectors are set to their default value by the FIXTRP code in the R/RUN/FRUN/SRUN load (KMON) code. Once a job is loaded, BPT, IOT, and TRAP vectors are saved and restored (preserved) across any job context switch by the CNTXSW code in RMON. The EMT vector is used to support programmed requests and should not be modified.

— *I/O queue element format changes for UMR support*

UNIBUS mapping registers (UMR) provide a 22-bit DMA memory mapping scheme that is different than the MMU scheme used by the CPU. Therefore, when both schemes are present at the same time, I/O queue elements must be able to keep track of both, which requires an additional offset.

— *Changes to extended memory subroutines for UMR support*

While including the previous versions, RT-11 V5.5 provides new versions of the $GETBY, $PUTBY, and $PUTWR subroutines. Version 5.5 also alters the usage of the $MPPHY subroutine and provides a new subroutine, $MPMEM. The new versions and altered usage are only significant under the following circumstances:

- The processor is a UNIBUS machine with more than 256K bytes of memory.
- The UB pseudohandler is loaded.
- The DMA device handler has been created or updated to recognize the UB pseudohandler.

If those conditions are not met, $GETBY, $PUTBY, $PUTWR, and $MPPHY continue to function as in the past. However, if those conditions are met, $GETBY, $PUTBY, and $PUTWR use the Q.MEM I/O queue element offset, rather than Q.PAR. Offset Q.MEM is described above. $MPPHY, although continuing to use offset Q.PAR, is then restricted to creating UNIBUS addresses for DMA that is mapped by UNIBUS mapping registers. The $MPMEM monitor routine is then used to create CPU memory addresses.

— *Abort Processing Changes*

Abort processing for Version 5.5 has been changed to more fully support internally queued device handlers.

Changes have been made to RMON's abort processing for handlers that have device status bit pattern combinations that involve ABTIO$ and HNDLR$. For RT-11 V5.5, the combination of ABTIO$ and HNDLR$, whether set or clear, determines how RMON performs abort processing for that handler and other handlers loaded in memory. So, you should think about those status bits collectively, rather than individually.

— *Abort code passed from RMON to the handler abort entry point*

This procedure has been modified. Job channel abort information is passed to the handler in R5. Therefore, you should modify any internally queued handler you have written to correctly process the information passed by R5 in the following manner:

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

— *Monitor fixed offsets*

The name for all the monitor fixed offsets has been defined in the .FIXDF macro is the system definition library, SYSTEM.MLB. If you assemble your program with SYSTEM.MLB, you can use the offsets defined in .FIXDF without defining their values in your code.

Including SYSTEM.MLB in an assembly can, depending on the number of invocations and expansions of its macros, require enlarging the size of the macro work file.

Although closely related, the offset names defined in .FIXDF do not all exactly correspond to the names used below and throughout the rest of the RT-11 documentation. If you prefer, you can continue to use the names described below and throughout the documentation so long as you continue to define their value in your code.

The following monitor fixed offsets have been added to the resident monitor database. See the *RT-11 System Internals Manual*:

```
SPSTAT                  EXTIND                  INDSTA
$MEMSZ                  $TCFIG                  $INDDV
MEMPTR                  P1EXT                   IMPLOC
KMONIN                  $PROGD                  $PROGF
WILDDF                  $JOBS                   $QHOOK
$H2UB                   $CNFG3                  $SLOT2
SPSIZE
```

— *Monitor floating offsets*

The following two monitor floating offsets were added to the resident monitor database:

```
CLITYP                  CLIFLG
```

— *Bit definitions*

The following bit masks were defined for fixed offset locations:

```
Offset  Bit Mask

CONFG2  LDREL$
        BUS$
        KXCPU$
        PROS$

SYSGEN  FPU11$
        TSXP$
```

```
SPSTAT  NEXT
        OFF
        ON
        KILL
        ACTIVE
        SHOW
        PRTSCR
        DATIME
        INTEN
        ERROR

INDSTA  CC$IND
        CC$GBL
        LN$IND
        IN$RUN
        IN$IND

$PROGD  $$KED
        $$K52
        $$KEX
        $$EDIT
        $$TECO

$PROGF  $$FORT
        $$F77

$CNFG3  CF3.UI
        CF3.UA
        CF3.UB
        CF3.DM
        CF3.64
        CF3.AT
        CF3.ON
```

C.7.2 Handlers

See the *RT–11 Device Handlers Manual* for the current information specific to particular device handlers.

Changes Made to All Handlers

- XM versions of all supported handlers are included on the RT-11 distribution kit.
- All RT-11 handlers are linked using the /NOBITMAP option.

