RSX–11M–PLUS

Release Notes

Order Number: AA–H427H–TC

RSX-11M-PLUS Release Notes

Order No. AA-H427H-TC

Revision/Update Information:
RSX-11M-PLUS Version 4.3

Digital Equipment Corporation Maynard, Massachusetts

First Printing, October 1979
Revised, April 1982
Revised, April 1983
Revised, July 1985
Revised, September 1987
Revised, May 1988
Revised, April 1989
Revised, January 1990

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.

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227–7013.

© Digital Equipment Corporation 1979, 1982, 1983, 1985, 1987, 1988, 1989, 1990
All rights reserved. Printed in U.S.A.

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

The following are trademarks of Digital Equipment Corporation:

DDCMP
DEC
DEC/CMS
DEC/MMS
DECnet
DECUS
DECwriter
DIBOL
LA50
LA75
LN01
LN03
LSI
MicroPDP–11
Micro/RSX
MicroVMS
PDP
Professional
Q-bus
RA70
RA82
RA90
RD31
RD32
RD50
RD54
ReGIS
RK07
RL01
RL02
RSTS
RSX
RSX–11M/S
TK50
TU81
UNIBUS
VAX
VAX rdb/VMS
VAXcluster
VAX DOCUMENT
VMS
VT

S1376

This document was prepared with VAX DOCUMENT, Version 1.2.

Contents

Preface vii

Chapter 1 New Features, Restrictions, and Supplementary Information

1.1 Summary of New Features 1–1
1.2 New Hardware Support in Version 4.3 1–2
1.2.1 Support for RA70 and RA90 Disk Drives 1–2
1.2.1.1 ERRLOG Support 1–2
1.2.2 Supplementary Hardware Information 1–2
1.2.2.1 Large Disks 1–2
1.3 Software Enhancements for Version 4.3 1–2
1.3.1 Task Builder Enhancement 1–2
1.3.2 PLAS Directive 1–3
1.3.3 Support for PDP–11 C Compiler 1–3
1.3.4 Backup and Restore Utility (BRU) 1–3
1.3.4.1 Compatibility with VAX–11 RSX and VAX CoProcessor/RSX 1–3
1.3.4.2 Enhancement to BRU and the MU Device Driver 1–3
1.3.4.3 Options for BRU Qualifiers /COMPARE and /VERIFY 1–4
1.3.4.4 BRU Error Messages 1–6
1.3.5 VFY Warning Message 1–7
1.3.6 Load Rates for Writes 1–7
1.3.7 PIP Wildcard Support 1–8
1.3.8 HELLO 1–9
1.3.9 Resource Accounting 1–9
1.4 Software Problems Corrected in Version 4.3 1–9
1.4.1 Executive Support for Volume Valid 1–9
1.4.2 Device Driver 1–10
1.4.3 Terminal Driver 1–10
1.4.4 BRU 1–10
1.4.5 EXEC 1–10

1.4.6 Data Caching 1–10
1.4.7 Installing EDT 1–10
1.4.8 Installing VMR with Deferred Binding 1–11
1.4.9 SET /INQUIRE Command 1–11
1.4.10 SORT Command 1–11
1.4.11 Task Builder 1–11
1.4.11.1 Bitmask Option 1–11
1.4.11.2 Overlaid Libraries 1–11
1.4.12 LP: Device 1–11
1.4.13 Replace Control Task (RCT) Memory Allocation 1–12
1.4.14 HELLO and HELP Overlays 1–12
1.4.15 Queue Manager (QMG) 1–12
1.4.16 Set Password Task (PSW) 1–12
1.4.17 VMR LOAD and UNLOAD Commands 1–12
1.5 Software Problems and Restrictions in Version 4.3 1–12
1.5.1 Error Log File Reporting 1–12
1.5.2 Report Generator Task History Qualifier 1–13
1.6 Supplementary Software Information 1–13
1.6.1 Disk Save and Compress Utility 1–13
1.6.2 File Control Services Changes Supplementary Information 1–13
1.6.3 TKB Restrictions and Supplementary Information 1–14
1.6.3.1 OTS Fast Map Restriction 1–14

