RT–11 System Release Notes

Order Number AA–5286L–TC

August 1991

This manual summarizes features of the RT–11 operating system, V5.0 through V5.6. Changes to RT–11 after the publication of Version 5.6 are included in V5NOTE.TXT distributed with the RT–11 system kit.

Revision/Update Information:	This manual supersedes *RT–11 System Release Notes,* AA–5286K–TC.
Operating System:	RT–11 Version 5.6

Digital Equipment Corporation
Maynard, Massachusetts

First Printing, March 1980
Updated, June 1980
Revised, March 1981
Revised, March 1983
Revised, December 1983
Updated, July 1984
Revised, April 1985
Revised, November 1985
Revised, July 1986
Revised, January 1988
Revised, August 1989
Revised, August 1991

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation.

Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

Any software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license. No responsibility is assumed for the use or reliability of software or equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

The Reader's Comments form at the end of this document requests your critical evaluation to assist in preparing future documentation.

S1525

This document was prepared using VAX DOCUMENT, Version 1.2.

Contents

Chapter 3 Current Restrictions and Corrected Problems

Chapter 4 Installation, Bootstrap, and Hardware Setup Procedures

Appendix A V5.6 Documentation Additions and Corrections

Preface

Intended Audience

This manual is for all RT–11 users, but especially system programmers.

Description of the Manual

This manual describes:

- Features of RT–11 Version 5.6.
- Changes in RT–11 Version 5.
- Current restrictions and corrected problems from Version 5.0 to 5.6, inclusive.
- Some installation kit procedures
- V5.6 Documentation Corrections

Document Structure

Chapter 1 — Features New and Changed in RT–11 Version 5.6

Summary of new and changed features of Version 5.6.

Chapter 2 — Changes in RT–11 Between V5.0 and V5.5

Summary of hardware, software, and documentation components added or changed between Version 5.0 and 5.5, inclusive.

Chapter 3 — Current Restrictions and Corrected Problems

Current restrictions and corrected problems, organized by functional category or utility. Version 5.6 restrictions and corrected problems are shown by the change bar in the page margin.

Chapter 4 — Installation, Bootstrap, and Hardware Setup Procedures

Recommended procedures for installing the distributed kit and hardware.

Appendix A — Documentation Additions and Corrections

Additions and corrections to data contained in specific manual(s) of the RT–11 documentation set.

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 Quick 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

- This manual is organized alphabetically according to the topics listed in each chapter's table of contents. For example, monitors are organized alphabetically under the title **monitors** and so on.

  See the page headings to find the topics.

- Change bars | in page margins indicate changes made in Version 5.6.

Chapter 1

Chapter 1 Features New and Changed in RT–11 Version 5.6

Chapter 1
Features New and Changed in RT–11 Version 5.6

Main V5.6 Features

These are 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 handers (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.

Main V5.6 Features

- 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
  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*.

- An updated, reorganized, and rewritten documentation set (13 volumes). See the *Guide to RT–11 Documentation*.

Documentation

The RT–11 V5.6 documentation set has been totally reorganized and updated. Each of the changed manuals contains new and revised information. The following table lists all the manuals in the documentation set. Next to each new manual is the name of the Version 5.5 manual that is its primary source.

Updated or New	Source
Basic Books	
Guide to RT–11 Documentation	
Introduction to RT–11	
RT–11 Commands Manual	*RT–11 System User's Guide*
RT–11 Quick Reference Manual	*RT–11 Mini-Reference Manual*
RT–11 Master Index	
RT–11 System Message Manual	
RT–11 System Release Notes	
Installation Specific Books	
RT–11 Installation Guide	
RT–11 System Generation Guide	
Programmer Oriented Books	
RT–11 System Utilities Manual, Part I	*RT–11 System Utilities Manual*
RT–11 System Utilities Manual, Part II	*RT–11 System Utilities Manual*
RT–11 IND Control Files Manual	*RT–11 System User's Guide*
RT–11 System Macro Library Manual	*RT–11 Programmer's Reference Manual*
RT–11 System Subroutine Library Manual	*RT–11 Programmer's Reference Manual*
RT–11 System Internals Manual	*RT–11 Software Support Manual*
RT–11 Device Handlers Manual	*RT–11 Software Support Manual*
RT–11 Volume and File Formats Manual	*RT–11 Software Support Manual*
DBG–11 Symbolic Debugger User's Guide	

The following RT–11 documents (not revised for this release) were revised for RT–11 V5.5:

PDP–11 Keypad Editor User's Guide
PDP–11 Keypad Editor Reference Card
RT–11 Automatic Installation Guide
RT–11 Update User's Guide
PDP–11 MACRO–11 Language Reference Manual

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.COM, 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

- **New Files**

  The following are new files in the distribution kit:

```
EDTG.MAC
ICMAP.MAC
MODES.MAC
MSB.COM
MUB.COM
PLAS.MAC
RMON.MAC
SB.MAC
SBFB.ANS
SHANDL.MAC
STRTAI.COM
STRTFB.COM
STRTSB.COM
STRTXB.COM
STRTXM.COM
STRTZB.COM
STRTZM.COM
SYSLBX.MAC
UHANDL.MAC
XB.MAC
XBXM.ANS
ZB.MAC
ZBZM.ANS
ZHANDL.MAC
ZM.MAC
```

- **Removed files**

  The following files were removed from the RT–11 software distribution kit:

  ```
  AI.CND
  AI.TBL
  AT.MAC
  BL.ANS
  BL.CND
  BL.TBL
  EDTGBL.MAC
  HD.CND
  HD.TBL
  RMONFB.MAC
  RMONSJ.MAC
  SJ.MAC
  SJFB.ANS
  SJFB.CND
  SJFB.TBL
  STARTA.COM
  STARTF.COM
  XM.TBL
  STARTS.COM
  STARTX.COM
  TT.MAC
  XM.ANS
  XM.CND
  XM.TBL
  ```

Hardware

- Support is added for the RA90/RA92 Disk (DU) storage devices:

  - **RA90**

    The RA90 is a 1.2G-byte fixed hard disk supported by the MSCP disk handler DU.

    Because the RA90 disk contains more than 64K blocks, it must be divided into multiple 64K-block partitions. When running under RT–11, each partition operates as a separate disk.

  - **RA92**

    The RA92 is a 1.5G-byte fixed hard disk supported by the MSCP disk handler DU.

    Because the RA92 disk contains more than 64K blocks, it must be divided into multiple 64K-block partitions. When running under RT–11, each partition operates as a separate disk.

- Support is added for the following printers

  - LA2100 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 of the RT–11 operating system. This new hardware will be supported on the 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*.

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.

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

Monitors

The monitors in RT–11 V5.6 are slightly different from those in prior 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 5.6, you can have any one of the following monitors:

Single-Job Monitors

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

Multi-Job Monitors

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

Explanations for Numbered Items in Monitor Descriptions

❶ A single-job monitor can run only one job at a time.

❷ An unmapped monitor does not have the extended-memory environment.

❸ A partially mapped monitor is one that has the traditional XM (extended-memory) environment.

❹ 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.

❺ A multi-job monitor can run up to eight jobs at a time.

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*.

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.

  LINK is also used as part of the process to produce supervisor libraries.

- **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*.

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

- Support for the V5.6 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).
- Support for removing the message handler (MQ) from the resident portion of the monitor. Support for MQ is included with all distributed multi-job monitors, but can be removed at SYSGEN, if desired.
- Support for the DH family interfaces in the multiterminal environment.
- New DZ operation with multiterminal support.

  Before V5.6, if a multiterminal monitor included support for DZ lines, all lines on the DZ controller would have DTR set, regardless of whether the line is local or remote or if the line is in use or not.

  In V5.6, 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.
- 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 can be written to use the terminal hooks in the multiterminal environment.
- Support for compiling and linking the build files in a completely virtual environment.
- Two new conditionals for naming the distributed monitors: SB and SUP$Y.
- New system conditionals

  ```
  DHLE$N      DHUS$1
  DH11$L      DHSP$D
  DH11$M      DHST$P
  DH11$N      DH$B0n
  DHCS$n      DH$B1n
  DHVC$n      LS$LUN
  DHSZ$n      LS$MTY
  DHUS$0      MTY$HK
  XL$LUN      XL$MTY
  ```
- A reorganized and updated SYSGEN procedure.

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**

  A major feature of V5.6 SYSLIB support is to add mapping routines beginning with letter **M**.

  The mapped version of a routine is called in the same manner as an unmapped version, but has an additional argument that 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
  ICRAW
  ICRRG
  IELRG
  IGCMAP
  IGCMX
  IUNMAP
  ISCCA
  JREAD
  JREADC
  JREADW
  JWRITC
  JREADW
  JWRITE
  JWRITW
  MAP
  MRCVD
  MRCVDC
  MREAD
  MREADW
  MRKT
  MSDAT
  MSDAT
  MSPFNW
  MREADC
  MRVCDW
  MSDS
  MSPFN
  MSDATC
  MSDATW
  MSPFNC
  MTATCH
  MWRITC
  MWRITE
  MWRITW

- **Other Changed SYSLIB Subroutines**

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

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

When Making an SPR

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.

Chapter 2

Chapter 2 Changes in RT–11 Between V5.0 and V5.5

Chapter 2

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

This chapter summarizes the hardware, software, and documentation components added to and changed in RT–11 between Versions 5.0 and 5.5, inclusive. See the master index and the appropriate manual for descriptions of each component.

Documentation Revised in V5.5

The following manuals were new or revised for RT–11 V5.5.

- *DBG–11 Symbolic Debugger User's Guide*
- *RT–11 Automatic Installation Guide*
- *Guide to RT–11 Documentation*
- *Introduction to RT–11*
- *RT–11 System Message Manual*
- *PDP–11 MACRO–11 Language Reference Manual*
- *PDP–11 Keypad Editor User's Guide*
- *RT–11 Update User's Guide*

Software Update Process Revised

RT–11 V5 provided an update process replacing changed software modules so that patches were no longer necessary.

Update kits are distributed periodically. Each update kit includes an automatic procedure for replacing software modules with the replacement software modules. The update procedure guides you through an interactive dialog to determine which products you want to update and which devices you are using. Then, the update software performs the module replacement operations.

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

Distribution Kit

The RT–11 software distribution kit changed.

See the online release notes, the *RT–11 Installation Guide*, and your distribution kit for the contents of the current distribution kit.

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

- **New Files**

  The following files were added to the RT–11 software distribution kit. See the *RT–11 Installation Guide* for a description of each file's contents:

  ```
  ATX.SYS
  DBGSYM.SAV
  INDEX.IDX
  INDEXA.IMG
  INDEXB.IMG
  INDEX.SAV
  INDEXX.SAV
  MDUP.AI
  MDUP.MU
  MONMRG.SAV
  MSB.COM
  MSCPCK.SAV
  MU.SYS
  MUB.COM
  MUX.SYS
  NU.MAC
  NUX.SYS
  SDH.SYS
  SDHX.SYS
  SDS.SYS
  SDSX.SYS
  SPOOL.SAV
  SYSTEM.MLB
  TU.MAC
  UBX.SYS
  UM.MAC
  XMEL.ANS
  ```

- **New File Types**

  The following new file types are recognized by RT–11:

  ```
  JBK   File type of most recently superseded version of a KED/KEX
        journal file (journal file equivalent of the BAK file type)
  JOU   Default file type for KED/KEX journal file
  KED   Default file type for KED/KEX macro or initialization file
  ```

- **Removed Files**

  The following files were removed from the RT–11 software distribution kit, are no longer supported, and are available from the DECUS library:

  ```
  CR.MAC, CR.SYS
  CT.MAC, CT.SYS
  DD.MAC, DD.SYS
  DP.MAC, DP.SYS
  DS.MAC, DS.SYS
  DT.MAC, DT.SYS
  K52.SAV
  PC.MAC, PC.SYS
  PD.MAC, PD.SYS
  RF.MAC, RF.SYS
  ```

  The following files were removed from the RT–11 software distribution kit, are not supported, and are not available from the DECUS library:

  ```
  GETSTR.FOR
  INDEX.IMG  (Old single (nonsplit) INDEX data file
             replaced by INDEXA.IMG and INDEXB.IMG)
  PUTSTR.FOR
  RT11PI.SYS  (Automatic installation monitor for Professional 300
               series processors)
  VTHDLR.OBJ
  VTMAC.MAC
  ```

- **Source Kit**

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

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

Processors

Support was added for the following processors:

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

  The T–11 processor chip is supported by the RT–11 foreground/background (FB) monitor in the SBC–11/21 PLUS single-board computer configuration. The original SBC–11/21 is also supported by the FB monitor.

- **F–11 Microprocessor Based Boards and Systems**

  The PDP–11/23 PLUS is supported by RT–11 V5, including full 22-bit addressing (up to 4M bytes) under the extended memory (XM) monitor.

- **J–11 Microprocessor Based Boards and Systems**

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

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

  RT–11 supports the MicroPDP–11/53 (KDJ11–D) processor system.

Storage Devices

Support was added for the following mass storage devices:

- **RA60 Disk** (DU)

  The RA60 disk is a 204M-byte (formatted) removable disk supported by the MSCP disk class handler, DU.

  Because the RA60 disk contains more than 64K blocks, it must be divided into multiple 64K-block partitions. When running under RT–11, each partition operates as a separate disk.

- **RA80/RA81/RA82 Disks** (DU)

  RA80

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

  Because the RA80 disk contains more than 64K blocks, it must be divided into multiple 64K-block partitions. When running under RT–11, each partition operates as a separate disk.

  RA81

  The RA81 is a 406M-byte fixed hard disk. Because the RA81 contains more than 64K blocks, it must be divided into multiple 64K-block partitions. Each partition operates as a separate disk.

  By default, you can separate the RA81 into up to 8 partitions, letting you directly address up to a maximum of 262M bytes. To access the remainder of the disk, you should perform a system generation and request extended device-unit support, as described in the *RT–11 System Generation Guide*.

  You can also use the JREAD and JWRITE subroutines to read and write nonfile-structured data to the rest of the disk.

  RA82

  RT–11 V5.5 supports the RA82 disk. The RA82 is a 600M-byte fixed hard disk. Because the RA82 contains more than 64K blocks, it must be divided into multiple 64K-block partitions. Each partition operates as a separate disk.

  By default, you can separate the RA82 into up to 8 partitions, letting you directly address up to only 262M bytes. To access the remainder of the disk, you should perform a system generation and request extended device-unit support, as described in the *RT–11 System Generation Guide*.

  You can also use the JREAD and JWRITE subroutines to read and write non-file-structured data to the rest of the disk.

- **RC25 Disk** (DU)

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

- **RD31/RD32/RD50/RD51/RD52/RD53/RD54 Disks** (DU)

  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 must be divided into multiple 64K-block partitions. Each partition operates as a separate disk.

  All disks can be used as system or data volumes.

- **RX33 Diskette Drive** (DU)

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

  FORMAT supports formatting the RX33 diskette. Formatting RX33 diskettes requires an RX33 diskette drive and an RQDX3 controller of at least microcode revision 2.

  CAUTION

  Do not format RX50 diskettes in the RX33 drive

  RX50 diskettes do not contain 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** (DU)

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

  RX50 diskettes on Q-bus and UNIBUS processors are MSCP (mass storage communication protocol) devices supported by the MSCP disk class handler, DU.

- **TK25 Magtape** (MS)

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

- **TK50 Magtape** (MU)

  RT–11 supports the TK50 magtape drive.

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

  The RT–11 distribution kit is available on TK50 magtape. You can install RT–11 from the TK50 magtape, using the automatic installation procedure.

- **TS05 (TSU05/TSV05) Magtapes** (MS)

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

  Data is recorded on an industry-standard 1600 bpi phase-encoded 9-track 1/2-in format magtape. The TS05 can store up to 28M bytes (in formatted 2K records) on a standard 10 1/2-in, 2400-ft reel of tape. TS05 tape drives can accept three tape reel sizes: 7 in, 8-1/2 in, and 10-1/2 in.

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

- **TU81+ Magtape** (MU)

  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).

- **VT300 and 400 Series Terminals**

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

Other Hardware

Support was added for the following hardware:

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

Automatic Installation Procedure

Support was added for an automatic installation and verification procedure. This procedure installs RT–11 by conducting an interactive dialog at the console terminal.

The following changes, additions, or new information applies to the automatic installation procedure.

- The installation verification procedure runs the IND control file, VERIFY.COM, rather than IVP.COM. Therefore, to run the installation verification procedure, issue the following command:

  ```
  .RUN IND VERIFY RET
  ```

  VERIFY.COM then runs the installation verification program, IVP.MAC.

- The terminal identification portion of the automatic installation verification program (TERMID.MAC) has been updated. TERMID.MAC recognizes the existence of current terminals. TERMID.MAC is assembled, linked, and run as part of the automatic installation verification procedure.

- STARTA.COM has been changed to use the .STRUCTURE directive in IND to determine the file structure of a specified working disk.

  If the .STRUCTURE directive finds a recognized foreign or an unknown file structure, the procedure warns you and gives you the opportunity to continue or stop the procedure.

  If the .STRUCTURE directive finds an RT–11 file structure, the volume is examined for files. If files are found, the installation procedure warns you and gives you the opportunity to continue or stop the procedure.

  See the *RT–11 IND Control Files Manual* for a description of the .STRUCTURE IND directive.

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.

The following changes were made to the system generation procedure:

- The system macro definition library, SYSTEM.MLB, is assembled with other source modules to generate RT–11 monitors and device handlers. Therefore SYSTEM.MLB must reside on the source (SRC) device when a system generation is performed. The SYSGEN dialog contains a prompt indicating this requirement.

  Including SYSTEM.MLB in the assembly can, depending on the number of invocations and expansions of its macros, require enlarging the size of the macro work file. Changing the work file size is described in the *RT–11 Installation Guide*.

- The procedure to build monitors links the monitor bootstrap (BSTRAP) and resident monitor (RMON) in two separate stages. This operation is transparent and embedded within the SYSGEN procedure. You should, however, be aware of it.

  This change is not relevant to customizing monitors by using customization patches, as customization patches use file address values. You should, however, use the SIPP /A option whenever you customize a generated (or distributed) monitor.

- Because the VT11 and VS60 graphics modules have been removed from the software distribution kit, support for the VT11 and VS60 has been removed from SYSGEN. The *GRAPHICS OPTIONS* section of the SYSGEN dialog is therefore deleted.

- 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.

- Previously, if SYSGEN needed to load a device handler, that device handler was not unloaded at the end of the SYSGEN procedure. SYSGEN now unloads any device handlers that it loaded.

- 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 hardboot 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.

  Digital recommends 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.

  If you are performing a system generation for an unmapped monitor, SYSGEN asks if you want DU to perform the bad-block replacement. IF you are using an RC25 or the RA series disks under the unmapped monitors, you should request DU bad-block replacement.