DL

— The .DRDEF macro used to build DL specifies DMA=YES and PERMUMR=1 to reserve one permanent UNIBUS mapping register (UMR) on UNIBUS processors that support UMR.

— Previously, valid values for the SET dd: RETRY=n command for device handlers were varied. Valid RETRY values for most device handlers that support this command (including DL) are 1 through 127(decimal). The default number of retries is 8.

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

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

DM

— The .DRDEF macro used to build DM specifies DMA=YES and PERMUMR=1 to reserve one permanent UNIBUS mapping register (UMR) on UNIBUS processors that support UMR.

— Previously, valid values for the SET dd: RETRY=n command for device handlers were varied. Valid RETRY values for most device handlers that support this command (including DM) are 1 through 127(decimal). The default number of retries is 8.

— The DM handler 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

DU

Support was added for the DU (MSCP) disk handler. The DU handler supports disk systems that implement the mass storage communication protocol (MSCP). RT–11 supports the following MSCP storage systems: RX33 and RX50 diskettes, RC25 disk, RA60, RA80, RA81, and RA82 disks, RD31, RD32, RD51, RD52, RD53, and RD54 disks.

Note the following:

— Changes have been made to the possible size of the DU translation table and the method you use to access (read and write to) the table. The structure of the table and offset names were changed. See the *RT-11 Device Handlers Manual* for information.

— You can generate support for full MSCP error logging for devices controlled by the DU handler. See the *RT-11 System Generation Guide* for information.

— You can generate support for extended device units for DU, letting you access up to 64(decimal) DU units at the same time. See the *RT-11 Device Handlers Manual* for information.

— DU bad-block replacement is, by default, supported under the XM monitor. DU bad-block replacement is available for the unmapped monitors through the system generation procedure.

— The .DRDEF macro used to build DU specifies DMA=YES and PERMUMR=2 to reserve two permanent UNIBUS mapping registers (UMRs) on UNIBUS processors that support UMR. Also specified is SERIAL=YES to force serialized satisfaction of I/O requests.

— The restriction forbidding the creation of a second DU handler under the XM monitor is removed. As before, you cannot run multiple DU handlers through the same MSCP controller; each handler must have a separate controller. See the *RT-11 Device Handlers Manual* for information.

— Previously, valid values for the SET dd: RETRY=n command for device handlers were varied. Valid RETRY values for most device handlers that support this command (including DU) are 1 through 127(decimal). The default number of retries is 8.

— DU has a status word containing information about the last operation performed by the handler. The status word is called STATU$, and is located at an offset from the base of DU. See the *RT-11 Device Handlers Manual* for information on STATU$.

— You can specify an option during SYSGEN that lets you boot RT-11 from any DU port. If you do not specify DU multiport booting during SYSGEN, you can boot RT-11 from DU port 0 only. The procedure to enable multiport DU booting is described in the *RT-11 Device Handlers Manual*.

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

  Special functions SF.AWR and SF.ARD are especially useful because they return status information in the first word of the return buffer. Status information includes any occurrence of a bad-block error, forced error, or drive error. See the *RT-11 Device Handlers Manual* for information.

— All MSCP (DU) hard-disk systems support bad-block replacement (BBR), performed either by the disk controller or as a feature of the DU handler. For those MSCP hard disks for which BBR is provided by the controller, no support is required by the DU handler; bad-block replacement is transparent to RT-11.

  In MSCP systems that use an RQDX1, RQDX2, or RQDX3 controller, BBR is performed by the controller. In those systems, BBR is done automatically by the hardware and does not require bad-block support in the DU handler.

  In MSCP systems that use a KDA50, UDA50, KLESI-QA, or KLESI-UA controller, BBR can be performed by the DU handler.

  The following table lists the MSCP controllers and drives supported by RT-11 and indicates whether bad-block replacement (BBR) is performed by the controller or the DU handler. (There is no BBR support for RX50 devices or write-only media.)

  MSCP Bad-Block Replacement (BBR)

MSCP Controller	Bad-Block Replaced by:	MSCP Drive
RQDX1	controller	RD51-RD52
RQDX2	controller	RD51-RD52-RD53
RQDX3	controller	RD31-RD32-RD51-RD52-RD53-RD54
KLESI-QA	handler	RC25-RCF25
KLESI-UA	handler	RC25-RCF25
UDA50	handler	RA60-RA80-RA81-RA82
KDA50	handler	RA60-RA80-RA81-RA82

The distributed DU for the XM monitor (DUX.SYS) supports handler BBR. If you are going to use MSCP disks that require handler BBR under a monitor other than XM, you should perform a system generation and request support for DU handler bad-block replacement under the unmapped monitors. Once you have generated such support, you can change monitors and continue DU handler bad-block replacement.