Chapter 2 Corrections to Documentation

2.1 RSX–11M–PLUS Batch and Queue Manual 2–1
2.2 RSX–11M–PLUS Command Language Manual 2–1
2.3 RSX–11M–PLUS Indirect Command Processor Manual 2–6
2.4 RSX–11M–PLUS MCR Operations Manual 2–7
2.5 RSX–11M–PLUS and Micro/RSX Crash Dump Analyzer Reference Manual 2–15
2.6 RSX–11M–PLUS and Micro/RSX Debugging Reference Manual 2–16
2.7 RSX–11M–PLUS and Micro/RSX Error Logging Manual 2–16
2.8 RSX–11M–PLUS and Micro/RSX Executive Reference Manual 2–17
2.9 RSX–11M–PLUS and Micro/RSX Guide to Writing an I/O Driver 2–24
2.10 RSX–11M–PLUS and Micro/RSX I/O Drivers Reference Manual 2–25
2.11 RSX–11M–PLUS and Micro/RSX System Management Guide 2–34
2.12 RSX–11M–PLUS and Micro/RSX System Library Routines Reference Manual 2–35
2.13 RSX–11M–PLUS and Micro/RSX Task Builder Manual 2–36
2.14 RSX–11M/M–PLUS and Micro/RSX I/O Operations Reference Manual 2–43
2.15 RSX–11M/M–PLUS RMS–11 Macro Programmer's Guide 2–43
2.16 RSX–11M/M–PLUS RMS–11 User's Guide 2–44
2.17 RSX–11M/M–PLUS RMS–11 Utilities 2–46

Chapter 3 RMS–11 Version 2.0 Software Modifications, Restrictions, and Supplementary Information

3.1 RMS–11 Corrections 3–1
3.1.1 RMS–11 Local Access 3–1
3.1.2 RMS–11 Ease of Use 3–2
3.1.3 RMS–11 Utilities Enhancements 3–2
3.2 Software Restrictions 3–2
3.2.1 RMS–11 Access Methods 3–2
3.2.2 RMS–11 Remote Access Methods 3–2
3.2.3 RMS–11 Utilities 3–2
3.3 RMS–11 Supplementary Information 3–3
3.3.1 Enhancements to RMSDES Utility 3–3
3.3.2 RMS–11 Installation 3–3
3.3.2.1 Location of RMS–11 Files on the Distribution Kit 3–3
3.3.2.2 Startup Command Procedures 3–7
3.3.2.3 Utility Configurations 3–8
3.3.2.4 Utility Command and ODL Files 3–8
3.3.2.5 Rebuilding the Utilities 3–10

Chapter 4 Layered Product Support

4.1 Correction Files 4–1
4.2 Installing Layered Products Restriction 4–2

Appendix A The Digital Equipment Computer Users Society (DECUS)

A.1 DECUS Goals A–1
A.2 DECUS Activities A–2
A.3 DECUS Membership A–2

Appendix B Reporting Problems

Appendix C Applying Corrections to Source Files

C.1 Updating an Executive Source Module C–1
C.2 Updating an MCR Source Module C–3
C.3 Updating a DCL Source Module C–5
C.4 Applying Corrections to the File Systems (F11ACP) C–6

C.5 Applying Corrections to the Reconfiguration Tasks CON and HRC C–7
C.6 Applying FCS Corrections C–8
C.7 Updating the FCS Sources C–9
C.8 Assembling FCS C–9
C.9 Replacing the FCS Object Modules C–12

Index

Tables

2–1 Number of Index File Headers 2–2
2–2 Default Number of Files—Approximation 2–2
2–3 Number of Index File Headers 2–10
2–4 Default Number of Files—Approximation 2–11
3–1 RMS–11 Files on RSX–11M–PLUS Distribution Kit 3–4
4–1 Layered Product Correction File References 4–1

Preface

Manual Objectives

The *RSX–11M–PLUS Release Notes* contains important information for using the RSX–11M–PLUS Version 4.3 operating system. Read this document before generating and using your system.

This manual contains the following categories of information:

- Descriptions of new software features as well as changes to existing software features
- Descriptions of software and hardware restrictions
- Supplementary information, which appeared in previous release notes but is still applicable
- Corrections to documentation errors and omissions
- Software modifications, restrictions, and supplementary information for RMS–11 Version 2.0
- Correction files and restrictions for layered products

Intended Audience

This manual is intended for all users of the RSX–11M–PLUS Version 4.3 operating system.

Document Structure

Chapter 1 describes software features, software and hardware restrictions, and supplementary information for the RSX–11M–PLUS Version 4.3 operating system.

Chapter 2 corrects errors and omissions in the RSX–11M–PLUS documentation set.

Chapter 3 describes software modifications, restrictions, and supplementary information for RMS–11 Version 2.0.

Chapter 4 contains correction files and restrictions for layered products.

Appendix A describes the Digital Equipment Computer Users Society (DECUS).

Appendix B describes the procedure for submitting a Software Performance Report (SPR).

Appendix C describes the procedure for applying corrections to source files.