- SYSGEN displays the default CSR addresses and vector addresses for each port you want the MSCP disk class handler (DU) to support.

- In system generation dialog questions that ask for CSR and vector addresses, the default values are displayed in parentheses. Those default values are determined by the CSR and vector addresses that you entered in response to previous questions. The standard defaults are displayed, if you accepted the defaults in response to previous questions.

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 2100.

- **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.

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.

The following list summarizes the changes made to RT–11 commands, most of which are DCL ones:

- **ABORT**

  The ABORT command was created to let you abort from the shared console a foreground or system job assigned to a private console terminal with the FRUN or SRUN /TERMINAL:n option. The ABORT command cannot abort a job with SCCA in effect.

- **BACKUP**

  The BACKUP command was created and then greatly improved as a way to store information in savesets.

- **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. With RT–11 file-structured device copy operations, the bad-block replacement table on the RL01, RL02, RK06, or RK07 output device is automatically preserved.

  — The /DELETE option no longer inhibits the copying of protected input files, but a warning message is still generated as each protected file is copied.

  - When you use the /DELETE option with the COPY command, you are no longer prompted for confirmation of the deletion. If you want this prompt, you must use the /QUERY option.
  - When you copy files, the protection status of the output file will be the same as the protection status of the input file, unless you use the /PROTECTION or /NOPROTECTION option.
  - You no longer need to use the /SYSTEM option to copy SYS files unless you use wildcards in the input file type.
  - You can use the /VERIFY option for files as well as for entire volumes. The /VERIFY option is invalid with the /ASCII and /BINARY options.

- **DELETE**
  - The DELETE command no longer prompts you for confirmation, unless you use wildcards in the file specification.
  - You no longer need to use the /SYSTEM option to delete SYS files unless you use wildcards in the input file type.

- **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 /BEGIN option is position independent. The command DIRECTORY /BEGIN is valid.
  - You can use the /VOLUMEID[:ONLY] option with /INTERCHANGE to print the volume ID of an interchange diskette.

- **DISMOUNT**

  The DISMOUNT command is no longer coupled with the LD (logical-disk subsetting) utility. Rather, DISMOUNT is a general DCL command, applicable to any runnable handler that supports the DISMOUNT command, such as LD. The runnable handler must contain code to process the DISMOUNT command when the handler is chained to by KMON. DISMOUNT support is generated by specifying *stat2=HS2.MO* for the .DREST macro *stat2* parameter when the runnable handler is built.

- **EXECUTE**

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

- **FORMAT**

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

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

- **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.

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

- **LINK**

  The *value* parameter for the /BOUNDARY option is optional (LINK /BOUNDARY[:value]). Not specifying the value parameter causes LINK to prompt for up to eight separate PSECT boundary addresses. You terminate the prompt sequence by pressing RET with no specified value.

- **MACRO**

  The /PASS:1 option was eliminated.

- **MOUNT**

  The MOUNT command is no longer coupled with the LD (logical disk subsetting) utility. Rather, MOUNT is a general keyboard monitor command, applicable to any runnable handler that supports the MOUNT command, such as LD. The runnable handler must contain code to process the MOUNT command when the handler is chained to by KMON. MOUNT support is generated by specifying *stat2=HS2.MO* for the .DREST macro *stat2* parameter when the runnable handler is built.

- **PRINT**

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

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

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

- **PROTECT**

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

- **R**

  For RT–11 V5.5, the R command is changed to be identical to the RUN command, with one exception: the default location for the utility run by the R command is still the SY device; and the default location for the utility run by the RUN command is still the DK device.

  The R and RUN commands are alike in two ways:

  — They use the bitmap in the core control block (CCB) to load jobs (previously the R command did not use that bitmap).

  — With each command, you can specify any random-access device as the location of the utility to be run.

- **RECALL**

  Requires that you run the Single-Line Command Editor under the XM monitor. RECALL displays and retrieves commands you previously issued. Issuing the RECALL command with no option recalls the previous command.

- **RENAME**

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

- **REMOVE**

  When you are 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. Therefore, you can use the RUN command to execute virtual jobs.

- **SET**

  — The SET SL conditions INSERT and REPLACE are removed from the documentation. Use 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**

  — The SHOW and SHOW SUBSET commands indicate the default device, if present, by displaying an asterisk next to the device handler or logical disk unit. (A default device is created by using the ASSIGN dev * command.)

  — The 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 that, because of the handler characteristics or your system configuration, cannot be installed in some manner on your system. The three possible displays are:

Display	Meaning
not BSTRAP installable	Handler must be installed by INSTALL command
not KMON installable	Handler must be installed at system boot
not installable	Handler cannot be installed on this system

- SHOW SUBSET displays any logical name assigned to a logical disk. For example, if you mount a 1000 block file DU4:FOO.DSK on logical device unit LD0 and assign it the name FOO, SHOW SUBSET displays:

  ```
  LD0 is DU4:FOO.DSK[1000.] = FOO
  ```

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

- SHOW DEVICES accepts the argument MU (SHOW DEVICES:MU), which displays MU status information for the PORT and UNIT of each installed TMSCP controller in the same manner that the DU argument displays MSCP status information. (Partitioning is invalid with magtape devices.)

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

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

- The shortest valid abbreviation for COMMANDS is COM.

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

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

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

Extended unit support
SET CLOCK and SET FORTRA conditionals
Total amount of memory
Active command file processor: KMON or IND
SL status: ON, OFF, or KMON
Default editor for EDIT command
Status of .SCCA support and the .SCCA flag when enabled
EAE (extended arithmetic element) hardware option

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

— The SHOW DEVICES command lets you obtain information about a specific device by using the command SHOW DEVICE:xx. The variable xx represents the 2-letter permanent device name. The CSR and vectors for each device displayed are also given.

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

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

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

For RT–11 V5.5, the format of the SHOW QUEUE display for SPOOL is changed. The following example display shows two output devices, LS0 and LP0, attached to SPOOL, with LS0 printing and LP0 idle:

```
Unit  Device  Status
SP0:  LS0:    ACTIVE,      56 blocks spooled
SP1:  LP0:    IDLE

  944 Free blocks in workfile
```

— 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. If the SPOOL utility is running, SHOW QUEUE requires RESORC.SAV be on device SY. If the QUEUE utility is running, SHOW QUEUE requires RESORC.SAV and QUEMAN.SAV be on device SY.

- **SRUN**

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

- **TYPE**

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

- **UNPROTECT**

  Removes protection from a file so you can delete it.

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.

These changes were made:

- **BATCH**

  The RT–11 environmant must be SET RUN NOVBGEXE before entering BATCH. Under BATCH, the following restrictions apply to DCL commands that might be executed in RT–11 Mode:

  - Do not Set RUN VBGEXE
  - Do not invoke VBGEXE
  - Do not use V or VRUN
  - Do not run I-D space programs

- **BINCOM**

  - You 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. The *Introduction to RT–11* contains tutorial information about backing up data, using BUP.

  The following improvements were added to BUP:

  - DL (RL01/02) and DM (RK06/07) volumes require a two-step initialization process before being used as backup media. First, such volumes must be initialized by DUP, using the INITIALIZE/REPLACE command. That command creates the software bad-block replacement table structure on the volume. Second, such volumes must be initialized by BUP, using the BACKUP/INITIALIZE command. Bad-block replacement on those volumes is then performed transparently by the device handler during backup operations.
  - You can back up a smaller disk volume to a larger disk volume or to a disk volume of the same size. BUP no longer returns the error message, ?BUP-F-Enough space on one volume - use PIP.
  - Previously, RT–11 did not print any warning message if you restored a device image to your system device. Doing that can cause an uncoupling of RMON and the system files and could corrupt your system device. BUP now displays the following warning when you issue the command BACKUP/RESTORE /DEVICE (BUP /X/I) and your system device is the output volume:

```
Restoring over SYSTEM volume!
SY:/Restore; Are you sure?
```

- BUP supports the following new or changed options:

/G	Inhibits the bad-block scan on disk output volumes.
/M	Inhibits rewinding magtape before appending next saveset to that magtape.
/S	Indicates the saveset containing a file you want to restore or the saveset from which you want to obtain directories.
/V:ONL	Valid only when used with /X option. Causes a comparison between named saveset and restored file or volume.
/W	Suppresses various informational messages.
/Y	Inhibits prompting for various responses otherwise required from the terminal.
/Z	Must be specified to initialize a magtape; is automatically included with disk operations.

- BUP can verify data transferred to or from a device. Use the /V option for this purpose. BUP prints an error message if the input data does not match the output data.

- **CONFIG**

  The following two options are added:

Option	Function
/D[:yes][:no]	With no optional parameter, causes CONFIG to return an error message in addition to setting bits in USERRB. You can assume success if no error message is returned. /D:no turns off /D:yes.
/F:addr	Determines whether file location addr exists.
/T	Determines if an MSCP device unit contains fixed or removable media and whether that media is available.

- **DBG–11**

  Support was added for DBG–11, a symbolic debugging package that lets you interactively debug a MACRO–11 program. See the *DBG–11 Symbolic Debugger User's Guide*.

- **DIR**

  - Previously, when you performed a directory (DIR command) on a magtape, you received the directory listing, the magtape rewound, and you then received the trailer information (number of files). You now receive the directory listing with the trailer information while the tape is rewinding. You receive the monitor prompt after the magtape has rewound.

    The change above also applies when the directory information is directed to your printer (DIRECTORY/PRINTER command).

— RT–11 V5 includes the following additional directory program (DIR) options:

/T	Includes in the directory listing only those files that are protected against deletion.
/U	Includes in the directory listing only those files that are not protected against deletion.

- **DUMP**

  The following changes apply to dumping magtape records:

  — When the record is shorter than 256 words, DUMP displays only those words actually read, rather than using a fixed length 256 word format.

  — When the record is longer than 256 words, DUMP displays only the first 256 words and displays (TRUNCATED) in the status line that precedes the displayed data.

- **DUP**

  — DUP supports a /J (/IGNORE) option that must be coupled with /I (/DEVICE). COPY/DEVICE/IGNORE causes any errors returned by a bad block on the input or output device to be ignored. The bad block on the device that returns the error and a corresponding block on the other device are not copied. An error message displays which device (input or output) contains the bad block and the bad-block number.

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

  — DUP consults internal tables when processing commands, except in the case of the COPY/DEVICE command as follows:

  DUP no longer consults internal tables to determine if a device supports a software (DL and DM type) bad-block replacement table. Handlers that support such a replacement table should include the .DREST macro and specify the *replace* parameter.

  DUP also no longer consults internal tables to determine if a device returns an extra error word on absolute read/write special functions (377 and 376). Handlers for devices that return such an extra error word should include the .DREST macro and specify the *DVM.DM* argument for the *mod* parameter.

  DUP continues to consult internal tables to determine if a device is a magtape.

  — Warning and error messages for DUP/I/Y (COPY/DEVICE/NOQUERY) and DUP/I/F/Y (COPY/DEVICE/FILES/NOQUERY) operations have been changed. The changed messages are listed in the *RT–11 System Message Manual*.

  — Informational system prompts displayed during DUP/I (COPY/DEVICE) and DUP/I/F (COPY/DEVICE/FILES) operations have been changed.

  — The /H option when combined with the /K or /B option is valid (with certain restrictions) to clear bad blocks caused by soft errors on MSCP class devices.

The /H option causes RT–11 to read each block and, if it encounters an error, write the block out and then read it again. That forces the controller to clear (make usable) blocks that return a soft error. Use /H/K or /H/B only with blank media or with media you have just backed up. This functionality is available only as a CSI-level command and is not implemented as a KMON command option.

The distributed RT–11 supports the /H option for RD31, RD32, RD51, RD52, RD53, and RD54 MSCP devices. You must perform a SYSGEN and request DU handler bad-block replacement support to use the /H option with RC25, RA60, RA80, RA81, RA82, and RA9* devices.

The /H option when combined with the /K or /B option is recommended only for MSCP class devices.

- When you use the /I option to copy a larger volume to a smaller one, DUP asks for confirmation before copying the volume.
- You can abort a /W (WAIT) operation.
- It is no longer necessary to customize DUP to use variable-size volumes. Instead, a handler attribute VARSZ$ can be set to indicate a variable-size volume. Use the stat argument for the .DRDEF programmed request to set VARSZ$ (bit mask 400). Then, you can use the .DSTATUS programmed request to determine whether the handler supports variable-size volumes (and .SPFUN 373).

- **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**

  - For RT–11 V5.5, the maximum number of devices for which you can generate error logging support (ERL$U=nn) is changed to 22(decimal), and each statistical entry requires 11(decimal) words. By default, the system generation procedure continues to generate support for 10(decimal) device units.
  - RT–11 V5.5 provides (T)MSCP error logging for MSCP devices. See the *RT–11 System Utilities Manual* and the *RT–11 Device Handlers Manual* for information.
  - Calls to the error logger from within user-written handlers no longer need to be serialized. For RT–11 V5.5, the error logger is reentrant and calls to the error logger no longer need to be issued only after a .FORK call.
  - If you attempt to examine the contents of the error log file before running ELINIT, ERRLOG returns the error message, ?ERRLOG-W-Invalid message received. You must run ELINIT before examining the error log.
  - The block number returned in the device error report is in decimal radix. Previously, the block number was in octal radix.

- The RT–11 V5 error logger reports the number of retries for a single error and the final status of the operation (success or failure). The error logger provides separate entries for retries only if the registers differ.
- You can choose to log successful I/O transfers and errors or only errors. Use the SET dd SUCCES command to log successes as well as errors and the SET dd NOSUCCES command to log only errors.

- **FILEX**
  - The default device for all FILEX operations is DK:.
  - Two additional FILEX options are:

/V[:ONL]	Use with /Z and /U[:n] simultaneously to write a volume identification during initialization of an interchange diskette. Use the [:ONL] argument to change an interchange diskette's volume ID without initializing the diskette. Use with /L or /F to list the volume ID of an interchange diskette when obtaining a directory listing.
/W	Initiates the operation but pauses and waits for you to mount different volumes.

- **FORMAT**
  - FORMAT supports extended device units by recognizing device units greater than 7. For example, the command FORMAT D10 is valid and recognized by FORMAT.
  - FORMAT pattern #12 (bit 11 set) is 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 occurs again, get a new copy of FORMAT.SAV and retry the operation. If the error persists, submit an SPR to Digital.
  - Attempting to format a disk that is not mounted returns the error message, ?FORMAT-F-Device not ready. If you receive that message and your disk is not mounted, mount your disk and be sure it is up to speed.
  - 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.
  - If you try to format a volume that contains protected files or try to format a volume while a foreground job is loaded, FORMAT warns you and asks you to confirm the operation.
  - You can abort a /W (WAIT) operation.
  - The table of verification bit patterns has been increased to 16 patterns. The last 4 of the 16 patterns are reserved for future use.
  - Formatting of devices at nonstandard addresses is supported and will occur automatically, based on the CSR location specified in the device handler. For

DU devices, any supported number of controllers is allowed. For all other devices, only a single controller is allowed.

Even though FORMAT accesses the hardware directly for some devices, FORMAT uses the handler file contents to determine the CSR address.

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). The indirect control-file processor executes control files. See the *RT–11 IND Control Files Manual*. Note the following:

  — For RT–11 V5.5, IND allows the opening of LP, LS, SP, and file-structured magtape handlers, for write operations (.OPEN directive), and the opening of any directory-structured device for read operations (.OPENR directive).

  This, along with support for .TESTDEVICE and .VOL on magtape devices, is the only support IND provides for special directory structured devices. However, this support does allow access to such devices from IND control files.

  — For RT–11 V5.5, IND provides a character/numeric symbol conversion enhancement to the existing symbol value substitution functionality. IND allows symbolic expression of nonprintable characters, allowing the display of nonprintable characters in a printable format. The conversion applies only between character and numeric symbols (not to logical symbols) and does not require the .ENABLE SUBSTITUTION directive.

  — Previously, the .READ directive incorrectly enforced the obsolete 80-characters-per-line restriction; that restriction was removed. Now, .READ also correctly allows 132 characters per line.

  — The .TESTDEVICE and .VOL directives support the <FILERR> special symbol. The <FILERR> special symbol returns a new code, 377, which lets you intercept some read-related errors and perform error recovery.

  — IND dynamically determines whether RT–11 is running the XM monitor. If IND is running under the XM monitor, a portion of IND permanently resides in extended memory. That portion is displayed by the SHOW MEMORY command. Use the REMOVE IND command to eliminate that portion of IND in extended memory.

  — IND dynamically allocates a region in extended memory for its symbol table when running RT–11 under the XM monitor. You can suppress the dynamic allocation, using a customization patch located in the *RT–11 Installation Guide*.

- The .STRUCTURE directive has been added to IND to determine the file structure of a specified file-structured device from boot block information.
- The .TESTDEVICE directive has been modified to include ATT or NAT (attached or not attached) in field nine of special symbol <EXSTRI>.
- The .ONERR directive has been modified to include processing integer under /over flow conditions.
- The .SETL directive has been modified to support the .AND, .OR, and .NOT logical test directives. When you use the .SETL directive, the logical symbol you specify is set to the value represented by the logical expression.
- The following operating modes have been added to the .ENABLE and .DISABLE directives. (These operating modes also apply to the .IFENABLED and .IFDISABLED directives.)

```
.ENABLE/.DISABLE ABORT
.ENABLE/.DISABLE CONTROL-Z
.ENABLE/.DISABLE TYPEAHEAD
```

- If the .ENABLE TIMEOUT directive is issued but the system does not include timer support, IND assigns special symbol <EXSTAT> the value 0, for warning, instead of printing an error message.

- **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*.

- Initialization files to give you certain operations and characteristics in every work session
- Ability to run the KED editor and manage a complete editing session from an indirect command (.KED) file
- A journal feature that allows the recovery of your work in the event of a user error, unexpected power-down, or similar malfunction
- Expanded macro capabilities, including the definition of more than one macro and the storage and retrieval of macros in auxiliary files and KED command files
- The ability to enter multiple KED commands without reentering the GOLD command sequence
- No movement of the cursor when a search target cannot be found
- Expanded HELP displays that describe the new commands
- Optional suppression of full screen HELP displays

- — The ability to disable the selection operation in the FIND function

  The following changes applied to the keypad editor prior to Version 5.5:

  When you are simultaneously using two KED/KEX editors and you enter (and keep open) command mode on the first editor and then exit or quit the second, you can reenter the first editor. You can then press CTRL/W to repaint the screen, enter command mode to exit or quit editing that file, or use a special function key to perform other functions.

  See the *PDP–11 Keypad Editor User's Guide* for complete information.