You can force MSCP class devices to clear bad blocks that contain soft errors by coupling the DUP /H option with the /B or /K option. The /H option is not available as a KMON command. You should only use the DUP /H/B or /H/K command options with blank media or a volume you have just backed up.

See the *RT-11 Device Handlers Manual* for information.

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

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

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

— You can determine the status information for a DU (MSCP) device, using the IGTDUS system subroutine. MSCP device status information includes the unit name, the physical device size, the availability of the unit, whether it is removable, and whether it is write protected. See the *RT-11 System Subroutine Library Manual* for details on IGTDUS.

— You can perform absolute (non-file-structured access) reads and writes to any MSCP device, using the JREAD and JWRITE system subroutines. JREAD and JWRITE use a 32-bit starting block number, which lets you read and write to any block on any DU device.

— Some SET commands are changed and new for the DU handler.

DX

Previously, valid values for the SET dd: RETRY=n command for device handlers were varied. Valid RETRY values for most device handlers that support this command (including DX) are 1 through 127(decimal). The default number of retries is 8.

DY

— The .DRDEF macro used to build DY specifies DMA=YES and PERMUMR=1 to reserve one permanent UNIBUS mapping register (UMR) on UNIBUS processors that support UMR.

— Previously, valid values for the SET dd: RETRY=n command for device handlers were varied. Valid RETRY values for DY are 2 through 127(decimal). The default number of retries is 8.

— On Q-bus processors, the DY handler supports only ECO Revision Level F and later controllers.

LD

Support was added for LD (Logical-Disk Subsetting Handler/Utility). The logical disk subsetting handler lets you define logical disks, which are subsets of physical disks. The *Introduction to RT-11* contains complete information on using logical disks. Note the following:

— Changes were made to the possible size of the LD translation tables, the method you use to access (read and write to) the tables, and the structure of tables and offset names have changed. See the *RT-11 Device Handlers Manual* for information.

— The MOUNT command causes LD to initially load any unloaded device containing a file you associate with a logical disk. That association remains in effect until you change it. However, if the device becomes unloaded, you must load it again to access that logical disk.

— You may need to issue the command SET LD CLEAN more than once to verify and correct nested logical disk assignments. Use the SHOW SUBSET command to determine logical disk subsetting assignments and repeat the SET LD CLEAN command as necessary.

— Attempting to nest a logical disk within the same logical disk (MOUNT LD3: LD3:foo.dsk) is invalid and returns the error message, ?LD-F-Invalid nesting.

— Nesting a logical disk within a higher-numbered logical disk (MOUNT LD2: LD3:foo.dsk) is not recommended and returns the warning message, ?LD-W-Invalid nesting.

LP

LP has been converted to a file-structured device handler to support the SET LP ENDPAG=n command. Issuing a directory operation (for example, the PRINT command) to LP while the printer is off line can cause the handler (and possibly the system) to appear hung. Place the printer on line or type CTRL/C twice to free the handler.

LS

— For RT-11 V5.4, the LS handler is distributed. and PDP-11 processors. The SYSGEN conditional that enables that support is LS$PDP=1 for the PDP-11 processors.

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

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

— RT-11 includes support for the LVP-16 plotter printer with the SET LS [NO]GRAPH command.

— LS has been converted to a file-structured-device handler to support the SET LS ENDPAG=n command. Issuing a directory operation (for example, the PRINT command) to LS while the printer is off line or XOFFed can cause the handler (and possibly the system) to appear hung. Place the printer on line or clear the XOFF condition, or type CTRL/C twice to free the handler.

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

MM

— The .DRDEF macro used to build MM (with FSM) specifies DMA=YES and PERMUMR=1 to reserve one permanent UNIBUS mapping register (UMR) on UNIBUS processors that support UMR. MM without FSM requires reserving no permanent UMRs.

  Also specified is SERIAL=YES to force serialized satisfaction of I/O requests.

— Previously, valid values for the SET dd: RETRY=n command for device handlers were varied. Valid RETRY values for most device handlers that support this command (including MM) are 1 through 127(decimal). The default number of retries is 8.

— After MM is loaded or fetched, it clears the RH11 controller when the first operation to that controller is performed. MM continues to also clear the controller in response to certain other operations, such as rewind completion and aborts.

— You can FETCH the MM handler in the XM environment.

MSCPCK.SAV

Support was added for MSCPCK (TMSCP Controller Checking Program), which checks your TMSCP (MU) device controller (if present) and reports its revision level.

MS

— The .DRDEF macro used to build MS (with FSM) specifies DMA=YES and PERMUMR=1 to reserve one permanent UNIBUS mapping register (UMR) on UNIBUS processors that support UMR. MS without FSM also requires reserving one permanent UMR.

  Also specified is SERIAL=YES to force serialized satisfaction of I/O requests.