- **LET**
  - — LET supports substitution strings of up to 32(decimal) characters. As before, you can define up to 5 symbols concurrently.
  - — On keyboards that contain function keys (F1 through F20), LET supports defining F6 through F10, F14, and F17 through F20, as symbols for string substitutions. For example, the following LET substitution defines the F7 function key as the string *MACRO / LIST / CROSSREFERENCE*:

    ```
    .LET F7=MACRO/LIST/CROSSREFERENCE RET
    ```

- **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 in a library module (and therefore resolve) any absolute global symbol, 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. The LIBR /A option is generally not used because it forces all global and absolute global symbols into the library module directory, making the directory quite large. It is generally better to associate any absolute global symbol with at least one relative global symbol.
  - — A problem with LINK's CSI processing has been corrected, so that with RT–11 V5 you can link an increased number of modules at one time.
  - — 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**
  - — RT–11 supports MACRO–11 V5.5.

- 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. However, the MDUP programs support VM and devices controlled by the DU handler. If you need to create MDUP support for a device not supported in the distributed MDUPs, see the *RT–11 Installation Guide*.
  - The magtape utility (MDUP) includes a /V:n option. The /V:n option checks that extended memory contains at least the number (n) blocks you specify. If extended memory contains n blocks, MDUP executes commands until another /V is encountered, whereupon MDUP stops executing the commands.

    If extended memory contains less than n blocks, MDUP does not execute commands until another /V is encountered, whereupon MDUP starts executing the commands.

- **ODT**

  A customization patch in the *RT–11 Installation Guide* lets you create a monitor-independent ODT debugger; that is, an ODT that does not require that the operating system be loaded in memory.

- **PAT**

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

- **PIP**

  - PIP performs a .RCTRLO request between files, when multiple files are copied to the terminal. This functionality lets you use CTRL/O to selectively stop the display of files on the terminal screen. Executing a CTRL/O discards output until another CTRL/O is executed, or until the beginning of the next file is reached, whereupon output to the terminal resumes.
  - If a PIP command line includes file transfers from magtape, PIP performs all file transfer operations requested on the command line in the order in which the files appear on the volume rather than the order in which they are specified in the command line.
  - The /C option accepts the [:date] argument. Use /C[:date] to include files of a certain date in the operation you specify.
  - You can abort a /E (WAIT) operation.
  - In RT–11 V5, you can use the /F (PROTECTION) and /Z (NOPROTECTION) options alone or for copy operations as well as for rename operations. It is no longer necessary to use /R (/RENAME) with /F or /Z.
  - The /Q (QUERY) option is no longer the default when deleting files, except when you include wildcards in the file specification.

- The /T (SETDATE) option accepts the [:date] argument, so you can assign files dates other than the current system date.
- The /Y (SYSTEM) option is necessary only when you specify wildcards in the input file types.
- The following PIP options are new:

/H	(VERIFY) Verifies that the output file matches the input file after a copy operation. This option is invalid with /A (ASCII) and /B (BINARY).
/I[:date]	(SINCE) Includes only those files created on or after the specified date.
/J[:date]	(BEFORE) Includes only those files created before the specified date.
/V	(MULTIVOLUME) Copies files from one input volume to two or more smaller output volumes.
/X	Prints an informational (I) message rather than a fatal (F) message when PIP cannot find a file specified in the command line.

- **Queue Package**

- The PRINT command is affected when you run SPOOL and QUEUE together.
- QUEUE appends a form-feed character <FF> to the end of each copy of a queued file, whether the output is to a disk, a serial line printer, or a parallel line printer.
- The SHOW QUEUE command is processed by the RESORC /Q option rather than by the QUEMAN /L option. However, the /L option remains for the sake of compatibility.
- When QUEUE sends a job consisting of more than one input file to an RT–11 file-structured device, QUEUE copies each input file to a separate output file with the same file name and type. The job name is printed in the JOBNAME field of the banner page. In RT–11 V4, all input files in the same job were concatenated into one output file with the file type .JOB.
- Input files are protected from deletion while QUEUE is copying them to the output device.
- The default number of banner pages printed when you use the /H command is determined by the number of banner pages you set as the default with the /P command.
- To save time, magtape input devices for QUEUE operations no longer rewind between files.
- QUEUE's work file is SY:QUFILE.WRK. In RT–11 V4, the work file was DK:QUFILE.TMP.
- The following are additional QUEMAN options:

/C[:date]	(DATE) Prints only those files with the specified creation date.
/I[:date]	(SINCE) Prints only those files created on or after the specified date.

/J[:date] (BEFORE) Prints only those files created before the specified date.

/Q Requests confirmation for each file to be included in the operation. QUEMAN prints the name of each file and pauses; you must respond Y for each file you want to include.

/W Prints on the console a log of the files included in the operation.

/X Allows QUEMAN to continue processing instead of halting when it cannot find a file specified in the command line.

- **RESORC**

  — RESORC supports a new /R option, which displays the status of UNIBUS mapping registers (UMRs); equivalent to SHOW UMR.

  — RESORC option /Z displays the status of the CLOCK and FORTRA conditions.

  — RESORC options /D and /S indicate the default device, if present, by displaying an asterisk (*) next to the device handler or logical disk unit. (A default device is created by using the ASSIGN dev * command.)

  — RESORC option /X 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*.

  — RESORC option /D displays installability information for any device handler that, because of the handler characteristics or your system configuration, cannot be installed in some manner on your system. The three possible displays are:

Display	Meaning
not BSTRAP installable	Handler must be installed by INSTALL command
not KMON installable	Handler must be installed at system boot
not installable	Handler cannot be installed on this system

  — RESORC option DU:/D displays additional DU status information. The port display indicates an invalid port by displaying an asterisk (*) before the port number. For example, port = *n, where n = 0,1,2,3, indicates that port n was not installed. If RESORC option DU:/D displays port = n, where n = 0,1,2,3, then port n is installed.

  — RESORC includes a /V option. The /V option displays the release and version numbers for any module in the RT–11 distribution kit.

  Use the /V option to supply the release and version numbers for any RT–11 modules quoted in an SPR (software performance report) submission.

  In the following example, the /V option reports the release and version numbers for the LS handler:

```
*LS.SYS/V

Release = V05, Version(s) = 3
```

- RESORC options /A, /H, and /Z display processor support for the EAE (extended arithmetic element) hardware option.
- RESORC options /A and /X display global regions in extended memory.
- RESORC options /A and /X display a fourth column in the extended memory map that indicates the type of region being listed.
- The /O option shows if SYSGEN special feature FPU (floating-point unit) support has been selected.
- RESORC displays the following additional processor types:

  ```
  PDP-11/53
  PDP-11/83
  PDP-11/73A
  PDP-11/73B
  PDP-11/84, UNIBUS
  KXJ11-CA
  ```

- The "KT11 Memory Management Unit" display reads "Memory Management Unit".
- The /A (ALL) option provides information about the total amount of memory on the system, logical disk subsetting assignments, and organization of physical memory.
- The /C (CONFIGURATION) option provides status information for SET KMON [NO]IND, SET EXIT [NO]SWAP, SET EDIT, SET SL ON/OFF/KMON, extended device-unit support, and the global .SCCA flag.
- The /D (DEVICES) option accepts the optional argument dd (dd:/D), where dd represents the 2-letter permanent device name. You can use the argument dd to obtain information about a specific device.
- The /H option includes the total amount of memory on the system.
- The /O option also reports whether global .SCCA support was chosen during system generation.
- The RESORC utility includes the following options:

  /Q — Lists the contents of the queue for QUEUE or SPOOL, depending on which is running. The SHOW QUEUE keyboard command is performed by this option.

  /S — Displays information about logical disk subsetting assignments.

  /X — Displays information about the organization of physical memory: where jobs and handlers are loaded and where KMON and the USR will reside.

- **SETUP**

  Support was added for SETUP (Hardware Setup Utility), which lets you choose operating characteristics for video terminals, printers, and some system clocks. Note the following:

  — SETUP supports the options NORMAL and [NO]REVERSE.

  — You can specify the color amber for color1 or color2 in the SETUP [color1] [ON color2] command. You can create and store the exact shade of amber you want, using the SETUP SETCOLOR command.

  — SETUP supports two additional destinations for only the LANG:code mode: KB and VIDEO. The LANG:code mode is not supported on a UNIBUS or Q-bus processors.

  — SETUP supports the construction of display colors from primary colors, using the SETCOLOR mode. SETCOLOR mode is not supported on UNIBUS or Q-bus processors.

  — SETUP supports the NRC (national replacement character set). The NRC is not supported on UNIBUS or Q-bus processors. Support for each NRC keyboard is selected with the SETUP LANG:code command. SETUP also includes the following modes for more complete NRC support: COMPOSE, DATA, RETAIN, and TYPE. NRC support includes the use of Dead Diacritical keys with various NRC keyboards. No valid Dead Diacritical keys appear on the USA keyboard.

  — SETUP supports the COMPOSE CHARACTER key with the modes COMPOSE and NOCOMPOSE. The COMPOSE CHARACTER key is not supported on UNIBUS or Q-bus processors. COMPOSE mode lets you use combinations of two keyboard keys to construct characters from the various national replacement character (NRC) sets.

  — SETUP supports 480INTERLACE. 480INTERLACE is not supported on UNIBUS or Q-bus processors.

  — SETUP supports [NO]INTERLACE mode. [NO]INTERLACE mode is not supported on UNIBUS or Q-bus processors. The shortest valid abbreviation for INTERLACE is INT.

  — SETUP supports functionality for laser printers with the LANDSCAPE and PORTRAIT modes.

  — SETUP RESET returns the text and background colors to the settings that were set with the most recent SETUP SAVE command.

- **SIPP**

  — Because of changes made to the way monitors are linked and the way monitor map files are built, you should always use the /A option whenever you use SIPP to perform a customization on a distributed or generated monitor.

- When SIPP is used to patch a file, the creation date of the patched file is changed to the current system date. If no modifications are made, the date remains unchanged.
- When using SIPP to create an indirect command file, the command file contains the command RUN SIPP rather than R SIPP. RUN SIPP lets you run the command file from a volume 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.

  These are some of the changes:

  - SL supports the VT300 series terminals when they are set to "VT300-7bit" mode.

    The SET SL ASK command recognizes VT300 series terminals, and when the terminal is a VT300, that command displays the following message:

    ```
    ?SL-I-Supporting as VT102, terminal is - Generic VT300
    ```

  - The following three SET conditions are no longer supported:

    ```
    SET SL VT101
    SET SL VT62
    SET SL VT52
    ```

    In place of SET SL VT101, use the SET SL VT100 command.

  - SL no longer supports VT52 and VT62 terminals.
  - SL can edit single characters in two modes: INSERT mode and REPLACE mode. INSERT mode lets you place a character in the command line by shifting to the right all characters beyond that character. REPLACE mode lets you place a character in the command line without shifting any characters. SL is distributed in INSERT mode. You can alternate between INSERT and REPLACE mode by pressing CTRL/A. SL returns to INSERT mode at each command line.

    You can change the default mode for single-character editing from INSERT to REPLACE by performing a customization patch located in the *RT–11 Installation Guide*.

- **SLMIN**

  SLMIN (the minimum functionality Single-Line Command Editor) supports only those functions contained in previous versions of SLMIN. However, changes have been made to SLMIN to make its limited functionality consistent with the current SL. For example, the SAVE and GET SAVE functions use the key sequences PF1 S and PF1 X. Also, you can either press PF1 ▭ to recall the command-before-last or press ▭ twice.

  See the *Introduction to RT–11* for more information on SLMIN.

- **SLP**

  — SLP ignores any characters that precede the start-of-update character (-) in SLP command files. If SLP is unable to find the start-of-update character, SLP prints an error message and returns control to the CSI (indicated by the asterisk prompt).

  — You can update more than one file in a single SLP command file. Type a double slash (//) on a line by itself after the update text for each file. On the next line, type the command line that specifies the next input file to be updated and the command file name (the same command file that contains the update text). Then type the update text on the lines that follow. Type a single slash (/) on a separate line to indicate the end of a series of update texts.

  — The SLP utility includes the following additional options:

/C[:n]	Determines or validates the contents of the SLP input file or the SLP command file. Use /C to determine the checksum of a file. Use /C:n to verify the contents of a file. SLP computes the checksum for the file and compares the checksum to the value you specify for n.
/N	Suppresses the creation of a backup file when SLP updates the input file.

- **SPOOL**

  Support was added for SPOOL (Transparent Spooling Package). The transparent spooler (SPOOL) sends output to the printer. Once SPOOL is running, its operations are transparent. Anytime you send output to a printer, either explicitly by issuing commands (such as COPY and PRINT) or by using commands and options that send output to the line printer by default (such as COMPILE/LIST), SPOOL accepts the output and sends it to the printer.

  These are Version 5.5 changes:

  — For Version 5.5, RT–11 distributes a virtual version of SPOOL:

  ```
  SPOOL.SAV
  ```

  SPOOL.SAV can be run only under the XM monitor, and together with the spooler handler, SPX.SYS, occupies significantly less low memory than SPOOL.REL.

  — SPX.SYS supports up to eight input devices; the handler syntax, SPn, where *n* is 0 to 7, is valid.

  — SPOOL.SAV accepts concurrent printing input from up to eight jobs and can direct files to up to eight output devices. You can concurrently print output from more than one job on the same printer or multiple printers.

  — SPOOL, as part of its startup procedure, locks the USR for approximately 0.5 seconds to ensure that SPOOL can process commands before further commands are allowed.

- For RT–11 V5.5, SPOOL supports a new SET SP EXIT command. Use SET SP EXIT, rather than ABORT, to abort SPOOL, as SET SP EXIT allows SPOOL to synchronize its shut down procedure with the rest of the system.

If you use VM as the work device for SPOOL, you should attach VM to SPOOL. However, that makes VM unavailable for other use. The following series of commands lets you attach VM as the work device for SPOOL but still makes VM available for other use. These commands can be included in your start-up command file; if you include them, be sure to remove or disable any commands you now use to start SPOOL.

Notice in the series of commands that no unit number is coupled with VM for the ASSIGN command. Previous versions of this manual incorrectly included unit number 7 (VM7) with the ASSIGN command. Also, the commands are changed to run SPOOL as a system job, using the virtual version (.SAV) of SPOOL.

```
ASSIGN LS: SO0:
ASSIGN VM: SFD:
SRUN SPOOL.SAV/PAUSE
LOAD LS:=SPOOL
LOAD VM7:=SPOOL
ASSIGN SP: LP:
ASSIGN SP0: LP0:
RESUME SPOOL
```

(VM does not recognize unit numbers. However, specifying a unit number for VM in the LOAD command above lets the RT–11 monitor assign more than one function to VM.)

Other changes made to SPOOL:

- Previously, under the XM monitor, the time-of-day string was not printed on flagpages unless the SP handler was loaded. SPOOL supports time-of-day functionality under XM without requiring that the SP handler be loaded.
- The PRINT command is affected when you run SPOOL and QUEUE together.
- The PRINT/FLAGPAGE:n option, when a value is specified for n, overrides the SET SP FLAG=n command. When no value is specified for n with the /FLAGPAGE:n option, the value for n is set by the SET SP FLAG=n command.

- **SRCCOM**

  - For RT–11 V5.5, the /I[:EXA] and /I:GEN options are added to SRCCOM:

Option	Meaning
/I[:EXA]	The default. Causes a case-sensitive file comparison. The upper and lower case of the same character are treated as a difference.
/I:GEN	Causes a case-insensitive file comparison. The upper and lower case of the same character are not treated as a difference.

  - The syntax of the SRCCOM command has changed to:

```
[[out-filespec][,SLP-filespec]=]old-filespec,new-filespec[/options]
```

The syntax element, *SLP-filespec*, lets you create a differences file and a SLP command file in the same command line. With RT–11 V4, you could create only one or the other. Because you can specify both with the RT–11 V5 syntax in your command line, the /P option has been eliminated.

— You can use wildcards with SRCCOM to compare multiple source files.

- **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.

  TRANSF progressively restores packet size during continued success from transmitting at a reduced packet size. See VTCOM in this section for more information.

- **UCL**

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

  — It is a documented restriction that you cannot call indirect files (@filename) in a command definition. However, the *Introduction to RT–11* describes a procedure you can use to circumvent that restriction.

  — 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.
- Previously, VTCOM and TRANSF reduced the transmission packet size by half for each error that occurred during a data transmission. Transmission errors successively halved the packet size from a maximum of 512 bytes to a minimum of 16 bytes. The packet size could decrease in size but not increase in size. Therefore, momentary transmission problems could significantly lengthen the time required to transmit data.

  VTCOM and TRANSF 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.

Monitors

See the *RT–11 System Internals Manual* for a description of the current RT–11 monitors.

These were the changes to the monitors.

- **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 Region Support in Extended Memory**

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

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

- 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

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.

New and Changed Individual Handlers

- **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.

System Subroutine Library (SYSLIB)

See the *RT–11 System Subroutine Library Manual* for the current descriptions of SISLIB 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
  IASIGN
  ICDFN
  IFETCH
  IFREEC
  IGETC
  IGETSP
  ILUN
  INTSET
  IQSET
  PUTSTR
  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
  ```

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
.DREST (Device Handlers Only)
.DRINS (Device Handlers Only)
.DRPTR (Device Handlers Only)
.DRSPF (Device Handlers Only)
.DRTAB (Device Handlers Only)
.DRUSE (Device Handlers Only)
.FPROT
.GFDAT
.GFINF
.GFSTA
.PEEK
.POKE
.PVAL
.SFDAT
.SFINF
.SFSTA
SOB macro
```

— 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
```

Chapter 3

Chapter 3 Current Restrictions and Corrected Problems

Chapter 3

Current Restrictions and Corrected Problems

This chapter lists the current restrictions, corrected software problems, and corrections and additions for the RT–11 documentation set.

Change bars indicate restrictions added in the V5.6 release and problems corrected for the V5.6 release.

You can use the master index and the on-line help utility, HELP, to find more information on topics described in this chapter.

General Handler Restrictions

The following restrictions apply to all RT–11 V5.6 handlers:

- Handlers that are sysgened with device timeout support must be loaded under mapped monitors.

  This is necessary 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 incorrectly 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 5.6 distributes no handler that is runnable and bootable. Therefore, this restriction does not apply to any distributed RT–11 handler.

- You can hang or crash your system if you unload a device handler that has outstanding I/O requests from a job. You must abort the job that is using that handler before you unload the handler.

General Handler Restrictions

- 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. Digital does not recommend the 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).

Individual Handler Restrictions and Problems

The following restrictions apply to only the indicated RT–11 V5.6 handler.

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. Further, only partition 0 is hardware bootable. All partitions are software bootable.
- 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.
- Digital recommends 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. Setting odd parity ensures that at least the parity bit is set for a null byte. Setting parity to even (NOODDPAR) violates that requirement. Digital controllers can then either lose (fail to distinguish) that null byte or set bit 4, which stores an octal value of 020 with bad parity. The MM and MT handlers cannot distinguish bytes with created parity errors from those with true parity errors, which makes magtapes created by BUP, DUP, and PIP unreadable.

  Also, you cannot perform magtape-to-magtape copies on drives controlled by MM when you have set MM to NOODDPAR. Doing that produces immediate unrecoverable errors, depending on the utility program used.

MU

- Digital recommends you LOAD the MU handler when performing BUP or COPY operations with TMSCP magtape devices.

NQ and NU

- You must explicitly load the NQ and NU handlers, using the LOAD command. You cannot fetch those handlers.
- 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 if you are running the single-line (SL) command editor. You must explicitly SET SL OFF if you want to set your 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

- You cannot 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).
- The VM handler (under unmapped monitors) cannot be used on a PDP–11/23 processor with MSV11 memory, configured for a 2K-word I/O page, unless you perform customization patch 2.6.20, described in the *RT–11 Installation Guide*.

XL

- You should explicitly load the XL handler, using the LOAD command. You should not fetch this handler. If VTCOM is run as a system job, you must load it.

Monitor Restrictions

The following restrictions apply to RT–11 V5.6 monitors:

- The RT–11 operating system serializes access to the USR between jobs. It does not serialize access within a job, in particular it does not serialize between the mainline code and completion code. If the mainline code and completion code both issue USR requests, a completion routine request may interrupt a mainline request and cause unpredictable errors.

  Digital recommends that you do not issue USR requests in mainline and completion routines in the same program, unless your program design precludes a completion routine from issuing a request while a request from the mainline is being processed.

- Do not assign unit numbers to the system device logical name SY or the default data device logical name DK. Do not, for example, use the designations SY3 or DK1 (or SY0 or DK0).

- Because of instruction processing inherent in the DCJ11 microprocessor CPU, all processors that contain that CPU may indicate an incorrect PC value in the message displayed for a monitor trap to 4. See the Processor Restrictions section for information.

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

- When you run under a mapped monitor and you use any of the following handlers, you must use the LOAD command to explicitly load them: BA, NQ, NU, SD, and ST. You cannot fetch those handlers.

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

- FRUNing or SRUNing a privileged foreground job containing extended memory overlays causes the system to crash. Do not FRUN/SRUN privileged foreground jobs containing extended memory overlays.

- Privileged foreground jobs cannot use virtual overlays. Any attempt to do so can crash the system. That rule does not apply to virtual foreground jobs; they run correctly with any type of overlay.

- 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.

Peripheral Device Restrictions

The following restrictions apply to the indicated RT–11 V5.6 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, if you upgrade from the RQDX1 or RQDX2 controller to the RQDX3 controller, you must reformat your RD5x disk. Perform the following procedure:
  1. Back up the disk written by the RQDX1 or RQDX2 controller, using that controller.
  2. Format the disk with the appropriate Digital formatting utility for RD series disks.
  3. Restore your data to the disk.

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 device is designed to perform optimally when streaming. The performance degrades significantly when not streaming. You will notice that performance degradation when operating the TK50 in a start/stop mode.

- The TK50 magtape drive may not immediately recognize a change in the write protection status of a magtape cartridge when that cartridge is loaded in the drive. A change in magtape cartridge write protection is not recognized until a significant event occurs. A significant event would be, for example, a change in the magtape track being read or written.

  You can guarantee recognition of a change in write protection status by unloading the magtape cartridge, changing the write protection on that cartridge, and then loading it. You do not need to remove the magtape cartridge from the drive. Push the red load/unload switch (to unload), wait for the green light to come on, change the write protection on the magtape cartridge, and then push the red load/unload switch (to load).

- You may experience a delay in receiving the monitor prompt (.) when you abort operations on the TK50 magtape drive by issuing a double CTRL/C. The TK50 magtape drive might not relinquish control to the monitor until the drive finishes performing an operation that it independently initiated. Whether you receive the monitor prompt immediately depends on the operation being performed by the TK50 when you abort.

  For example, when you instruct the TK50 to back up 10 or more records, it accomplishes that operation by rewinding to BOT (beginning of tape) and spacing forward to the desired record. If you abort that operation while the TK50 is rewinding to BOT, you will not receive the monitor prompt until the TK50 has reached BOT. However, if you specifically instruct the TK50 to rewind to BOT and then abort that operation, you receive the monitor prompt immediately.

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

Processor Restrictions

The following restrictions apply to the indicated RT–11 V5.6 processors:

- The presence of symbol KXCPU$ in the second configuration word, $CNFG2, does not imply support for KXT–11C or KXJ–11C processors.
- Because of instruction processing inherent in the DCJ11 microprocessor CPU, all processors that contain that 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. 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. In the case of 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.
- A system generated to support DU multiport booting cannot be run on an SBC–11 /21 (Falcon processor).
- When running RT–11 on an SBC–11/21, the SBC–11/21 must be jumpered for map selection 0, as specified in the *SBC–11/21 Falcon User's Guide*.
- LS, to be used on the available SLU serial port of an SBC–11/21 or SBC–11/21 PLUS processor, must be rebuilt with LS$PRI = 5.
- XL, to be used on the available SLU serial port of an SBC–11/21 or SBC–11/21 PLUS processor, must be rebuilt with XL$PRI = 5 or XL$SBC = 1.
- DD, to be used on the available SLU serial port of an SBC–11/21 or SBC–11/21 PLUS processor, must be rebuilt with DD$PRI = 5.

System Subroutine and Function Restrictions

The following restrictions apply to RT–11 V5.6 system subroutines and functions:

IASIGN

With FORTRAN–77, do not use the OPEN statement *NAME*= keyword if you have called the IASIGN function. The FORTRAN–77 OTS aborts and returns the error, "Duplicate file specifications". Instead, call IASIGN and specify the device and file specification by using the *idev* and *ifiltyp* parameters. You can then optionally open the file, using the OPEN statement without the *NAME*= keyword.

IGETR

GETR/IGETR is retained for backwards compatibility only. It contains bugs that affect much of its functionality. Use MGETR in its place.

Programmed Request Restrictions

The following restrictions apply to RT–11 V5.6 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.
- **.CDFN**

  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.

Utility Restrictions

Programs that have been designed to not check for wildcard filenames may create filenames with wildcard names in them. Some RT–11 utilities in fact do this.

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

BINCOM

- After encountering an error, such as an I/O error on an input or output file, BINCOM reports the error but might not flush the current buffer of differences. If you receive an I/O error message from BINCOM, do not rely on having received a report of all the differences in that buffer.
- When using wildcarding, the wildcard expression must be the same for both input file specifications. For example, the following expression is valid:

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

  While the following is not valid:

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

BUP

- DL (RL01/02) and DM (RK06/07) volumes require a two-step initialization process before being used as backup media. First, such volumes must be initialized by DUP, using the INITIALIZE/REPLACE command. That command creates the software bad-block replacement table structure on the volume. Second, such volumes must be initialized by BUP, using the BACKUP /INITALIZE command. Bad-block replacement on those volumes is then 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.

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. SD may not be run under another name. Another name will not be accepted.
- If you run the soft-I/O version of SD with an active foreground job (for example, VTCOM), the system ignores the keyboard entries. This is 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.

DUP

- The command COPY/DEVICE cannot be used by RT–11 V5 to copy images from magtapes to disks when the images were copied to the magtape using RT–11 V4. Use RT–11 V4 for that operation.
- FORMAT/VER[:ONLY] with DUP

  Prior to V5.6, DUP would use the contents of the replacement table from the home block of an RK06/07 or RL01/02 volume as found, possibly producing incorrect reports for DIR/BAD if the table was not valid. In V5.6, DUP validates the table before use.

  Prior to V5.6, DUP did not read the entire replacement table from the home block of an RK06/07 or RL01/02 volume. In V5.6, the entire replacement table is read.

  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:.

ERROR LOGGER

- You cannot run more than one Error Logger, and the job name for that single Error Logger must be ERRLOG.

IND

- 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.
- You cannot place CSI commands or DCL commands that require more than one line in IND control files. To execute multiline CSI commands and multiline DCL commands from an IND control file, create an indirect command file that contains the command and call the command file from the control file by using the $@ syntax.

  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.

  For example, the following IND control file executes a CSI command by creating an indirect command file, then calling the indirect command file:

  ```
  .IFF PIP .GOTO 1
  .OPEN SECOND.COM
  .DATA R PIP
  .DATA A.MAC=B.MAC
  .DATA ^C
  .CLOSE
  $@SECOND
  .1:  .ASKS ...
  ```

  Instead of creating the indirect command file, you can achieve the same result by using the following CCL command example:

  ```
  .IFT PIP PIP A.MAC=B.MAC
  ```

- You cannot run the BATCH processor and IND simultaneously. You must have SET KMON NOIND in effect to run BATCH. Also, you cannot run BATCH from an IND control file.

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.

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 by Digital 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
  ```

LIBR

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

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.

MACRO

- In MACRO–11 programs, all variable names of the form ...Vx are reserved by Digital. 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 by Digital. 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).

MDUP

- You cannot use the distributed MDUP programs with LSI–11 processors. 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.

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).

RESORC

- SHOW CONFIGURATION does not reliably display the existance 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.

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.

SIPP

- For RT–11 V5.6, 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 a program you specify for a SIPP session is overlaid, 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 you use SIPP 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.

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.

  That restriction does not apply to programs that use special mode input.

- When the single-line editor (SL) is enabled, the command COPY TT: filespec does not work.
- Problems 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 that problem; however, the Digital-recommended work-around is to SET SL KMON (rather than SET SL ON) or have your program set the EDIT$ bit (bit 4) in the JSW.
- SL and BATCH cannot be used together.

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.

TRANSFER

- TRANSF.EXE and the VMS Version 5.0 LAT were incompatible. Transferring files with this connection could cause premature LAT disconnection. A subseqent 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).

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.

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 USR.
- 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.

VTCOM

- VTCOM requires monitor timer support. The distributed SB monitor does not contain timer support, and therefore the distributed SB monitor does not support VTCOM. You must perform a system generation procedure for the SB monitor and request timer support before you can use VTCOM under SB.

Multiterminal Restrictions

The following restrictions apply to systems with the RT–11 V5.6 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 multiterminal 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.

Automatic Installation Corrections

The following problem in automatic installation has been corrected in RT–11 V5:

- If you requested no SYSGEN when running CONFIG.COM, SBFB.ANS remained on the configured disk. CONFIG.COM now deletes SBFB.ANS from the configured disk if you request no SYSGEN.

Monitor Corrections

The following monitor problems have been corrected in RT–11 V5:

- Previously, if the first action of the day was to create a file, and the system had been running since before midnight, the newly created file was given the previous day's date. This has been corrected in RT–11 V5.6.
- For RT–11 V5.3, any RT–11 monitor that was generated with memory parity support would not boot on machines immediately following a power-up. For RT–11 V5.4, monitors that are generated with memory parity support boot correctly on power-up.
- Previously, after initial start-up (cold boot), a monitor built for memory parity support running a program that read locations not initialized by the bootstrap could cause a memory parity error trap. Now, the bootstrap performs a read/write operation on all locations, including those below the 28K-word boundary, during the memory sizing operation, which corrects parity for all memory locations.
- The system no longer halts when a monitor without memory parity support is booted on a KDJ11-A and a cache parity error occurs.

BSTRAP

The following problems with the monitor bootstrap module have been corrected:

- Previously, BSTRAP could size ROM memory as RAM memory when that ROM memory was contiguous with RAM. For RT–11 V5.4E, BSTRAP correctly discriminates between contiguous ROM and RAM memory.
- Previously, BSTRAP incorrectly cleared the Q-bus bit (bit 6) in RMON fixed offset CONFG2 when a monitor that was generated with memory parity support was bootstrapped on a machine without memory parity hardware. For RT–11 V5.5, BSTRAP correctly does not clear that bit under those circumstances.
- Previously BSTRAP incorrectly determined a PDP–11/94 as a Q-bus and improperly initialized CONFG2 (offset 370 in the resident monitor fixed offsets) indicating such. In XM, this had the further effect of making the UB handler uninstallable. Without UB, the memory beyond 248 KB could not be used for DMA.

  For RT–11 V5.6, BSTRAP correctly initializes CONFIG2 for PDP–11/94 processsors.

KMON

The following problems with the keyboard monitor have been corrected:

- Previously, KMON occasionally did not load relocatable format (.REL) files correctly when those files were run with the FRUN or SRUN command. KMON occasionally stopped processing relocation lists prematurely because it incorrectly interpreted a -1 or -2 in the second word of a two-word item in a relocation list as the list terminator. For RT–11 V5.4G, KMON correctly distinguishes between a single-word -1 or -2 list terminator and a -1 or -2 in the second word of a 2-word item of a list, and no longer prematurely terminates list processing.
- Previously, issuing the CLOSE command where there existed an open file residing on a device controlled by a nonresident or nonloaded handler could return the error message, ?MON-F-No Device nnnnnn, and require a reboot. For RT–11 V5.4E, the CLOSE command works as documented.
- Previously, in indirect command files, KMON always executed a hard exit when the command SET TT QUIET was followed by a CTRL/C. Such a combination incorrectly caused a hard exit when embedded within an interrelated command operation. For example, if the combination was embedded between a GET and a START command, KMON cleared the starting address of the job and treated the START command as invalid. For RT–11 V5.5, KMON correctly treats such a combination, when embedded within an interrelated command operation, as a soft exit, and that problem does not occur.
- For SET commands, KMON sets the high bit in R1 on entry to handler SET routines when the device name is specified on the command line without a unit number. Previously, the high bit was set for device names only between AA and US. For RT–11 V5.5, if appropriate, KMON sets the high bit for all device names.
- The keyboard monitor R and RUN commands no longer automatically load locations 500-777 of the save image. The Chain Bit (bit 8 of offset 44 in block 0) must now be set to do that. You must manually set that bit before running programs such as MUBASIC that do not set that bit but require that the save image be loaded.
- Previously, issuing a command to UNLOAD a nonexistent job or device handler could crash the system when the monitor was built using certain SYSGEN conditions. For RT–11 V5.4D, unloading a nonexistent job or device handler does not crash the system under any monitor.
- Previously, KMON incorrectly processed only the first condition in a multicondition SET command for the TT handler. KMON ignored any conditions following the first and returned no error message. For RT–11 V5.4C, KMON correctly processes all conditions in a multicondition SET command for the TT handler.
- Previously, issuing a CLOSE command could hang or crash your system under the following circumstances: a channel was still open to the device containing the file you were closing, and that device was not loaded. Those circumstances could corrupt the fetched handler for that device. For RT–11 V5.4B, the CLOSE

command correctly fetches and releases the device handler, and the problem does not occur.

- Previously, the ABORT command did not correctly stop a job under the XM monitor when that job had called and enabled the .SCCA request. ABORT set bit 15 in a random word and not the .SCCA terminal status word. For RT–11 V5.4B, ABORT correctly sets bit 15 in the .SCCA terminal status word and correctly stops a job under those circumstances. This, however, works only for programs that check bit 15 and terminate if the bit is set. VTCOM is an example of a program that does not.
- When KMON processed an indirect command file containing multiple indirect command files, KMON saved the command file depth status before determining if all nested indirect command files existed. If one of the nested indirect command files did not exist, and error was set NONE, KMON reported the nonexistent file and then processed the next file twice. Now, KMON correctly saves the command file depth status while reporting the nonexistent file and then properly restores it. The next indirect command file is correctly processed only once.
- Previously, a virtual job could not pass information when chaining to another job. Also, you could not chain to a virtual job. Now, a virtual job can pass information when chaining to another job, and you can chain to a virtual job.
- When the EXECUTE command was issued and the device was not resident, KMON might have returned the error message ?KMON-F-File not found FORTRA.SAV. KMON now returns the error message ?KMON-F-Handler must be loaded <DEV:>.
- The EXECUTE command when issued for a device and file name, but no file type, causes KMON to first search for .FOR, then .MAC, then .DBL file extensions. If no filespec is found with those extensions, KMON previously might have returned the error message ?KMON-F-File not found FORTRA.SAV. KMON under those circumstances now returns the error message ?KMON-F-File not found DEV:FILENAM.TYP.
- Previously, nested indirect control files and nested indirect command files that contained a CTRL/C could skip the command following the CTRL/C. Control could revert to the calling file before the nested file finished processing. That could happen to indirect control files nested within indirect command files, and indirect command files nested within indirect control files. Now, nested files correctly finish processing under those circumstances.
- Lowercase characters within a command file are no longer converted to uppercase if the program receives the command line by issuing a .GTLIN request with the lowercase bit set.
- Virtual jobs now correctly initialize synchronous system trap vectors.
- The user stack in KMON containing context switching information was being written over by a background job overlaying KMON, or a background job in the process of overlaying KMON was being written-over by the context information for a system or foreground job. In either case, the background job's context

information or the background job itself was corrupted. Now, the user stack points to the transitional stack in RMON before doing the read operation that overlays KMON. The background job's context information and the job itself are not corrupted.

- When KMON parsed a file name longer than six characters, KMON printed the error message ?KMON-F-Command file not at end of line. In RT–11 V5, KMON prints the more appropriate message ?KMON-F-Error in file spec.
- With previous versions of RT–11, the device had to be loaded prior to chaining to another file. In RT–11 V5, devices need not be loaded.
- The command DELETE * now expands to the command DELETE *.* rather than the command *.NULL.
- When you omitted an argument from an option that required one, a trap to 4 occurred. In RT–11 V5, when you omit a required argument, the trap to 4 does not occur and KMON prints the error message ?KMON-F-Invalid value specified with option.
- If you performed a chain exit when running under a monitor with system job support, vector areas 472-476 were destroyed. Those vector areas are now preserved.
- Typing digits in a command name caused KMON to print the error message ?KMON-F-Ambiguous command. That message is appropriate for commands that contain only alphabetic characters. In RT–11 V5, using digits rather than characters in the command generates the error message ?KMON-F-Invalid command.
- In RT–11 V5, KMON prints the error message ?KMON-F-Invalid set parameter, rather than ?KMON-F-Illegal command, if the handler specified in the command rejects parameters.

MTTINT

The following problems with the multiterminal interrupt routine module have been corrected:

- RT–11 rejected (did not answer) incoming calls to the shared console that was SET to a remote DL line, unless a program was running that had attached to that DL line. RT–11 now accepts (answers) incoming calls under those circumstances.
- Generating a multiterminal XM monitor with BATCH support but without system job support no longer causes a branch out-of-range condition. Previously, this error occurred during the phase of SYSGEN that builds the multiterminal interrupt routine module (MTTINT).

RMON

The following problems with the resident monitor have been corrected:

- For RT–11 V5.4, a theoretical race condition could have occured between two jobs that called the .FPROT or .SFDAT requests when the second (higher-priority) job called either request before the first job completed either request. The request could complete successfully for the second job, but when the first job regained control, the data areas it was defining could have been corrupted, leading to unpredictable results. The unpredictable results did not include directory corruption or a system crash. For RT–11 V5.5, that condition can no longer occur.