— Previously, valid values for the SET dd: RETRY=n command for device handlers were varied. Valid RETRY values for most device handlers that support this command (including MS) are 1 through 127(decimal). The default number of retries is 8.

— You can FETCH the MS handler under the XM monitor.

— MS supports 100 in/s streaming for only the TS05 magtapes. However, RT-11 contains no programs (including BUP) that use 100 in/s streaming for TS05 magtapes under mapped monitors.

MT

— The .DRDEF macro used to build MT (with FSM) specifies DMA=YES and PERMUMR=1 to reserve one permanent UNIBUS mapping register (UMR) on UNIBUS processors that support UMR. MT without FSM requires reserving no permanent UMRs.

  Also specified is SERIAL=YES to force serialized satisfaction of I/O requests.

— Previously, valid values for the SET dd: RETRY=n command for device handlers were varied. Valid RETRY values for most device handlers that support this command (including MT) are 1 through 127(decimal). The default number of retries is 8.

— You can FETCH the MT handler in the XM environment.

MU

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

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

Note the following:

— The .DRDEF macro used to build MU (with FSM) specifies DMA=YES and PERMUMR=3 to reserve three permanent UNIBUS mapping registers (UMRs) on UNIBUS processors that support UMR. MU without FSM requires two permanent UMRs.

  Also specified is SERIAL=YES to force serialized satisfaction of I/O requests.

— You can generate support for (T)MSCP error logging for devices controlled by the MU handler. See the *RT-11 System Utilities Manual, Part I* and the *RT-11 Device Handlers Manual* for information.

— You can determine the status information for an MU (TMSCP) device, using the IGTDUS system subroutine. TMSCP device status information includes the unit name, the availability of the unit, and whether it is write protected.

— The distributed MU handler supports only one TMSCP controller. The default CSR and vector addresses for that controller remain 774500 and 260. You must perform a SYSGEN to build support in MU for the second and subsequent TMSCP controllers. See the *RT-11 Device Handlers Manual* for information.

— Previously, valid values for the SET dd: RETRY=n command for device handlers were varied. Valid RETRY values for most device handlers that support this command (including MU) are 1 through 127(decimal). The default number of retries is 8. However, this retry count is only used for certain initialization procedures. Retries for read/write operations are handled by the TMSCP controller and are not settable.

— You can FETCH the MU handler in the XM environment.

NQ, NU

Support was added for the NQ and NU (Ethernet Controller Hardware) handlers. The Ethernet handlers provide hardware support for Ethernet class controllers.

— The NQ handler supports the DELQA and DEQNA Ethernet controllers for Q-bus processors.

— The NU handler supports the DEUNA and DELUA Ethernet controllers for UNIBUS processors.

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

Note the following:

— The .DRDEF macro used to build NU specifies DMA=YES and PERMUMR=3 to reserve three permanent UNIBUS mapping registers (UMRs) on UNIBUS processors that support UMR.

— The distributed NU handler for the XM monitor (NUX.SYS) installs when the distributed XM monitor is booted.

  Bootstrap installation of NU requires device timeout support in both the handler and the monitor. If you build an XM monitor (and NU handler) without device timeout support, NU rejects boot-time installation and allows installation only with the INSTALL NU command, preventing a system hang during booting.

— For RT-11 V5.5, the NU handler does not install if the Ethernet interface hardware fails its self-test.

RK

The .DRDEF macro used to build RK specifies DMA=YES to support UMRs on UNIBUS processors.

SD

Support was added for the SD (DBG–11 Debugger) pseudohandler. Four variants of the SD handler (SDH, SDHX, SDS, and SDSX) support the DBG–11 symbolic debugging package. See the *DBG–11 Symbolic Debugger User's Guide* for the current description of the DBG–11 debugger.

SP

Support was added for the SP (Transparent Spooler) pseudohandler. SP supports the transparent spooler (SPOOL). The SP handler intercepts output directed to the printer and sends it to SPOOL for temporary storage and printing.

Numerous changes have been made to the SP handler and the SPOOL utility. See the *RT-11 Device Handlers Manual* for information on SP and the *RT-11 System Utilities Manual, Part II* for information on SPOOL.

End page support (SET SPn ENDPAG=x) can be generated without flagpage support. Include the new SYSGEN conditional SP$EPS=1 to support end pages without necessarily supporting flagpages.

TT

The default SET options for the multiterminal TT handler are:

```
SCOPE,CRLF,PAGE
```

The default SET options for the multiterminal TT handler were previously NOSCOPE,CRLF,PAGE.