- Previously, the single-terminal XM monitor incorrectly displayed the prompt, JOB>, when CTRL/X was pressed. For RT–11 V5.4F, pressing CTRL/X correctly displays the prompt, Job?, in agreement with all other monitors and the documentation.
- Previously, the RMON EMT processor could hang or crash the system when attempting to service a request block that contained an out-of-range EMT function code. The EMT processor, EMTPRO, performed an incorrect range check on EMT function codes. For RT–11 V5.4F, EMTPRO performs a correct range-check for EMT requests, and returns the error message, ?MON-F-Inv EMT <address>, when an out-of-range EMT function code is specified in the request block.
- Previously, RMON under FB and XM suppressed any .GTLIN prompt issued from a program that was run from an indirect command file when SET TT QUIET was in effect and the .GTLIN *type* parameter was set. The problem did not occur if the program was run as the last command in the file. For RT–11 V5.5, RMON processes the .GTLIN request as documented under those circumstances.
- The .ABTIO request now aborts only I/O queue elements belonging to the specified channel of the issuing job. Also, .ABTIO no longer loses any of the queue elements that it aborts.
- Programmed requests .FPROT and .SFDAT now work when issued from virtual jobs. The resident monitor no longer fails to relocate the address of the argument block.

USR

The following problems with the user service routines module have been corrected:

- Previously, a handler that was set to be both special directory structured and file structured (bits SPECL$ and FILST$ set in the device status word) would not gain control when directory operations were requested. For RT–11 V5.4D, a handler with this bit combination set in its device status word correctly gains control when directory operations are requested.
- The .CSISPC programmed request incorrectly closed channels 0, 1, and 2 when it found an error in a command received from the terminal or an indirect file. For RT–11 V5.4, the .CSISPC programmed request correctly does not modify any channels.
- When running the IND control processor under all monitors, any .GTLIN prompt from a program to the terminal was suppressed. Now, a .GTLIN prompt to the terminal from a program running under IND is not suppressed.

- When an indirect command file contained blank lines for utility version numbers, the version numbers were not always displayed. Version numbers are now displayed.
- Files created on a device were not always allocated the smallest empty space for the file length requested. The smallest empty space for the file length requested is now allocated.

Utility Corrections

The following problems have been corrected for utility programs in RT–11 V5:

BATCH

- Previously, specifying a unit number for BA with the UNLOAD command (UNLOAD BAn) incorrectly let you unload BA while it was running, which could crash the system. Now, KMON checks if BA is running before executing the UNLOAD BAn command and correctly prohibits execution of that command if BA is running.
- For RT–11 V5.2 and V5.3, BATCH was not updated to support the LS and LP handlers as file-structured devices. BATCH refused installation and returned the error message, ?MON-F-No device, or ?BATCH-F-Invalid LOG device. For RT–11 V5.4, BATCH was updated to support the LS and LP handlers and works as documented.

BINCOM

- Prior to RT–11 V5.6, a read error would occur when comparing two tapes (DIFF /DEV). This has been corrected in RT–11 V5.6.
- Prior to RT–11 V5.6, when comparing a device to a file (such as logical-disk file) a "*file is longer*" message was always produced. This has been corrected in RT–11 V5.6.
- Previously, if an invalid or nonexistent device was specified on a DIFFERENCES /BINARY command line, BINCOM incorrectly returned the error message, ?BINCOM-E-Input file not found <dev:filnam.typ>. For RT–11 V5.4F, BINCOM correctly returns the error message, ?BINCOM-F-Invalid device <dev>, under those circumstances.
- Previously, BINCOM could produce inconsistent output when using wildcard file specifications. If the contents of files were identical, sometimes two blank lines were omitted from the output. For RT–11 V5.4E, blank lines are no longer omitted from output files under those circumstances.
- Previously, performing a binary comparison involving a magtape, where a file specification included wildcards (* or %), could crash or corrupt the system. For RT–11 V5.4C, performing a binary comparison under those circumstances works as documented.
- BINCOM could corrupt data in its internal buffers before comparing it when the /END:n option was used and the following conditions applied:
  - — The shorter of two unequal length files being compared was of size n.
  - — The BINCOM internal buffers were a particular multiple of the shorter file segment being compared. That problem is corrected in RT–11 V5.4.

- When a wildcard binary comparison between files on two devices was directed to an output file, using a command such as follows, BINCOM wrote the output file only when the last two files compared contained differences:

  ```
  DIFFERENCES/BINARY/OUTPUT:TEST.DIF dd:*.* dd:*.*
  ```

  No output file was written even if files (other than the last two) contained differences. Now, BINCOM correctly writes the output file when any files on the devices are different.

- When using BINCOM to create a SIPP command file, the flag to strip the leading zeros was cleared at an incorrect time, which caused significant zeros in addresses higher than 7776 to be lost in the created SIPP command file. BINCOM now correctly clears the flag, and the SIPP file is correctly created.

- When comparing two diskettes, a bad block on one diskette was treated as end-of-device by BINCOM, which resulted in the other diskette being reported as longer. Now, BINCOM distinguishes the end of volume from a bad block. The problem is corrected.

- When a comparison was being performed and one or both of the files or volumes being compared resided on a diskette, BINCOM generated a hard error message if it reached the end of the diskette. In RT–11 V5, BINCOM returns to the keyboard monitor when BINCOM reaches the end of the diskette.

BUP

- Previously, BUP could restore the wrong data blocks when the BACKUP /RESTORE/DEVICE/FILE command was used to restore a file that spanned two or more disk volumes. The problem was dependent on the input device handler and existed for DU devices, but not DY devices. For RT–11 V5.4G, the problem is corrected.

- Previously, BUP would fail and could crash your system if you attempted to restore a file from a device saveset on magtape when that file began after block 32,767 on the original disk image. For RT–11 V5.4C, BUP correctly restores a file from a device saveset under those circumstances.

- Previously, RT–11 did not print any warning message if you restored a device image to your system device. Doing that can cause an uncoupling of RMON and the system files and could corrupt your system device. For RT–11 V5.4B, BUP displays the following warning when you issue the command BACKUP /RESTORE/DEVICE and your system device is the output volume:

  ```
  Restoring over SYSTEM volume!
  SY:/Restore; Are you sure?
  ```

- Previously, BUP attempted to back up files containing zero blocks. For RT–11 V5.4B, BUP refuses to back up files containing zero blocks and returns the following error message:

  ```
  ?BUP-F-No blocks to backup
  ```

- BUP for RT–11 V5.4A corrects several problems with the /VERIFY option, when that option is used to verify multivolume magtape backups. BUP for RT–11 V5.2 fails in the verify pass at the end of the last volume of a multivolume set. BUP for RT–11 V5.3 and V5.4 fails in the verify pass at the end of the first volume if another volume is to follow. BUP for RT–11 V5.4A correctly verifies multivolume magtape volumes.

- Previously, BUP wrote one extra magtape block when the number of disk blocks being read was divisible by eight but not divisible by eight times the number of memory buffers. For RT–11 V5.4A, BUP no longer writes an extra block under those circumstances.

- Previously, BUP displayed the following error message when you attempted to back up to a magtape drive that was write protected, not ready, or off line: ?BUP-F-Directory input error on ddn:, where ddn: was the input disk drive. For RT–11 V5.4A, BUP returns the following error message under those circumstances: ?BUP-F-Output error Mxn:, where *Mxn* is the magtape drive.

- Previously, BUP incorrectly reported an input error on the output disk drive when an error was encountered reading back a magtape block during a BACKUP /RESTORE/VERIFY operation. For RT–11 V5.4A, BUP correctly returns the error message, ?BUP-F-Input error mxn:ddn.BUP, where *mxn* is the magtape drive and ddn.BUP is the name of the restored file image on the output drive.

- Previously, BUP did not append the device and backup set number to the error message it returned (?BUP-F-Output error) if an error occurred while writing a magtape. For RT–11 V5.4A, under those circumstances BUP correctly appends the device and backup set name and returns the error message, ?BUP-F-Output error Mxn:ddn.BUP.

- Previously, BUP could fail to correctly initialize, back up, and restore data when the output volume was an RL01/RL02 (DL) or RK06/RK07 (DM) that was initialized with a non-RT–11 format. Results of a backup and restore operation using such a volume were unpredictable and often incorrect. For RT–11 V5.4, BUP correctly performs backup and restore operations for those volumes when they were previously initialized as non-RT–11 volumes.

- Previously, BUP could fail to restore a single file from a magtape backup volume when file directory segments on the backed-up volume were not forward linked. BUP could not correctly search the directory of a volume on a magtape backup if that directory contained one or more segments that were not forward linked to the other segments. For RT–11 V5.4, BUP correctly follows the links of the directory segments and successfully restores single files under those circumstances.

- Previously, BUP could incorrectly determine the size of an RK05 disk. When backing up a file that was larger than a single RK05 disk, BUP would incorrectly inform you to use PIP to copy the file to a single RK05. For RT–11 V5.4, BUP correctly sizes RK05 disks and no longer informs you to use PIP under those circumstances.

- Previously, when you attempted to use VBGEXE to run BUP, BUP returned the error message, ?BUP-F-Channel in use. For RT–11 V5.4, you can successfully use VBGEXE to run BUP.
- Previously, when a diskette was incorrectly inserted in a drive or a magtape was off line during a backup operation, BUP issued a fatal-level error message and aborted the operation. Now, under those circumstances, BUP issues an error-level error message and prompts you to mount another volume in the drive.
- When performing file backup operations, V5.2 BUP incorrectly wrote an output file of 1280_{10} blocks regardless of the input file size. That was true only for V5.2 file backup operations. Device backup operations under V5.2 performed as documented. For RT–11 V5.3, BUP correctly backs up files and devices as documented. See the V5.3 file V5NOTE.TXT for complete information.
- A copy operation could corrupt the output file when the number of blocks of the file buffer in memory was not a multiple of the output tape transfer block size. Now, the file buffer in memory need not be a multiple of the tape transfer size for BUP copy operations.
- BUP could not be used to back up a disk to magtape if the disk contained bad blocks covered with FILE.BAD files. Now, that restriction is removed.
- When running under the XM monitor, BUP was restricted to a single output volume. Now, backup is no longer restricted to a single volume.
- When backing up an RL02 to a file-structured TSV05, BUP gave an incorrect BUP-F-Read error message, although the transfer was completed correctly. Now, BUP no longer gives that incorrect error message.
- When backing up to a second RL02 disk, BUP gave an incorrect error message ?BUP-W-Incompatible output volume size. Now, BUP no longer gives that incorrect error message.
- When running under the XM monitor, BUP gave an incorrect EOT message when backing up to multiple tape volumes. Now, BUP no longer gives that incorrect EOT message.

DBG–11

- Previously you had to remove all breakpoints set in permanent areas of memory before unloading DBG–11 since unloading DBG–11 before removing breakpoints in those areas could cause unpredictable results.

  In RT–11 V5.6, breakpoints are automatically removed by DBG–11 at UNLOAD time.

DIR

- Previously, DIR incorrectly reported an error when you specifed multiple input devices and the first physical device name equated to a second or subsequent logical device name. For RT–11 V5.5, DIR translates all specified input devices to physical device names before checking for device validity.

- Previously, if you included the /SINCE and /BEFORE options in a single command line and in that order, DIR always indicated no files within the specified range on the device, even when appropriate files existed. For RT–11 V5.4C, including the /SINCE and /BEFORE options in any order in a single command line works as documented.
- When you omitted a portion of the date in the DIRECTORY/SINCE[:date] command, DIR did not correctly append values from the current system date. DIR now correctly appends current system date values for any you omit in that command.
- DIR no longer prints the incorrect ?MON-F- output error message prefix but instead prints the correct ?DIR-F- prefix, when the output device is write locked.

DUMP

- Previously, DUMP displayed a slash for the "*" character when decoding RAD50. This has been corrected in RT–11 V5.6.
- Previously, attempting to dump the contents of magtape device MU0 displayed the error message, ?DUMP-E-Unexpected EOF. For RT–11 V5.5, DUMP works as documented with TMSCP magtape units.
- Previously, DUMP did not check diskette drives for hard errors and incorrectly returned the error message, ?DUMP-F-Invalid command, when for example, the diskette drive contained no media or the door was open. For RT–11 V5.4E, DUMP correctly returns the error message, ?DUMP-F-Directory input error <dd:>, when it encounters such a hard error.
- Previously, if you assigned the DUMP default output device (LP) to a file-structured device (for example, ASSIGN DU1 LP), DUMP did not recognize the assigned device as the default output device and returned the error message, ?DUMP-F-No LP, when you attempted to run DUMP. For RT–11 V5.4, DUMP recognizes the assigned device as the default output device and does not return the error message under those circumstances.
- Previously, if you used the CSI form to run DUMP and specified an output device (other than TT:) without a file name, the command failed and returned the error message, ?CSI-F-Invalid command. For RT–11 V5.4, the command works as documented and places the output on the device you specify with the default file name DUMP.DMP.
- The DUMP/ONLY:n DM0: command previously caused the error message, ?DUMP-F-Invalid option /E, to be displayed, when the desired block number was less than 32768_{10}. The DUMP/ONLY:n DM0: command now works properly regardless of the block number.
- When a diskette was being dumped, DUMP generated a hard error message when DUMP reached the end of the diskette. In RT–11 V5, DUMP returns to the keyboard monitor when DUMP reaches the end of the diskette.

DUP

- Previously, when the command DIRECTORY/BADBLOCKS/START:0/END:0 DLx: is issued, the entire area of the disk was scanned.

  This has been corrected in RT–11 V5.6.

- Previously, DUP incorrectly allowed the attempted soft boot of nonsystem volumes, which halted the system. For RT–11 V5.5, DUP correctly returns an error message when the BOOT command is issued to a nonsystem volume.

- Previously, the command INITIALIZE/BADBLOCK/REPLACE:RETAIN did not execute correctly. Coding errors caused a conflict between the /BAD- BLOCKS and /REPLACE:RETAIN options, and FILE.BAD was not created for any bad blocks. For RT–11 V5.5, that command executes as documented.

- Previously, the bad block replacement table on an output RL01/02 or RK06/07 volume was not preserved when a COPY/DEVICE/FILE command was issued from a magtape input volume to the disk. The bad block replacement table for the disk that was originally copied to magtape was incorrectly copied to the output disk. For RT–11 V5.5 the bad block replacement table for those disk volumes is correctly preserved across device restoration operations.

- Previously, DUP incorrectly initialized magtapes by writing random characters to the ANSI VOL1 label record. For RT–11 V5.4B, DUP correctly initializes magtapes by writing blanks (ASCII space characters) in character positions 52 through 79 of the ANSI VOL1 label record, which conforms to ANSI standard X3.27-1978.

- For RT–11 V5.3, DUP would only soft boot processors that contained the KT11 memory management unit. For RT–11 V5.4, DUP does not require the KT11 memory management unit to perform a soft boot.

- DUP did not append a wildcard file specification (*.*) to a CSI command if you omitted it in a CSI command line that contained the /F and /I options (dd:file.ext /F=dd:/I). Therefore, DUP returned the error message ?DUP-F-Invalid command. Now, DUP appends the wildcard file specification if you omit it in a CSI command line that contains the /F/I options.

- If you wanted to create a file, using the CREATE/ALLO:nnn command, and there was insufficient space to create the file, the DUP utility displayed the error message ?DUP-F-Conflicting SYSGEN options. DUP now displays the error message ?DUP-F-No room for file.

- DUP no longer traps to 4 (odd address/bus timeout trap vector) when you initialize a volume with bad data in the extra bytes field of the directory header.

- Using the /DEVICE/FILE/WAIT options combination (DUP/I/F/W) in a COPY command is now supported, and DUP no longer returns an error message ?DUP-F-Invalid command.

- The BOOT/FOREIGN command (DUP/Q) leaves interrupts enabled on the device to be booted. It does not completely establish the hardware environment created by a CPU HALT/RESET followed by the specification of a device to a boot ROM. Some operating systems may not tolerate a hardware environment with interrupts enabled during their hardware boot code. Therefore, the BOOT /FOREIGN command is restricted to those operating systems which tolerate interrupts being enabled.

- The CREATE/ALLOCATE:-1 (filespec[-1]=/C) command incorrectly entered a file size of 65535 blocks in a device directory regardless of the actual amount of available space. CREATE/ALLOCATE:-1 now correctly enters the maximum space available in the device directory.
- When several DUP options were specified with the /WAIT (DUP /W) option, DUP prompted to mount the input volume once for each operation requested. In RT–11 V5, the prompt to mount a volume is printed only once.
- When using the command INIT/BAD/WAIT ddn: (DUP/B/W), where ddn is the device in which the system disk resides, DUP did not prompt to mount the input volume a second time for the initialization operation. Therefore, DUP initialized the system volume instead, destroying the boot blocks. DUP now correctly prompts you to mount the input volume.
- When using the BOOT/WAIT command, DUP printed the prompt ?DUP-W-Foreground loaded. Are you sure? twice. In RT–11 V5.1, the prompt is printed only once.
- RT11SB is bootable on SBC–11/21 Falcon processors.

ERROR LOGGER

- Previously, attempting to run ELINIT twice returned the fatal level error message, ?ERRLOG-F-Invalid message received. For RT–11 V5.4C, running ELINIT more than once returns the warning, ?ERRLOG-W-Invalid message received.
- The error logger did not report the correct time of error. The time of error is now reported correctly.

FORMAT

- The FORMAT utility now works more reliably in reporting bad blocks on RC25 disk volumes.
- Previously, formatting with verification (FORMAT/VERIFY) an RX33 diskette with the DU handler not loaded produced a fatal monitor error after the formatting completed. For RT–11 V5.4D, such a format operation works as documented and produces no fatal monitor error.
- Also, virgin RX33 formatting is supported.

HELP

- The HELP/PRINT topic command no longer causes the system to hang.
- Previously, the command HELP ? returned an incorrect error message. Now, that command returns the error message ?HELP-F-HELP not available for ?.

IND

- Previously, the .STRUCTURE directive, at exit, incorrectly forced IND to the default radix, OCTAL. For RT–11 V5.5, the .STRUCTURE directive no longer affects the current radix.
- Previously, the .CLOSE directive did not reset (disable) data mode when it was issued from within an indirect control file. For RT–11 V5.5, the .CLOSE directive correctly disables data mode, as documented.
- Previously, TYPEAHEAD mode only controlled the flushing of the keyboard input buffer when a .ASKx request was issued. For RT–11 V5.5, the keyboard input buffer is flushed whenever the status of TYPEAHEAD mode is changed.
- Previously, the .TESTDEVICE directive incorrectly returned off line (OFL) status for a DM device that was logically possible but physically nonexistent. By default, the DM handler contains valid table entries for eight devices. If a configuration contained less than eight DM devices, a .TESTDEVICE directive to a nonexistent DM device incorrectly returned OFL status rather than the correct NSD (No Such Device). For RT–11 V5.4B, the .TESTDEVICE directive correctly returns the NSD status for logically possible but physically nonexistent DM devices.
- For RT–11 V5.3, the .TESTDEVICE directive sometimes returned incorrect mounted (MTD) and unmounted (NMT) status information for logical disks. IND did not correctly track the new data table information in the LD(X).SYS file. For RT–11 V5.4, IND correctly tracks the LD(X).SYS file data tables, and the logical disk mount information returned by the .TESTDEVICE directive is correct.
- Previously, the .TESTDEVICE directive incorrectly returned off-line (OFL) status for a DL device that was logically possible but physically nonexistent. By default, the DL handler contains valid table entries for two DL devices. If a configuration using the distributed DL handler included only one DL device (DL0), a .TESTDEVICE to the nonexistent DL1 device incorrectly returned OFL status, rather than the correct NSD (no such device). For RT–11 V5.4, the .TESTDEVICE directive correctly returns the NSD status for logically possible but nonexistent DL devices.
- Previously, the .TESTDEVICE directive incorrectly returned the LOD (loaded) status for magtape handlers that were in fact not loaded. Other status information returned on unloaded magtape handlers was also randomly incorrect. For RT–11 V5.4, the .TESTDEVICE returns correct status information on unloaded magtape handlers.
- Previously, the .TESTDEVICE directive could incorrectly return size and status information for some DU devices. That problem is corrected for RT–11 V5.4.
- Previously, IND did not interpret some errors returned by PIP and DUP. Therefore, IND did not correctly process all PIP and DUP errors. For RT–11 V5.4, that error information is now mapped to IND’s internal error table, and IND processes PIP and DUP errors correctly.
- Previously, when the .TESTDEVICE or .VOL directive encountered some device read-related errors, IND would exit with a fatal error condition. That problem

happened even if you included the .ONERR directive in your code, because the .ONERR directive attempted to access the <FILERR> special symbol and .TESTDEVICE and .VOL did not support <FILERR>. Now, the .TESTDEVICE and .VOL directives support the <FILERR> special symbol. You can perform error recovery in your code that prevents IND from exiting due to read-related error conditions.

- The .TESTDEVICE directive incorrectly reported the status of the SPOOL device handler SP as OFL (offline). The .TESTDEVICE directive now correctly reports the status as UNK (unknown).
- The .TESTDEVICE directive incorrectly reported the status of all logical disks as OFL (offline), whether a logical disk was mounted or not. That problem was caused by a change made to the LD assignment table format for RT–11 V5.3 that was not recognized by .TESTDEVICE. For RT–11 V5.4, .TESTDEVICE correctly reports the status of logical disks.
- IND trapped to 4 when the .TEST directive was used on a valid string symbol. That problem occurred because of invalid pointers during the .TEST directive processing. Now, the .TEST directive no longer traps to 4 when it is used with a valid string symbol.
- Previously, the .TEST directive failed to detect nonalphanumeric characters appearing in a string symbol after the first occurrence of a lowercase alphabetic character. For example, performing a .TEST SYMBOL where SYMBOL contained the string "v/" would incorrectly set the special symbol <ALPHAN> to TRUE. For RT–11 V5.4B, the .TEST directive recognizes nonalphanumeric characters appearing in a string symbol after a lowercase alphabetic character, and <ALPHAN> would be correctly set to FALSE for this example.
- IND aborted with a fatal error when a read error occurred while the .STRUCTURE directive was determining a device file structure. Now, the .STRUCTURE directive is modified to return an 'UNKNOWN' (code 0) structure if a read error occurs. Digital continues to recommend that you issue the .TESTDEVICE directive before the .STRUCTURE directive to make sure a device is on line and mounted.
- Executing a control file, using the DCL keyboard command syntax RUN IND ctrl-filespec[/options] or IND ctrl-filespec[/options] did not let you pass parameters. Those commands now let you pass parameters.
- IND now correctly handles symbol names of more than six characters. However, IND continues to recognize only the first six characters.
- IND contained a restriction that blank and tab characters could not be placed between operators and numeric symbols. IND no longer contains that restriction, so numeric expressions can be formatted for better readability.
- The .TESTDEVICE directive no longer returns an incorrect device size in <EXSTRI> when reporting the status of a device having one size.

- Previously, the .TESTDEVICE directive returned the error messages, ?IND-F-Device error, and ?IND-F-Device read error, if you used the KMON command LOAD ddx:=F or LOAD ddx:=job to attach to a foreground or system job. Now the .TESTDEVICE directive includes ATT or NAT (attached or not attached) in field nine of special symbol <EXSTRI>, and the two error messages are not displayed.
- Previously, a problem occurred with using the .VOL directive if it was issued for a device assigned to the foreground or a system job, using the LOAD ddn:=F or LOAD ddn:=job commands. That operation returned the error message, ?IND-F-Data file error, Invalid device or unit. Now, the .VOL directive returns the error message ?IND-F-Invalid device or unit, device is attached. That error can be intercepted using the .ONERR directive, and you should use the .TESTDEVICE directive to determine if the device is indeed attached and to what job.

INDEX

- INDEX is not a supported keyboard command and can only be invoked using a CCL or CSI command. Previously, if you invoked INDEX and incorrectly included any parameter in the command line, INDEX terminated abnormally after the second screen of information. For RT–11 V5.4E, INDEX correctly ignores any command line parameters.

KED

- Previously, pressing the SELECT key and then pressing the delete key could cause the select function to work incorrectly. If you advanced the cursor, KED did not select the first character in your selection. If you backed up the cursor, KED incorrectly also selected the character to the right of the cursor. For RT–11 V5.5, pressing the delete key no longer causes the select range to malfunction.
- Previously, KED could incorrectly return the prompt, ?KED-W-File not found - Create it (Y/N)?, when there was no volume in the specified drive. For RT–11 V5.4C, KED returns the error message, ?KED-F-Unable to open input file.
- The FIND command now works for most control characters except ^B, ^C, ESCAPE, ^F, and ^X.
- The LOCAL command now works when applied to a range containing a form-feed (<FF>) character.
- KED displayed an I/O error message when it reached the end of an inspected disk. KED now displays the end-of-file symbol.
- The KED chain interface now works properly and is defined in the .XKEDF macro in the file SYSTEM.MLB.

LD

- For RT–11 V5.3, the LD handlers LD.SYS (appropriate for the unmapped monitors) would crash the system if a MOUNT command was issued after the LD handler was loaded. The LD handler for the XM monitor (LDX.SYS) did not exhibit this problem. For RT–11 V5.4, the LD handler no longer crashes the system if a logical disk is mounted after LD is loaded.

- Previously, mounting a logical disk which resides on a write-protected RT–11 volume yielded an unnecessary warning message whenever the logical disk file was already protected.

LET

- Previously, LET would convert all characters in a defined string to uppercase. This has been corrected in RT–11 V5.6.

LS/XL

- Prior to V5.6, if an LS or XL handler is FETCHed or LOADed under a multiterminal system, and the handler was built to use a CSR or VECTOR which was also specified for use with a DL line of the multiterminal system, the system might crash.

  With V5.6, both the LS and the XL handlers, when FETCHed or LOADed under a multiterminal system, check to see if the CSR or VECTOR they were built to use conflicts with a CSR or VECTOR of the multiterminal configuration, and refuses to be FETCHed or LOADed if so.

  If you use the LS or XL handlers and wish to fetch or load under a multiterminal system, build for multiterminal handler hooks support (which is not much more since you must have built a multiterminal monitor), set LS to use the console DL (SET LS LINE=0), move the console (SET TT CONSOL=1) and then you can use LS.

  After you have completed using LS, you can unload it, set the console back to the boot-time console (SET TT CONSOL=0) and the system will continue working, which is better than as currently documented (which says that if you try to set console back to line 0, the system will crash).

LIBR

- Previously, fatal level error messages generated by the LIBRARY command occasionally returned the LIBR utility CSI prompt (*), rather than correctly returning control to the keyboard monitor. For RT–11 V5.4D, that problem is corrected.
- When you incorrectly coupled the /UPDATE option with the output file name, the error message returned by LIBR incorrectly included the input file name in the error message. Now, LIBR correctly includes the output file name in the error message.
- Under certain circumstances, LIBR generated macro libraries where the last entry would not appear in the library directory. That problem normally occurred when adding to SYSMAC or HELP, and no LIBR error message was displayed. Now, all entries into macro libraries appear in the library directory.
- The LIBR utility failed if a forms library directory exceeded one block. LIBR now allows forms library directories to exceed one block.

LINK

- Previously, under the following circumstances, LINK incorrectly processed a transfer address that occurred within a library object module. The result was a load module with an incorrect starting address. The object module containing the transfer address was:

  1. physically located in the library following another object module that contributed to the same program section in which the transfer address was defined; and
  2. the transfer address was defined in a relocatable concatenated program section.

  For RT–11 V5.5, LINK correctly processes transfer addresses under those circumstances.
- Previously, certain fatal error conditions caused the LINK command to incorrectly display the CSI prompt (*). For RT–11 V5.4F, the LINK command correctly returns to the monitor prompt (.) after encountering any fatal error condition.
- Previously, LINK did not give the correct value of zero to undefined symbols specified by the .WEAK directive in overlaid files. For RT–11 V5.4A, LINK correctly gives a value of zero to undefined symbols under those circumstances.
- Previously, if you directed the binary output file to the NL device in a LINK command line, LINK failed and returned the error message, ?LINK-F-Storing text beyond high limit. LINK did not recognize that you were directing the output to the null (NL) device and failed, producing that message indicating a read/write error. For RT–11 V5.4, LINK does not fail when you direct binary file output to the NL device in a LINK command line.

- Previously, when link maps contained more than one page of undefined global symbols, the header information was incomplete on the second and following pages. For RT–11 V5.4, the link maps contain complete header information on all pages.
- RT–11 V5.3 removes the restriction stating that the total combined size of all virtual overlay segments created by the LINK /V option cannot exceed 96K words. Now, the total combined size cannot exceed 2016K words.
- Previously, when you linked programs, using the /DUPLICATE option, LINK occasionally returned the error message,

  ```
  ?LINK-F-Invalid RLD symbol in DEV:FILNAM.OBJ
  ```

  and prematurely aborted. For RT–11 V5.4, that problem is corrected.
- Previously, when you linked overlaid programs, using the /INCLUDE option, LINK occasionally returned the error message, ?LINK-F-Internal error, and prematurely aborted. For RT–11 V5.4, that problem is corrected.
- The /DUPLICATE option incorrectly repeated absolute symbols in the link map for each duplication of a library module containing the absolute symbol. Now, only the defined global symbols are entered in the link map.
- In some cases when you used the /U option, LINK displayed the error message ?LINK-F-Address space exceeded, even though the program's high limit was less than 32K words. That problem was caused by the improper initialization of a variable and has been corrected.
- In some cases when you linked valid object modules, LINK displayed the error message ?LINK-F-Invalid GSD in DEV:FILNAM.TYP. That problem was caused by code that processes transfer address entries in the global symbol directory (GSD) and has been corrected.
- When absolute binary (.LDA) files were linked with ODT or VDT, the load address of the .LDA file was placed such that it was written over and corrupted by the ODT or VDT module declaration macro, which produced an incorrect link map. Absolute binary files now link with ODT or VDT at the correct load address (above ODT or VDT) and produce an accurate link map. When preparing Software Performance Reports (SPRs), include the version information from ODT/VDT that is found in the link map as .AUDIT and .ODT or .VDT symbol values.
- LINK generated a trap to 4 error when you linked an object module that did not contain a program section.

  LINK assumed that there would always be at least one program section in a module. This is not a problem with MACRO and FORTRAN, because they always produce at least one program section per module. It was a potential problem with other compilers if they can produce a module that does not contain any program sections.

  This is corrected in RT–11 V5.6.

PAT

- Previously, PAT occasionally garbled the file name in error messages that returned a file name. For RT–11 V5.4G, PAT no longer garbles file names in error messages.

PIP

- Previously, PIP made permanent an incomplete file that resulted from a COPY /CONCATENATE command and an unfound input file fatal error. For RT–11 V5.5, PIP correctly discards such an incomplete file.
- Previously, if the output file that resulted from a COPY/ASCII operation was exactly 512_{10}+1-character long, the final character was discarded. For RT–11 V5.5, PIP does not discard the last character.
- For RT–11 V5.5, PIP correctly accepts the COPY/DELETE/MULTIVOLUME command.
- Previously, PIP could incorrectly return the error message, ?PIP-F-Device in use, when attempting to write a file to a magtape that was full. For RT–11 V5.4C, under those circumstances PIP correctly returns the error message, ?PIP-F-Device full.
- Previously, if PIP was writing the first file to a magtape and that magtape's controller lost position, PIP searched back on that magtape to find a tape mark. Some magtape controllers return only a BOT and not a tape mark. Therefore, PIP could get into an infinite loop searching for the nonexistent tape mark. For RT–11 V5.4C, PIP checks for both BOT and a tape mark on magtapes when the magtape controller loses position while PIP is writing the first file to that magtape.
- Previously, PIP and the FSM (magtape file structure module) could incorrectly handle a position-lost write error returned by a magtape handler. The interaction between PIP and the FSM could backspace the magtape to the wrong position and incorrectly preserve a partial section of the bad file, leaving the magtape in an indeterminate state. For RT–11 V5.4C, PIP and the FSM properly interact and perform the backspacing. The bad file is correctly not preserved, and the magtape is left in a valid state.
- Previously, using any wildcard in the input file name with the COPY/PROTECT command caused the monitor to trap to 4. For RT–11 V5.4B, the COPY /PROTECT command works as documented.
- Previously, the FB monitor failed to perform copy operations from one non-file-structured device to another non-file-structured device. The specific case of copy operations from TT to TT also failed under the XM monitor. For RT–11 V5.4B, copy operations from one non-file-structured device to another work as documented.
- Previously, characters copied from your terminal (TT:) to an output device (such as LP:) were always copied in uppercase. For RT–11 V5.4, characters are copied from your terminal with no change in case.

- Previously, if the command DELETE/QUERY filnam.SYS was included in an indirect command file, responding N (no) to the prompt returned by that command incorrectly returned the message ?PIP-W-Reboot. However, the specified file was correctly not deleted. For RT–11 V5.4, PIP does not return that message under those circumstances.
- When a hard error occurred on a magtape, such as an attempt to copy to a write-protected magtape, PIP incorrectly trapped to 4 or 10. Now, PIP correctly returns an error message indicating a hard error condition.
- The COPY/SETDATE command, when the input and output filespecs were the same, incorrectly moved the specified input file while changing the file date. That is, the command actually performed a COPY operation. For RT–11 V5.4, the COPY/SETDATE command only changes the file date when the input and output filespecs are the same and does not perform a COPY operation.
- PIP incorrectly treated printers (LP) as sequential access devices for COPY operations from the terminal (TT). Therefore, a COPY command from TT to LP produced an invalid device error. Now, PIP correctly copies from the terminal to the printer.
- When the system device (SY) and the data device (DK) were assigned to the same device, a COPY operation from SY to DK incorrectly performed the COPY operation twice. Now, PIP renames the file to the output file name rather than copying it.
- The COPY/ALLOCATE:size command created a file with the current date rather than the input file date. Now, that command correctly uses the input file date as the creation date for the output file.
- When the COPY/MULTIVOLUME/NOREPLACE (PIP /E and DUP /R) command was issued, PIP processed the first output volume correctly. However, when the second output volume was mounted, PIP copied the next file from the input volume to the second output volume regardless of whether the file already existed on that output volume. PIP was not reading the directory of the second output volume before beginning the COPY operation. PIP now reads the directory of each volume immediately after it is mounted, and the /MULTIVOLUME /NOREPLACE options work correctly.
- When COPY/PREDELETE (PIP /O) was performed and the specified input and output volumes were the same, PIP deleted the file. In RT–11 V5, COPY /PREDELETE performs a RENAME operation when the input and output volumes are the same.
- When wildcards were used in a RENAME/NOREPLACE (PIP /N) operation, the first file was found but not renamed, and an error message appeared. RENAME /NOREPLACE now works as documented in the *RT–11 System User's Guide* and *RT–11 System Utilities Manual, Part II*.
- When the /WAIT option (PIP /E) was specified, PIP attempted to read the output volume's directory before it prompted to mount the output volume. Therefore, an error occurred (if no volume was present in the output device) or the wrong

volume could be read. PIP now prompts you to mount the output volume before attempting to read its directory.

QUEUE

- QUEUE no longer appends a carriage return to the end of files that are queued to VM.
- QUEMAN displayed an error message when you issued the 2-slash (//) line terminator option on a line by itself. QUEMAN now accepts the 2-slash line terminator option on a line by itself.
- QUEUE did not recognize and correctly print asterisk (*) and percent (%) characters used in job names on banner pages. QUEUE now correctly handles asterisk and percent characters for banner pages.

RESORC

- For RT–11 V5.4, the SHOW CONFIGURATION command no longer displays 'Global .SCCA flag is disabled' when global .SCCA support is not requested during SYSGEN.
- Previously, when global SCCA was enabled, a portion of the SHOW CONFIG data was displayed twice. That problem has been corrected.
- The SHOW DEV:DU command now displays the DU PORT, UNIT, and PARTITION translation table as shown in the *RT–11 System Utilities Manual, Part II*.
- When more than 14 logical names were assigned to a device, the SHOW command (RESORC) printed a corrupted listing. More than 17 logical name assignments to a single device caused RESORC to hang the system. RESORC now correctly prints listings, and more than 17 logical device name assignments to a single device no longer hang the system.
- Previously, the SHOW CONFIGURATION command on a PDP–11/93 or PDP–11 /94 processor listed the processor type as unknown and incorrectly indicated that there was cache memory on the system.

  This has been corrected in RT–11 V5.6.

SETUP

- Previously, the SETUP GENERIC command returned an error message if you were using a PDP–11 processor with a VT200 series terminal. For RT–11 V5.4, the SETUP GENERIC command works as documented with VT200 series terminals.
- For RT–11 V5.3, the SETUP SMOOTH command did not work on PDP–11 processors as documented. For RT–11 V5.4, SETUP SMOOTH does work as documented on PDP–11 processors.
- SETUP NOCOMPOSE now works as documented in the *RT–11 System User's Guide*.

SIPP

- Previously, SIPP could incorrectly compute the actual number of overlays in a program with multiple overlays because the correct overlay information was not always read into memory. The incorrect computation could cause SIPP to display incorrect values for some virtual overlaid file contents. Also, SIPP could incorrectly return the error message, ?SIPP-E-Invalid segment number, when the requested segment was valid. For RT–11 V5.4F, SIPP correctly computes multiple overlays, displays correct values, and no longer reports valid overlay segments as invalid.
- Previously, if you used the /L option and the device was write locked, SIPP incorrectly returned the error message ?MON-F-Dir IO error <address>. For RT–11 V5.4, SIPP works as documented and does not return that error message under those circumstances.
- Previously, if you used CCL command mode to run SIPP and included the checksum option (/C), the checksum prompt was not returned and SIPP saved none of the modifications. For RT–11 V5.4, when you use CCL to run SIPP and include the checksum option, the command works as documented.