UB

Support was added for the UB (UNIBUS Mapping Register) pseudohandler. UB provides DMA (direct memory access) for I/O devices on UNIBUS processors above the previously-imposed 18-bit memory boundary. UB supports DMA by manipulating UNIBUS mapping registers. UB is supported only on UNIBUS processors that contain more than 256K bytes of extended memory and only under an extended-memory monitor.

VM

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

See the *Introduction to RT-11* for information on using VM as your system volume.

— For RT-11 V5.5, the .DRDEF macro used to build VM specifies DMA=NO to indicate UB compatibility on UNIBUS processors.

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

  When you create a bootable VM device, the VM handler you copy to that device must have its base set to the same value as the VM handler used to copy it to the VM device. That procedure is done automatically when you copy the VM handler residing on your system device to the VM device.

See the *Introduction to RT-11* for information on using VM as a system device and the *RT-11 Device Handlers Manual* for information about the VM handler.

XL

Support was added for XL (Communication Port Handler). XL is a non-file-structured communications handler that supports the virtual terminal communication package, VTCOM. However, its design does not preclude its use in other communication programs. The XL handler supports DL(V)-11 communication ports. XL is required when you use VTCOM.

Note the following:

— XL refuses to be unloaded until you pause or exit from VTCOM.

— 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.

C.7.3 System Subroutine Library (SYSLIB)

See the *RT–11 System Subroutine Library Manual* for the current descriptions of SYSLIB subroutines. See the *RT-11 System Utilities Manual, Part II* for a description of XHANDL.

Note: Because IQSET is no longer in SYSLIB.OBJ, FORTRAN programmers who need to add queue elements for certain other SYSLIB functions, should refer to the FORTRAN IV distributed FORLIB and the FORTRAN-77 distributed F77OTS.

New Routines

The following routines were added to SYSLIB:

```
CALL$F                  DATE/DATE4Y             IDATE
IABTIO                  ICLOSZ                  ICNTXS
IDCOMP                  IFPROT                  IFREER
IFWILD                  IGETR                   IGFDAT
IGFINF                  IGFSTA                  IGTDIR
IGTDUS                  IGTENT                  IHERR/ISERR
IPROTE/IUNPRO           IPUT                    ISDTTM
ISFDAT                  ISFINF                  ISFSTA
ISPCPS                  ISWILD                  IWEEKD
JREAD/JREADC/JREADF/JREADW
JWRITE/JWRITC/JWRITF/JWRITW
KPEEK                   KPOKE                   RAN/RANDU
$SYTRP                  XHANDL
```

Changes Between SYSLIB and FORTRAN OTS (FORLIB and F77OTS)

— The following functions and subroutines have been removed from the distributed RT-11 system subroutine library, SYSLIB. They were specific to FORTRAN programming and did not work without a resident FORTRAN OTS. They have been added to the FORTRAN IV distributed FORLIB and the FORTRAN-77 distributed F77OTS.

```
GETSTR                  IFREEC                  INTSET
IASIGN                  IGETC                   IQSET
ICDFN                   IGETSP                  PUTSTR
IFETCH                  ILUN                    SECNDS
```

— The functions and subroutines DATE, IDATE, RAN, and RANDU previously located in the distributed FORTRAN subroutine libraries, are now located in SYSLIB.OBJ.

Changes to SYSLIB Subroutines

The following subroutine library modules in SYSLIB changed.

```
GTLIN                   ICSTAT                  IGTDUS
ISFDAT                  ISPFN (ISPFN/ISPFNC/ISPFNF/ISPFNW)
SCCA
```

Extended-Memory Subroutines Added in V5.5

The following extended-memory subroutines were added to the file XMSUBS.MAC. See the *RT-11 System Internals Manual*.

```
$JBREL                  $MPMEM                  $XDEALC
```

C.7.4 System Macro Library (SYSMAC)

See the *RT-11 System Macro Library Manual* for the current descriptions of programmed requests and macros.

— The following programmed requests and macro, located in SYSMAC.SML, were added:

```
.ABTIO                  .ADDR                   .ASSUME
.BR                     .CALLK                  .CKXX
.CLOSZ                  .FPROT                  .GFDAT
.GFINF                  .GFSTA                  .PEEK
.POKE                   .PVAL                   .SFDAT
.SFINF                  .SFSTA                  SOB macro
.DREST (Device Handlers Only)
.DRINS (Device Handlers Only)
.DRPTR (Device Handlers Only)
.DRSPF (Device Handlers Only)
.DRTAB (Device Handlers Only)
.DRUSE (Device Handlers Only)
```

— The following system macros changed:

```
.CLOSE                  .CSIGEN                 .CSTAT
.DRBEG                  .DRDEF                  .DREND
.DRPTR                  .ELRG                   .FETCH
.FPROT                  .GTLIN                  .HERR/.SERR
.MAP                    .PEEK                   .POKE
.PURGE                  .PVAL                   .QELDF
.RDBBK                  .SAVESTATUS             .SCCA
.SFDAT                  .SPFUN
```

D Accessories

We are pleased to be able to include a number of useful accessories with RT-11 Version 5.7. These accessories are NOT supported and are for use on an AS-IS basis only. Software Performance Reports and requests for support for these accessories will be treated as suggestions only. Not all of these are available on the RX50 distribution for RT-11.

These Accessories Include:

Kermit for RT-11
TECO for RT-11
ARK (an Archiver)
DISSAV (a SAV file disassembler)

Other as-is accessories such as TCP/IP for RT-11 may be obtained by contacting Mentec.

We would like to thank the Columbia University Kermit Project for permission to distribute Kermit for RT-11 with this kit. Comprehensive documentation for Kermit, the Kermit protocols, and Kermit implementations for other hosts may be obtained from the following address:

The Kermit Project
Columbia University
612 West 115th Street
New York
NY 10025-7799

For information about Kermit on the web, visit

http://www/columbia.edu/kermit/index.html

We would like to thank the other contributors of Accessories for RT-11

E RT–11 V5.7 Year 2000 Testing Summary

RT–11 Version 5.7 has been thoroughly tested for Year 2000 Compliance and will correctly interpret years from 1972 through 2099.

RT–11 V5.7 has been found to be compliant with British Standards Institute (BSI) DISC PD2000–1, for the years tested: 1972 through 2027.

RT–11 V5.7 is not compliant with either ISO 8601:1988, or ANSI X3.30 standards with regard to date presentation. All dates are presented as DD-MMM-YYYY, or in some special cases dates will be displayed as DD-MMM-YY.

For those dates displayed using only two digit years, and based on PD2000–1 the century inferencing rules will specify that years greater than seventy (71) will be in the twentieth century (preceded by 19), and those less than seventy two (72) will be in the twenty–first century (preceded by 20).

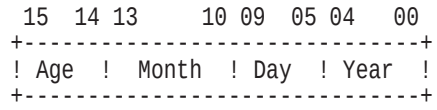
For dates input, inferencing is only permitted for two digit year references 72 through 99 inclusive, specifying the years 1972 through 1999 inclusive. No other dates may be permitted by inferencing.

All product components have been extensively tested to ensure compliance with the DISC PD2000–1 standard.

E.1 Internal Date Storage

The date format used by the utilities conforms to the standard time format used by RT–11, and all utilities display the year as four digits.

Below is shown the date/time format which is used by RT–11;



Prior to version 5.6, the age bits were intentionally unused. User applications may be affected by the changes incorporated for full Year 2000 compliance. User applications should be carefully checked to ensure that the use of bits 14 and 15 in the date word will not cause applications to have unexpected results.

E.2 Disk File System

The RT–11 file structure include a date word to maintain the file creation date. This word is identical to the format used by the system. If the system date/time is not set when the system is booted, a value of zero is written, which corresponds to an invalid date, since both the month and day are zero.

The valid range of dates for the file system are from 1–Jan–1972 through 31–Dec–2099.

All RT–11 Utilities tested correctly interpret the date word stored with each file.

E.3 Magnetic Tape Standards

Date information is stored within the ANSI Header 1 label (HDR1), which specifies the creation date of a given file. This date is specified to be of the form _yyddd, where "_" represents a blank, and "yy" is the year within the 1900s and "ddd" is the day within that year. Each digit is stored as a single ASCII digit (X3.27-1978). The level 4 standard replaces the "_" blank character with "0" for the year 2000 and the "yy" represents the year of the century.

RT–11 implements ANSI standard X3.27-1978, at Level 4. At this level, RT–11 can handle dates through 2099 without error. For the year 2000, a field previously described as a "blank" is changed to an ASCII "0".

Table E–1 RT–11/E ANSI tape dates

Date	ANSI
10–JAN–1998	_98010
10–JAN–2000	000010
10–JAN–2010	010010

RT–11 implements ANSI compliant level 4 records, on all utilities tested. This ensures that all media transfers will be compatible with other systems implementing the ANSI level 4 standard.

E.4 Foreign System Interchange

The FILEX utility was tested on DOS directory structures, and produced a correct directory throughout the supported date range.

The other functions of FILEX have not be extensively tested, however the code is in place to support the valid dates supported by the file structure.

E.5 Networking

DECnet/RT is no longer a supported product and is not included.

E.6 Backup Utilities

The backup utilities included with RT–11 are BUP, DUP, and to a lesser degree PIP. Only PIP includes date range selection, and as with the normal file system utilities, these function properly for all supported dates.

There are no date range selection options for either BUP, or DUP, however, date field preservation is observed, and all dates are properly propagated.