SL

- Specifying a unit number for SL with the UNLOAD command (UNLOAD SLn) incorrectly let you unload SL while it was running, which could crash the system. Now, SL checks if it is running before executing the UNLOAD SLn command and correctly prohibits execution of that command if SL is running.
- SL is now supported with virtual jobs.
- Previously, SL did not disable address checking under certain circumstances, which could cause the MACRO–11 assembler, when run under the CSI command format, to abort with the error ?MON-F-Inv addr. For RT–11 V5.4C, SL correctly disables address checking and no longer can cause the MACRO–11 assembler to abort.

SPOOL

- If a SETUP PRINTER command was issued while SPOOL was running, the system would hang or trap to 4. Now, issuing the SETUP PRINTER command while SPOOL is running works as documented.

SRCCOM

- A "*Too many differences*" error message from SRCCOM may mean that there was not enough memory available to perform the compare. This usually occurs during wildcard comparisons on devices that contain a lot of files. One user action in this instance is to see Section 3 of the *RT–11 System Message Manual* for information on how to increase memory space. Another, is to compare one subset of files at a time.

- When wildcard comparing with SRCCOM, if one of the pairs of files being compared has too many differences, a fatal message is produced and SRCCOM terminates.
- Previously, performing an ASCII text comparison involving a magtape, where a file specification included wildcards (* or %), could crash or corrupt the system. For RT–11 V5.4C, performing an ASCII text comparison under those circumstances works as documented.
- Previously, the DIFFERENCES command with the /OUTPUT:filespec option could produce a corrupted output file when wildcards were used for input file names. For RT–11 V5.4A, a corrupted output file is not produced under those circumstances.
- When the DIFFERENCES /CHANGEBAR (SRCCOM /D) option was specified with the console as the output device, the message ?SRCCOM-I-No differences found, would overwrite the end of the console output. That message is now printed on a new line.
- Previously, SRCCOM ignored the last line of a file that does not end with <LF>, <VT>, or <FF>.

  SRCCOM used a library routine to get a line of text from the files for comparison. This routine returned the line to SRCCOM only if it ends with <LF>, <VT>, or <FF>. In the case of the last line in a file, not terminated with one of the preceding characters, an EOF error wsa returned and SRCCOM did not include the line in the comparison. The library routine is modified in RT–11 V5.6 to recognize the EOF as a valid line terminator, and return the line to SRCCOM.

TRANSF

- TRANSF now forces the terminal to NOCRLF when you run RT–11 under a nonmultiterminal monitor.
- If you type a CTRL/O during a file transfer, the CSI prompt character is now displayed when the transfer is completed.

UCL

- Previously, UCL did not consider a command containing an asterisk (minimum abbreviation character) as a valid replacement for the same command that did not contain the asterisk. UCL did not clear the replaced command. For RT–11 V5.4, UCL clears the replaced command under those circumstances.
- The UCL data file (UCL.DAT) was occasionally corrupted before the maximum number of commands was entered. The corruption occurred because the data file held only 256_{10} characters for the command definition field, and that number was exceeded before the maximum number of commands was entered. Now, a customization patch lets you change the size of the UCL data file to any maximum size you specify.
- Previously, when UCL was run via RUN UCL, it displayed an error message instead of displaying its version number.

- Prior to RT–11 V5.6, UCL did not handle string lengths correctly in some cases.
  - — SHOW COMMANDS displayed command definitions incorrectly if they were 127 or 128 characters long.
  - — Command definitions which replace previously defined commands could overflow the definition table entry.
  - — There was no limit to the length of the command passed from UCL to KMON, so the chain buffer may overflow, corrupting the stack.

VTCOM

- Previously, VTCOM would hang when it received an XOFF from the host, and you then sent an XON (CTRL/Q) to the host from the VTCOM command prompt. You could not abort VTCOM and would need to reboot your system. Now, VTCOM does not hang under those circumstances.
- The EXIT and PAUSE commands now close any open logging file before exiting to the monitor.

Device Handler Corrections

The following device handler problems have been corrected in RT–11 V5:

DL

- Previously, the DL handler did not check .READx/.WRITx requests from a non-file-structured .LOOKUP to determine if they correctly accessed replaced blocks only through the bad-block replacement table. For RT–11 V5.4, the DL handler rejects any attempt to access replaced blocks other than through the bad-block replacement table and returns the carry bit set. Special function (.SPFUN) reads and writes continue to directly access any block on the device.
- Previously, when you issued the COPY/DEVICE/RETAIN command to the DL handler, DL incorrectly read two blocks beyond the data area on the input volume. DL then wrote those two blocks to the first two replaced blocks on the output volume, thereby corrupting those first two blocks. For RT–11 V5.4, the DL handler only reads from and writes to the data area when you use the COPY/DEVICE/RETAIN command.
- Previously, some invalid SPFUN codes directed to DL could cause a default action to occur. For RT–11 V5.4, all invalid SPFUN codes directed to DL are correctly ignored.
- The DL handler no longer occasionally causes a system crash when used under a monitor with error logging support on a system with an RLV12 controller.
- When a disk reported certain hard errors to the DL handler, the handler would reverse seek one track and then reseek to the desired track. The DL handler now homes to track 0 and then reseeks to the desired track, which improves the chance for recovery from the hard error.

DM

- Previously, the command INITIALIZE/BADBLOCKS did not work correctly on RK06/RK07 (DM) disks and returned the error message, ?DUP-F-Invalid directory DMn:. For RT–11 V5.5, that command works as documented with DM devices.
- Previously, when performing an absolute read or write (377 or 376) special function, and the starting block number was xxxx76, where xxxx was any number, data was read from or written to incorrect memory locations. For RT–11 V5.4F, absolute read or write operations with DM perform as documented.
- Previously, the DM handler did not check .READx/.WRITx requests from a non-file-structured .LOOKUP to determine if they correctly accessed replaced blocks only through the bad-block replacement table. For RT–11 V5.4, the DM handler rejects any attempt to access replaced blocks other than through the bad-block replacement table and returns the carry bit set. Special function (.SPFUN) reads and writes continue to directly access any block on the device.
- Previously, when you issued the COPY/DEVICE/RETAIN command to the DM handler, DM incorrectly read two blocks beyond the data area on the input

volume. DM then wrote those two blocks to the first two replaced blocks on the output volume, thereby corrupting those first two blocks. For RT–11 V5.4, DM only reads from and writes to the data area when you use the COPY/DEVICE /RETAIN command.

- Previously, some invalid SPFUN codes directed to DM could cause an indeterminate action to occur. For RT–11 V5.4, all invalid SPFUN codes directed to DM are correctly ignored.

DU

- Previously, the DU handler required customization to enable multiport booting and SBC11/23 booting. For RT–11 V5.6, there is no requirement for customization. In addition, support for booting partitions is now available in RT–11 V5.6.
- Previously, when performing an absolute read or write (377 or 376) special function, and the starting block number was xxxx76, where xxxx was any number, data was read from or written to incorrect memory locations. For RT–11 V5.4F, absolute read or write operations with DU perform as documented.
- Previously, an I/O request with a zero word count (a seek operation) directed to some MSCP controllers could cause the system to crash or corrupt data. For RT–11 V5.4D, seek operations to all MSCP controllers function properly.
- If the system crashes while the DU handler (DUX.SYS) is performing bad-block replacement on a disk, restarting the system and accessing that disk should cause the DU handler to recover bad-block replacement information and complete the bad-block replacement. Previously, that process did not work correctly. For RT–11 V5.4A, that process works correctly.
- Previously, if a bad block occurred during a special MSCP BYPASS recovery operation (.SPFUN 360 with blk argument set to 1) and the DU handler was responsible for bad-block replacement, bad-block replacement did not occur. DU attempted to replace a different block that might or might not be bad. That attempt was not always successful and could cause the system to crash. For RT–11 V5.4A, DU correctly replaces bad blocks during a special MSCP BYPASS function.
- For RT–11 V5.4, the command SET DU SUCCES works as documented.
- For RT–11 V5.4, you can issue the command SET DUx PORT=0 when you have a single port DU configuration. That command no longer returns an error message in a single port configuration.
- Previously, SPFUN 372 could only read the handler translation table. Now, SPFUN 372 writes and reads the table.
- The DU handler installation procedure did not check for port validity when installing a DU device unit that was SYSGENed for multiple ports. Attempting to access a DU device unit that was mapped to an invalid port caused the system to crash. Now, that procedure returns an error message and does not crash the system. Use the SHOW DEV:DUn command to determine if the port mapped to

a DU device unit is valid. An invalid port is indicated by an asterisk (*) displayed before the port number.

- DU did not correctly check and truncate non-file-structured .WRITE requests that crossed partition boundaries, which could cause data that was written across a partition to corrupt data already written in the next partition. DU now correctly checks and truncates requests that cross a partition boundary. The amount of data actually transferred is not reported to the program requesting I/O, but a hard error is indicated.
- After you boot an SBC–11/21 PLUS processor, the SHOW DEVICE command displays the standard DU CSR (172150), not the DU CSR (176150) used by the SBC–11/21 PLUS. If your CSR address is 176150, Digital recommends you use the SET DU CSR=176150 command after booting your system for the first time. That command enables RESORC to display the correct CSR when you issue the SHOW DEVICE command.
- Volume sizing code reported incorrect information when sizing a nonexistent MSCP unit number. Incorrect information was also returned on some subsequent size requests. The DU handler now returns correct sizing information under those circumstances.
- Errors produced an excessive number of retries because the retry count was incorrectly initialized. The retry count now initializes correctly.
- SPFUN 360 and 371 (special function bypass) were restricted in use only to privileged jobs because the response and command buffer was required to be in the low 28K words of memory. The response and command buffer is no longer required to be in the low 28K words of memory; that restriction is removed.
- Previously using the SET DUn CSR command on a single-ported DU handler in XM did not work.

  This has been corrected in RT–11 V5.6.

DY

- Previously, you could not generate support, using SYSGEN, for a second DY controller if DY included error logging support. For RT–11 V5.5, that restriction no longer applies.
- Previously, DY accepted a RETRY value of 1 (SET DY RETRY=1), which could produce an error condition on the first I/O operation when the DY handler was used in single density mode. The initial read operation when density changes requires 2 retries. For RT–11 V5.5, the smallest RETRY value supported by DY is 2, thereby correcting the problem.
- Previously, the state of the drive density bit in the RX02 drive status register was not always preserved by the DY handler. Some operating systems require that bit to determine media density during bootstrap operations. Therefore, the command BOOT/FOREIGN could fail on RX02 drives. For RT–11 V5.4E, the DY handler correctly preserves the drive density bit in the RX02 drive status register.

- Previously, attempting to access an RX02 diskette subsystem on a Q-bus processor when the RX02 was not powered-up would hang the system. The system had to be rebooted. For RT–11 V5.5, the RX02 subsystem can be powered-up and accessed.
- For RT–11 V5.5, many I/O operations to RX02 diskette drives that previously caused a reset on the drive no longer cause a reset to occur.
- An overflow of the SET table area of the RX02 handler occurred if the handler was built to include second controller and error logger support. The overflow no longer occurs.

LD

- Previously, LD incorrectly processed multiblock read or write operations that extended beyond the last block of an LD unit. LD did not truncate the read or write operation and incorrectly used the specified value in the word count parameter as an address location, writing a zero to that address before returning a hard error. For RT–11 V5.4G, LD under those circumstances correctly only truncates the operation and returns a hard error.
- Previously, start-up command files aborted if they attempted to access a logical disk that was mounted on a noninstalled device. LD returned the fatal error severity message ?LD-F-Device not installed DEV:. Now, the severity of the error message returned by LD under those circumstances is reduced to a warning ?LD-W-Device not installed DEV:. You receive the warning message and the start-up command file no longer aborts.
- Previously, you could not customize the LD handler name suffix; you had to use a suffix of X or M. Now, you can customize the LD handler suffix character. You must perform customization D.3, Changing the Handler File-Name Suffix, located in the *RT–11 System Generation Guide*. You must also perform the customization described in the *RT–11 System Generation Guide* to make that suffix character known to the monitor bootstrap.
- Previously, a fetch error occurring during an operation requiring LD to fetch a handler could corrupt LD's internal table. Now, a fetch error occurring under those circumstances correctly returns the error message ?LD-F-Fetch error.
- Previously, the error message ?LD-F-No file specified <DEV:> printed extraneous characters. It no longer does that.
- Monitors built with nonstandard suffixes caused the SET LD CLEAN and SHOW SUBSET commands to not work properly. Those commands now work properly with monitors built with nonstandard suffixes.
- Error messages that contain an option or an option with a value are no longer corrupted.
- LD no longer does an odd address reference (trap to 4) when it encounters an unknown option letter.
- LDX and LDM no longer print a corrupted version number.

LP

The following problem with the LP handler has been corrected:

Previously, on PDP–11s with the LP11 (M7258) UNIBUS interface, the last character printed was reprinted after recovery from CPU power-fail or from an abort that occurred during printing with the printer off-line. Typically, this character was a <LF> (SET LP ENDPAG=0) or <FF> (SET LP ENDPAG=n, where n > 0).

This is corrected in RT–11 V5.6 by setting a <CR> as the last character printed.

LP/LS

- Previously, the load routine for LS corrupted the contents of memory location (vector) 350 when LS was fetched or loaded into memory. For RT–11 V5.4G, the load routine for LS functions correctly.
- CTS-300 SYSGEN now builds the LS handler for Professional 300 series processors.
- A coding error in the handler sometimes caused loss of characters when you ran RT–11 on a Professional 300 series machine. That problem has been corrected.

MM/MS/MT/MU (Magnetic Tape Handlers in General)

- Previously, a .LOOKUP operation to the last file on a magtape would fail when the file structure module (FSM) within the magtape handler used the sequence number to access the last file immediately after creation. For RT–11 V5.4D, that operation succeeds under those circumstances.
- Previously, the FSM (file structure module) functioned incorrectly when a magtape was positioned after the last file and the /POSITION:n option, where n was the number of files on that magtape, was included in a COPY command. Those circumstances incorrectly returned the error message ?PIP-F-File not found <dev:filenam.typ>. For RT–11 V5.4C, the FSM functions correctly, and the file on a magtape is found and copied under those circumstances.

MM

- Previously, the MM hardware handler for the XM monitor (MMHDX.SYS) would not work. For RT–11 V5.4C, the MMHDX.SYS handler works as documented. (You have to perform a SYSGEN to build that handler.)
- Previously, the MM handler could occasionally cause a system crash or a monitor trap to 4 condition when a hard error occurred during a write (.WRITE or .SPFUN 371) operation. For RT–11 V5.4B, the MM handler responds correctly by performing recovery or returning a hard error code under those circumstances.

MQ

- Previously, the MQ handler did not verify the validity of a job before sending a queue element. For RT–11 V5.5, MQ validates a job by checking bits in the job state word (I.STATE) before sending a queue element to the job.

- Previously, the MQ handler used only the .READx wcnt parameter value to determine the size of a data transfer. A mismatch between the wcnt parameter value for a .WRITx and a .READx request could cause data or code corruption. For RT–11 V5.4, the MQ handler uses the smaller of the .READx and .WRITx wcnt parameter values to determine the data transfer size, and memory corruption can no longer occur under those circumstances.

MS

- Version 5.5 corrects a number of problems that made the MS handler not completely reliable with multiple magtape units.
- Previously, MS did not check for the existence of a second device, and attempting I/O to a nonexistent second MS device (MS1) could cause the system to crash. For RT–11 V5.5, MS at load or fetch time, checks for the existence of a second device, and if not found, rejects any attempted I/O to that nonexistent device.
- Previously, MS could incorrectly control streaming mode on multiple TSV05 devices when both devices were used at the same time. Changes to the streaming flag made for one device could cause a change in streaming mode for both devices. For RT–11 V5.5, MS correctly maintains flag bits for individual device units.

  Also, V5.5 corrects a problem with the MS abort code that could cause problems when using TSV05 devices in streaming mode. Previously, the abort code, when called, could turn off streaming improperly and possibly corrupt memory or make the system crash.
- Previously, issuing special function SF.MOR (372) caused a new tape mounted in the device to be unusable until a .CLOSE, .REOPEN, and .LOOKUP with sequence number 0, were issued to the device. For RT–11 V5.5, special function SF.MOR no longer makes a new tape unusable under those circumstances.
- Previously, a coding error in the MS handler could cause a system crash after MS attempted to recover from a tape drive unexpectedly going off line. For RT–11 V5.4C, the MS handler code is corrected, and that situation can no longer make the system crash.
- RT–11 V5.3 corrected a problem in which a double CTRL/C issued to a program using the MS magtape while it was in motion could cause the system to hang. That problem was caused by MS attempting to dismiss queue elements already dismissed by the monitor.

MT

- Previously, when called with an .SPFUN 375 or .SPFUN 376, the MT handler for the XM monitor (MTX.SYS) incorrectly returned a negative value for the number of records not skipped. For RT–11 V5.4G, the MT handler under XM correctly returns a positive value for that number, as documented.
- Previously, the MT handler could occasionally cause a system crash or a monitor trap to 4 condition when a hard error occurred during a write (.WRITE or SPFUN 371) operation. For RT–11 V5.4B, the MT handler responds correctly by performing recovery or returning a hard error code under those circumstances.

MU

- The MU handler initializes the port interface between the processor and the TMSCP magtape controller when you use the drive for the first time after fetching or loading MU, or when a new magtape is mounted. Previously, aborting magtape operations by typing double CTRL/C during that port initialization procedure could hang your system. For RT–11 V5.4B, aborting the port initialization does not hang your system. You can, however, experience a delay of up to 90 seconds before receiving the monitor prompt after aborting port initialization.
- Previously, aborting magtape operations by typing double CTRL/C while the system was .FETCHing or LOADing the MU handler could halt your system. For RT–11 V5.4B, you can successfully abort operations under those circumstances.
- Previously, the MU handler did not report EOT (end-of-tape) until the physical end-of-tape was reached with Write, Write Tape Mark, and Space Forward operations. For RT–11 V5.4A, the MU handler correctly reports EOT when it encounters the end-of-tape mark.
- Previously, you could make the system crash if you issued the BACKUP command to MU under XM when MU was not loaded, and you then typed CTRL/C in response to the initial BUP console prompt. For RT–11 V5.4A, typing CTRL/C under those circumstances correctly exits BUP and does not make the system crash.
- Previously, you could hang the MU handler if it was loaded in memory and an error caused the unit status to change to ’off-line’. For RT–11 V5.4A, the MU handler does not hang under those circumstances.
- Through the SYSGEN process, you can generate an MU hardware handler that does not contain the file-structure module (FSM). Also, through SYSGEN, you can generate support for up to four device units. Previously, the hardware handler incorrectly marked as valid those device units not requested during SYSGEN. Attempts to access invalid device units could halt or hang the handler and crash the system. The MU handler controller revision checking program, MSCPCK.SAV, attempted to check all units not marked as invalid and therefore could halt or hang the hardware MU handler.