E.7 File System Utilities

The TECO editor correctly reports the operating system date, including the age bits. Those TECO macros which manipulate the date word should be aware that the "^B" function will return the age bits as well as the traditional year bits.

All other file system utilities, most notable PIP, DIR, and QUEMAN correctly process all date, in the range specified.

E.7.1 CSI Date Input

The permitted date format on entry for RT–11 is more extensive than documented in the RT–11 manual, and does not conform to any "standard". Particularly, since date fields are separated with ":", rather than "–", the parsing syntax does not conform to any of the Industry Standard mechanisms.

Since the syntax was established well before any of the documented standards, certain restrictions were imposed to ensure that the syntax would not be ambiguous, and would therefore meet the BSI–DISC PD2000–1 standard.

For RT–11, a date is a variable length option, from 1 to 3 arguments, and allowed to be specified in an arbitrary manner. Due to the fact that date specifications are not required to be in a specific format, certain rules are applied when imputing dates.

Permitted formats for dates in CSI command strings are as follows:

Table E–2 CSI Date fields

Field	Rule(s)
Day	The day of the month is any field which is in the range from 1–31, inclusive. If two fields fall into the given range, the date is considered invalid. If no day fields are present, the current day from the system date is used.
Month	The month, if specified, must include three alpha characters, representing the English month abbreviation; i.e. JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC. If a month is not specified, the current month from the system date is used.
Year	The year, if specified, must be within the following range; If specified as two digits, 72 through 99, and if specified as four digits, 1972 through 2099. If the year is specified as two digits, it is assumed to be in the range 1972 through 1999. The following numeric values will be considered invalid, in order to ensure no false assumptions; 0, the range 32 through 71, the range 100 through 1971, and above 2099.

In previous versions of RT–11, some erroneous fields were ignored. For example, the numeric values 0, and 32 through 71, if entered, were ignored, and resulted in the current value for the field to be used. Therefore, the date, 01:JAN:00, could be interpreted as 1–Jan of the current year. If the current

year was 2000, the results would be correct. Otherwise, the results would be based upon the current year.

Also, any field which resulted in a negative value being operated on would be ignored. This is no longer the case. In previous versions, a date specification of "–1:–1:–1" would be interpreted the same as if there were no arguments. For RT–11 V5.7, this will result in a syntax error.

Note: ERROUT, unlike other CSI utilities requires the date elements to be entered as described in the documentation. i.e. dd:mmm:yy[yy]

Below are some examples of date specifications. It is assumed that the current system date is 31–Dec–2099, for the purposes of defaulting current values into the resulting date.

Table E–3 CSI Date examples

Date Specified	Result
1	1–DEC–2099
JAN	31–JAN–2099
2000	31–DEC–2000
1:JAN	1–JAN–2099
JAN:1	1–JAN–2099
2000:JAN	31–JAN–2000
JAN:2000	31–JAN–2000
1:2000	1–DEC–2000
2000:1	1–DEC–2000
1:JAN:2000	1–JAN–2000
2000:JAN:1	1–JAN–2000
JAN:1:2000	1–JAN–2000
JAN:2000:1	1–JAN–2000
1:2000:JAN	1–JAN–2000
2000:1:JAN	1–JAN–2000
2000:0:JAN	Invalid
1:JAN:00	Invalid
31:JAN:31	Invalid
1:JAN:70	Invalid
0:XXX:-1	Invalid
0:JAN:98	Invalid

These last values shown, noted as being invalid, were previously a legal syntax, with varying results.

E.8 Assemblers, Compilers, and Linkers

Each of the following utilities, was tested for compliance:

- MACRO–11
- LIBR
- LINK
- CRF

The primary area of testing was in output listing files to ensure that the date data presented was accurate. In each of the cases, all date information was presented correctly on the output listings. For MACRO–11, the date is presented in 2–digit format, and the century inferencing rules should be applied for the date range displayed in the listing from 1972 through 2071.

There were no date specific switches available for any of the utilities which is impacted by the Year 2000.

E.9 Batch and Queue

QUEMAN was fully tested for date range selection, and functioned properly. There are no delayed queuing functions included in RT–11, and as such the Year 2000 is not related to that functionality.

SPOOL also has no date related functionality, and is not affected by dates.

E.10 System Management

While most system management utilities do not involve date range selection, the Error logging subsystem is the primary exception. All dates are properly stored in the error log file, and date range selection using the ERROUT utility function properly.

The DATE function, as well as the DATIME utility do function properly for all supported dates, and correctly manipulate the age bits in the date word.

The SETUP utility will correctly process the date and time for the TOY clock, and verify the syntax of a date correctly. Since it interacts with the TOY clock, the maximum unambiguous date allowed is 2071, since a two digit date of 72 is assumed to be 1972 from the TOY clock. SETUP will allow the year to be set to any allowable value, up to 2099.

E.11 Certification

RT–11 V5.7 is compliant with the DISC PD2000–1 standard. It has been fully tested for compliance between the dates of 1–Jan–1972, and 31–Dec–2027, including leap year dates.

In any case where inferencing rules are required for use within the stated date range, those functions which specify two digits with year nvalues of 00 through 71 are inferred to be in the twenty-first century, and 72 through 99 are inferred to be in the twentieth century.

Below is listed a table of the functions which have been tested, and their theoretical date limitations;

Table E–4 RT–11 Utilities Tested

Utility	Tests	Limiting Date
Directives	.ABTIO, .CALLK, .CHAIN, .CLOSE, .CLOSZ, .CMKT, .CSIGE, .CSISP, .CSTAT, .CTIMI, .DATE, .DELET, .DPRIN, .DSTAT, .ENTER, .EXIT, .GFDAT, .GTIM, .GTJB, .MRKT, .RENAM, .SDTTM, .SFDAT, .TIMIO, .TWAIT, .WAIT, IWEEKD	Dec–2099
System—Date	Date rollover, leap year / new year	Dec–2099
System—File System	Disk File Structure	Dec–2099
System—File System	ANSI Tape Structure	Dec–2099
DCL COMMANDS		Dec–2099
BACKUP	No date dependencies	Dec–2099
BOOT	No date dependencies	Dec–2099
COMPILE	No date dependencies	Dec–2099
COPY	DUP,PIP,FILEX	Dec–2099
DATE		Dec–2099
DELETE	PIP,FILEX,QUEMAN	Dec–2099
DIBOL	No date dependencies	Dec–2099
DIFFERENCES	SRCCOM,BINCOM	Dec–2099
DIRECTORY	DIR,BUP,FILEX	Dec–2099
DUMP	No date dependencies	Dec–2099
EDIT	EDIT,TECO,KED,KEX	Dec–2099
INITIALIZE	DUP,BUP,FILEX	Dec–2099
LIBRARY	LIBR	Dec–2099
LINK	LINK	Dec–2099
MACRO	MACRO	Dec–2099
MOUNT	No date dependencies	Dec–2099
PRINT	QUEMAN,PIP	Dec–2099

Table E–4 (Cont.) RT–11 Utilities Tested

Utility	Tests	Limiting Date
PROTECT	PIP	Dec–2099
RENAME	PIP	Dec–2099
SHOW	RESORC,ERROUT,QUEMAN	Dec–2099
SQUEEZE	DUP	Dec–2099
TIME		Dec–2099
TYPE	PIP	Dec–2099
UNPROTECT	PIP	Dec–2099
UTILITIES		Dec–2099
BINCOM	No date dependencies	Dec–2099
BUP	file creation dates	Dec–2099
CONFIG	No date dependencies	Dec–2099
CONSOL	No date dependencies	Dec–2099
CRF	listing files	Dec–2099
DATIME	Date function correct	Dec–2099
DIR	DIR/BEFORE, DIR/DATE, DIR/SORT:DATE, DIR/SINCE	Dec–2099
DUMP	No date dependencies	Dec–2099
DUP	File creation dates	Dec–2099
EDIT	File creation dates	Dec–2099
ERRLOG	ERROUT date selection	Dec–2099
ERROUT	/F:date, /T:date	Dec–2099
FILEX	DOS disk directory	Dec–2099
FORMAT	No date dependencies	Dec–2099
LD	No date dependencies	Dec–2099
LET	No date dependencies	Dec–2099
LIBR	Library listings	Dec–2099
LINK	Map files	Dec–2099
MACRO	listing files	Dec–2099
NITEST	No date dependencies	Dec–2099
ODT	No date dependencies	Dec–2099
PAT	No date dependencies	Dec–2099
PIP	/C:date, /I:date, /J:date, /T:date	Dec–2099
QUEUE	No date dependencies	Dec–2099
QUEMAN	/C:date, /I:date, /J:date	Dec–2099
RESORC	No date dependencies	Dec–2099
RTMON	display banner	Dec–2099
SETUP	DATE:date	Dec–2099
SIPP	No date dependencies	Dec–2099

Table E–4 (Cont.) RT–11 Utilities Tested

Utility	Tests	Limiting Date
SLP	No date dependencies	Dec–2099
SPLIT	No date dependencies	Dec–2099
SPOOL	No date dependencies	Dec–2099
SRCCOM	No date dependencies	Dec–2099
TRANSF	No date dependencies	Dec–2099
VBGEXE	No date dependencies	Dec–2099
VTCOM	No date dependencies	Dec–2099