  For RT–11 V5.4A, the MU hardware handler correctly marks as valid only device units for which support is requested during SYSGEN.
- Previously an incorrect word count was returned on a shortened read from MU. Requesting MU to read a record of length greater than 16383 words yielded an incorrect value in the second word of the error and status block (parameter ’blk’) of the .SPFUN.

  The MU handler erroneously truncated any I/O transfer with a word count exceeding 16383 words, substituting 16383 words as the word count. After reading a short record, the number of words not read was incorrect since there was no indication that the MU handler used a word count other than the one specified by the user.

For example, attempting to read a record of maximum length (32767 words) ought to allow a calculation of actual record length as follows:

```
.SPFUN #AREA,#CHAN,#370,#BUF,#32767.,#BLK
```

Upon return, the C bit is set, and

```
BLK     contains 6 (indicating a short record was read)
BLK+2   contains n (number of words not read)
```

The number of words read, that is, the magnetic tape record length should be 32767 - n.

Because the MU handler truncated a word count greater than 16383, the number of words read is 16383 - n.

This has been corrected in RT–11 V5.6 by removing the truncation of word counts greater than 16383.

NQ/NU

- Prior to V5.6, an abort by channel (.ABTIO) performed on a channel open to an ethernet handler (NQ or NU) would result in an implicit DEALLOCATE UNIT, making the unit unusable until reallocated.

  In V5.6, this has been corrected. An abort by channel (.ABTIO) simply aborts all outstanding requests on that channel and leaves the channel allocated for further requests. The only time the channel is deallocated is when the job aborts or the user issues a DEALLOCATE UNIT spfun.
- Previously, the installation code in NU performed a self-test of the DEUNA or DELUA controller only if the controller had not successfully passed a self-test following the last power-up. For RT–11 V5.4C, the NU installation code has been modified to perform a controller self-test each time the NU handler is installed.
- A problem occurred with processing queue elements on the NQ handler's internal receive queue. Depending on the number of receive requests outstanding and the order in which they were satisfied, queue elements could become lost, causing the system to hang. For RT–11 V5.4, queue elements are correctly processed on the internal receive queue.

RK

- Previously, the RK handler did not account for the hardware differences between RKV11 and RK11 controllers regarding DMA transfer limits. Now, the RK handler returns a hard error if a transfer would have accessed memory beyond the limits addressable by the controller in use; that is, 64K bytes on the Q-bus and 256K bytes on the UNIBUS.

VM

- Previously, the system incorrectly accepted I/O requests to VM units other than VM0. Under some circumstances, files could be lost. For RT–11 Version 5.5, the system correctly accepts I/O requests to only VM0.

- Previously, a seek operation (.READ of .WRITE with wcnt=0) to a VM device could cause memory corruption and system failure. For RT–11 V5.5, a seek operation to VM works as expected (does nothing).
- Previously, the system crashed after you installed VM if the BASE was SET beyond the end of available memory. Now, the error message ?KMON-F-Invalid device installation xx:VM.SYS is displayed.
- VM failed to turn on 22-bit addressing when the boot code base address in extended memory was set at or slightly above the 18-bit/22-bit boundary. VM now correctly turns on 22-bit addressing when the boot code base address is set at or above the 18-bit/22-bit boundary. A further refinement was made to the VM code for RT–11 V5.4 that completely resolved this problem.

System Generation Corrections

The following system generation problems have been corrected in RT–11 V5:

- Previously, SYSGEN generated incorrect symbol names (system conditionals) for the second, third, and fourth CSR and VECTORs for TMSCP magtape units. For RT–11 V5.4B, SYSGEN generates correct symbol names.
- If an answer file was used as input to SYSGEN to generate a system for CTS-300, SYSGEN would not generate new printer source files. Now, SYSGEN.COM generates new source files, if necessary, when an answer file is used as input. The .CTS answer file created by SYSGEN shows the number of printers (LPNUM) requested and the CSR and vector addresses for each printer.

Multiterminal Lines

- Prior to V5.6, the base default CSR for REMOTE DL lines was dependent upon the number of LOCAL DL lines. This has been corrected for V5.6, the base default CSR for REMOTE DL lines is 175610.
- Prior to V5.6, it was possible for the user to specify a configuration of LOCAL and REMOTE DL lines which exceeded the maximum number of allowed DL lines. This has been corrected for V5.6, the maximum number of DL lines is now enforced.

  As a result, the following SYSGEN error message has been removed:

  ```
  ?SYSGEN-W-Exceeded total number of valid DL lines
  ```

System Macro Library Corrections

The following problems in the system macro library (SYSMAC) have been corrected in RT–11 V5: Previously, the following programmed requests processed the argument of R0 incorrectly:

.ABTIO
.CLOSE
.PURGE
.WAIT

These programmed requests generated incorrect results when R0 (or any address containing a reference to R0 ("@R0" for instance)) was used as the argument value.

The error was that the low byte of R0 was unconditionally cleared prior to being accessed.

This is corrected in RT–11 V5.6.

The following problems with programmed requests in the system macro library (SYSMAC) have been corrected:

.ABTIO

The .ABTIO request incorrectly aborted all a job's outstanding I/O operations; that is, it aborted all outstanding I/O on all channels. Also, all aborted I/O queue elements were lost. Now, .ABTIO correctly aborts a job's outstanding I/O operations only on the specified channel. The aborted I/O queue elements are correctly returned to the list of available queue elements.

.CALLK Example in V5.5

The CALLK example in 5.5 that used BLKMOV was seriously flawed. If you use it, you will most probably crash your system. See the *RT–11 System Macro Library Manual* for a corrected version.

.GTJB

Previously, under the single job monitor, .GTJB did not set the carry bit for invalid job IDs. This has been corrected in RT–11 V5.6.

.QSET

- When running under the XM monitor, .QSET incorrectly returned the number of queue elements requested in R0. Now, .QSET correctly returns the address of the first word beyond the allocated queue elements in R0.

.RSUM

- Previously, if a program issued more than 32767 .RSUM requests, unexpected results occurred. More than 32767 .RSUM requests caused the 16-bit word containing the count to overflow, and the .RSUM request then had the effect of a .SPND request. For RT–11 V5.4D, the resume count is held at the highest positive number, and the effect of the request does not change.

.SPFUN

- Previously, directing a .SPFUN 373 to a logically possible but physically nonexistent DM device did not return an error. Instead, it returned an incorrect size. For RT–11 V5.4B, directing a .SPFUN 373 to a logically possible but physically nonexistent DM device correctly returns an error.

.SPND

- Previously, if a program issued more than 32767 .SPND requests, unexpected results occurred. More than 32767 .SPND requests caused the 16-bit word containing the count to overflow, and the .SPND request then had the effect of a .RSUM request. For RT–11 V5.4D, the suspend count is held at the lowest negative number, and the effect of the request does not change.

.SYNCH

- Previously, the .SYNCH request was inconsistent in its treatment of argument addressing modes. The argument construction #*area* (without *pic*) worked as expected but the construction #*area,pic* did not expand properly. The construction *area,pic* had to be used. The macro was incompletely written. So as to not break possible existing source code, for RT–11 V5.5 the .SYNCH request expands both the construction #*area,pic* and *area,pic* into identical code.

System Subroutine Library Corrections

The following problems with subroutines in the system subroutine library (SYSLIB) have been corrected in RT–11 V5:

CONCAT

- A problem in the CONCAT function caused incorrect error reporting. An error was not correctly reported when both the length specified for the output string was shorter than the first input string, and the address of the first input string fell on a block boundary. CONCAT now correctly reports errors.

.DATE and File Creation Date

- Prior to V5.6, the file creation date might not reflect the actual date of creation. With V5.6, for all monitors which have timer support (TIME$R = 1) the date will now be correct (so long as the system date and time are set correctly).
- Prior to V5.6, the date returned to the .DATE programmed request might not be correct. With V5.6, for all monitors which have timer support (TIME$R = 1) the date will now be correct (so long as the system date and time are set correctly).

IASIGN

- IASIGN could suppress console (TT) carriage control if it was called before any output was sent to the console. Now, IASIGN no longer suppresses console carriage control if it is called before output is sent to the console.
- If a program did not specify carriage control, IASIGN set the default carriage control to expanded. Now, if carriage control is not specified, IASIGN causes the FORTRAN OTS to send expanded carriage control information to the terminal and line printer and unexpanded carriage control information to all other devices and files.
- IASIGN has been removed from SYSLIB and added to the FORTRAN IV distributed FORLIB and the FORTRAN–77 distributed F77OTS libraries.

ICHCPY

- Previously, ICHCPY incorrectly interpreted the optional jobblk parameter. A value specified for the first word of jobblk was incorrectly used as the location of the string rather than correctly as the first word of the string itself. For RT–11 V5.4, the ICHCPY subroutine works as documented.

IGETR

- Previously, IGETR could not be called from a FORTRAN program. Also, numerous problems occurred in the code with information mapping, sizing, and channel operations. For RT–11 V5.4, IGETR works as documented.

IGTDUS

- Previously, when IGTDUS was called with the first five parameters specified and the work parameter not specified, IGTDUS corrupted a random area of memory. That memory corruption could halt the processor or produce a trap condition. For V5.4E, that problem is corrected.
- Previously, calling IGTDUS to provide information about an MU (TMSCP) unit that was off line returned the fatal error, ?MON-F-Dir IO err <address>. For V5.4E, that problem is corrected, using the implicit .SERR error conditon handling described in this section.

ILUN

- The ILUN function called the FORTRAN IV OTS routine, $FCHNL, and returned the RT–11 channel number associated with a specific FORTRAN IV LUN. The $FCHNL routine assigned a channel number to a LUN if it was not already assigned one. When a user program looped to determine to which channels the LUNs were assigned, the program would eventually fill the channel address table and cause FORTRAN IV Error 21 (out of logical units). The ILUN function now calls a local copy of the $FCHNL routine. $FCHNL no longer assigns an available channel to an unassigned LUN.
- ILUN has been removed from SYSLIB and added to the FORTRAN IV distributed FORLIB and the FORTRAN–77 distributed F77OTS libraries.

INDEX

- Previously, INDEX incorrectly returned a 1 in the integer variable (parameter m) when the first byte of the string being sought (parameter pattrn) was a null. INDEX now correctly returns a 0 under those circumstances.

INTSET

- The linkage established by INTSET between the processor and device priorities was performed improperly, which resulted in the processor executing the interrupt at a processor priority level that was lower than the device priority level. Now, the processor executes the interrupt when the processor and device priority levels are the same.
- INTSET has been removed from SYSLIB and added to the FORTRAN IV distributed FORLIB and the FORTRAN–77 distributed F77OTS libraries.

JREAD/JREADC/JREADW

- Previously, the various JREAD subroutines did not use the device-identifier byte to validate a device. Rather, the JREAD subroutines checked the device handler's name against the string "DU". Consequently, the JREAD subroutines incorrectly returned an 'invalid device' message when used with a renamed DU handler. For RT–11 V5.4C, the JREAD subroutines correctly use the device-identifier byte to validate devices and to function correctly with renamed DU handlers.

JWRITE/JWRITC/JWRITW

- Previously, the various JWRITE subroutines did not use the device-identifier byte to validate a device. Rather, the JWRITE subroutines checked the device handler's name against the string "DU". Consequently, the JWRITE subroutines incorrectly returned an 'invalid device' message when used with a renamed DU handler. For RT–11 V5.4C, the JWRITE subroutines correctly use the device-identifier byte to validate devices and to function correctly with renamed DU handlers.

MRKT

- Previously, the MRKT system subroutine incorrectly passed the address of its ID parameter to the monitor's .MRKT programmed request service. For RT–11 V5.4F, MRKT correctly passes the value of the ID parameter to the .MRKT monitor service request.

MTIN, MTOUT

- The optional argument ocnt in the MTIN and MTOUT subroutines returned the value zero regardless of the setting of bit 6 in M.TSTS. The optional argument ocnt now returns a value equal to the number of characters transferred on return from the subroutines when bit 6 of M.TSTS is set.

SCOPY

- SCOPY did not set the err argument (to .TRUE.) when the input and output arguments specified the same array name, and the input array was correctly truncated to the value specified in the len argument. Now, the err argument is correctly set (to .TRUE.) under those circumstances.

MACRO–11 Assembler Corrections

See the *PDP–11 MACRO–11 Language Reference Manual* for corrections to the MACRO–11 assembler.

Error Severity-Level Corrections

Error level descriptions were inconsistent throughout the RT–11 documentation. They have been corrected and made consistent.

Miscellaneous Corrections

The following miscellaneous problems have been corrected prior to V5.6:

STARTF.COM The /BUFFER:256. option has been added to the FRUN SPOOL and SRUN SPOOL commands. Also, the SET USR NOSWAP command has been added before the spooler related commands, and the SET USR SWAP command has been added after the spooler related commands.

V5USER.TXT The system introductory text now includes a reference to the file V5NOTE.TXT.

Chapter 4

Chapter 4 Installation, Bootstrap, and Hardware Setup Procedures

Chapter 4

Installation, Bootstrap, and Hardware Setup Procedures

This chapter describes procedures you may need to follow, depending on your distribution kit and your hardware configuration. Additional information for this chapter appears in two files on the RT–11 distribution kit:

- V5NOTE.TXT may contain release note information formalized too late for inclusion in the printed release notes.
- CUSTOM.TXT contains a table that helps you determine the correct values and addresses to use when installing software customizations.

The *RT–11 Installation Guide* provides and describes software customizations. In the customizations, symbols are used in place of values and addresses. When you install utility software customizations, 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.)

See also Appendix B of the *RT–11 Commands Manual* for a list of printers supported by the SETUP command.

Procedures for Distribution Kits

All RT–11 distribution kits contain system software and automatic installation and verification software. However, automatic installation and verification is supported only with certain distribution kits: RL02, RX02, RX50, and TK50. If you attempt to bootstrap the automatic installation monitor, RT11AI.SYS, on a distribution kit that does not support automatic installation, the installation procedure informs you that automatic installation is not supported, and you must install your system manually.

You may need to follow one of the procedures described below, if your distribution kit does not support automatic installation or if your hardware configuration does not meet automatic installation requirements.

- When you bootstrap an RL02, RX02, RX50, or TK50 distribution kit, the automatic installation monitor, RT11AI.SYS, is bootstrapped. If your hardware configuration meets automatic installation requirements, the installation procedure tells you how to proceed. (See the *RT–11 Automatic Installation Guide* for more details on automatic installation.) If, however, your hardware configuration does not meet automatic installation requirements, the automatic installation procedure informs you that automatic installation is not supported, and you must install your system by following the procedures in the *RT–11 Installation Guide*. If your configuration does not include a line-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 9 of the RX50 distribution kit is bootable on CTI bus-based processors. Note, however, that modules relating to CTI bus-based processors are unsupported in Version 5.6.

Procedure for Printer Handlers

This section describes restrictions and workaround solutions for using printers with RT–11.

- If output to some printers is terminated prematurely, the print head may not be at the left margin. When output to the printer is resumed, printing may begin in the middle of the line. To prevent that, manually reset the printer.
- Some conditions cause data sent to a printer, using the LS handler, to be lost without warning. Those conditions are:
  - For all printers, the printer is powered down or placed off line.
  - For all printers except LA34, LA38, LA120, LS120, and serial LA180, the printer encounters an error condition while in operation.
- 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, Digital recommends 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 must send two form feeds to the printer. Include the command SET LP/LS ENDPAG=2 in your start-up command file.

  The printer will perform two form feeds, so you can remove the last file printed. The top of page will still be set correctly for printing the next file.
- For the LS handler, the default CSR address is 176500 and the default vector address is 300. If your printer is connected to a DL11/DLV11 interface with different addresses, use the SET LS CSR and SET LS VECTOR commands to modify the default values. You must then unload and load the LS handler for that command to take effect.

Recommended Terminal/Printer SET Options

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

LA50, LA75, 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
```

LA210

```
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, Digital recommends that you include the command in the appropriate STRTxx.COM file(s). However, SET LP and SET LS commands modify the permanent copy of the handler, so you need to issue them only once.

When running under a multiterminal monitor, you can set the characteristics of local terminals other than the boot-time console: Include the command SET TT CONSOLE=n in an appropriate 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.

Recommended Terminal Setup Mode Options

This section lists recommended setup modes for VT300, VT200, VT100 series; VT105; and LA120 series terminals. Terminal baud rates should be set to correspond to the computer interface. Setup options other than those discussed below should be set for operator preference or form requirements. You can permanently save setup options on VT300 and VT200 series terminals by selecting the save option at the first menu. To save permanently the setup options you select on VT100 or VT105 terminals, type SHIFT/S and on LA120 terminals type SHIFT/9.

- **VT300 Series Terminals**

  The following setup mode options are recommended for normal use on VT300 series terminals:

  Terminal mode—VT300 7-bit
  Control representation mode—interpret control
  New line mode—no new line
  Transmit flow control—XON/XOFF
  Transmit rate limiting—enabled
  Local echo—disabled
  Margin bell—off
  ⌫ key—delete
  Keypad comma—comma
  Set 8 column tabs

- **VT200 Series Terminals**

  The following setup mode options are recommended for normal use on VT200 series terminals:

  VT200 mode 7-bit controls
  No new line
  XOFF at either 64 or 125
  No local echo
  No margin bell
  Set 8 column tabs

- **VT100 Series and VT105 Terminals**

  The following setup mode options are recommended for normal use on VT100 series and VT105 terminals:

  Online
  Autorepeat on
  Margin bell off (when preparing FORTRAN programs, you may want to have the margin bell set on to notify you when column 72 is reached)
  Auto XON/XOFF on
  Wrap around on
  New line off
  Interlace off
  Parity off
  Bits per character 8
  Tabs set every 8 columns

- **LA120 Series Terminals**

  LA120 series terminals should be initialized to factory settings. Then, buffer control should be changed to small. The following factory setup options should not normally be changed:

  Line/local status
  Local echo
  Auto newline
  Parity and data bits
  Auto repeat
  Auto linefeed
  XON/XOFF
  Printer new line character

  Use the LA120 local form-feed key to make sure that the paper is positioned correctly at top of form before printing a listing.

Appendix A

Appendix A V5.6 Documentation Additions and Corrections

Appendix A

V5.6 Documentation Additions and Corrections

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*.

A Correction for the RT–11 Commands Manual

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

Replaced System Subroutine 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
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
C
        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
        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 SYSLIB Subroutine

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 (MAPPING)

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.

Form:

```
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 block. 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* 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 block is reserved to Digital.

`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 a variable that 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
```

A Correction for the 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.