Conventions

The following conventions are used in this manual:

Convention	Meaning
>	A right angle bracket is the default prompt for the Monitor Console Routine (MCR), which is one of the command interfaces used on RSX-11M-PLUS systems. All systems include MCR.
$	A dollar sign followed by a space is the default prompt of the Digital Command Language (DCL), which is one of the command interfaces used on RSX-11M-PLUS and Micro/RSX systems. Many systems include DCL.
xxx>	Three characters followed by a right angle bracket indicate the explicit prompt for a task, utility, or program on the system.
UPPERCASE	Uppercase letters in a command line indicate letters that must be entered as they are shown. For example, utility switches must always be entered as they are shown in format specifications.
command abbreviations	Where short forms of commands are allowed, the shortest form acceptable is represented by uppercase letters. The following example shows the minimum abbreviation allowed for the DCL command DIRECTORY: `$ DIR`
lowercase	Any command in lowercase must be substituted for. Usually the lowercase word identifies the kind of substitution expected, such as a filespec, which indicates that you should fill in a file specification. For example: `filename.filetype;version` This command indicates the values that comprise a file specification; values are substituted for each of these variables as appropriate.
/keyword, /qualifier, or /switch	A command element preceded by a slash (/) is an MCR keyword; a DCL qualifier; or a task, utility, or program switch. Keywords, qualifiers, and switches alter the action of the command they follow.
parameter	Required command fields are generally called parameters. The most common parameters are file specifications.

Convention	Meaning
,	Commas are used as separators for command line parameters and to indicate positional entries on a command line. Positional entries are those elements that must be in a certain place in the command line. Although you might omit elements that come before the desired element, the commas that separate them must still be included.
[g,m] [directory]	The convention [g,m] signifies a User Identification Code (UIC). The g is a group number and the m is a member number. The UIC identifies a user and is used mainly for controlling access to files and privileged system functions. This may also signify a User File Directory (UFD), commonly called a directory. A directory is the location of files. Other notations for directories are: [ggg,mmm], [gggmmm], [ufd], [name], and [directory]. The convention [directory] signifies a directory. Most directories have 1- to 9-character names, but some are in the same [g,m] form as the UIC. Where a UIC, UFD, or directory is required, only one set of brackets is shown (for example, [g,m]). Where the UIC, UFD, or directory is optional, two sets of brackets are shown (for example, [[g,m]]).
. . .	A vertical ellipsis shows where elements of command input or statements in an example or figure have been omitted because they are irrelevant to the point being discussed.
KEYNAME	This typeface denotes one of the keys on the terminal keyboard; for example, the RETURN key.
"print" and "type"	As these words are used in the text, the system prints and the user types. In interactive examples, what the user types is printed in red. System responses appear in black.

Chapter 1

New Features, Restrictions, and Supplementary Information

This chapter describes the new software features, lists software problems corrected in this release, lists software and hardware restrictions, and provides supplementary information for the RSX–11M–PLUS Version 4.3 operating system.

1.1 Summary of New Features

The RSX–11M–PLUS Version 4.3 operating system supports the following new features:

- New support for the RA70 and RA90 disk drives (in SA550 and SA650 arrays only).
- The Task Builder has been enhanced so tasks with memory resident overlays can be extended at runtime with the EXTK$ directive.
- The PLAS create region directive has a new option whereby the executive will create a region if, and only if, it can fit in memory. (The current create region directive will create it even if it cannot fit and, when the task tries to map this region, it will hang.)
- Support has been added to DCL to translate commands for the PDP–11 C compiler.
- BRU has been modified to produce more explicit error messages during the double buffered verify or compare operations.
- VFY now has an abort AST that displays a warning message.
- RMD now supports write load with data caching.
- PIP now supports wildcards as part of the directory specification. PIP is also 5 to 10 times faster doing wildcard lookups than in previous versions of RSX–11M–PLUS.
- HELLO now logs invalid login attempts on the console.

1.2 New Hardware Support in Version 4.3

1.2.1 Support for RA70 and RA90 Disk Drives

The following new devices are supported also:

- RA70 disk drive (in SA550 and SA650 arrays only).
- RA90 disk drive (in SA550 and SA650 arrays only).

The SA550 is an array of RA70s and RA82s and the SA650 is an array of RA70s and RA90s. Please note that RA82 disks were already supported.

1.2.1.1 ERRLOG Support

Error logging support has been added for RA70 and RA90 disk drives (in SA550 and SA650 arrays only).

1.2.2 Supplementary Hardware Information

For a complete list of hardware supported by RSX–11M–PLUS, refer to the *Software Product Description.*

1.2.2.1 Large Disks

Large disks exceed the previous Files–11 Structure Level 1 restriction on the maximum number of logical blocks. The new restriction allows a maximum of approximately 16 megablocks.

Please note that even though the RSX–11M–PLUS operating system supports a larger number of logical blocks, support for a larger maximum number of files has not been included. The maximum number of files allowed on a disk is still limited to 65,500.

1.3 Software Enhancements for Version 4.3

1.3.1 Task Builder Enhancement

Previously, the EXTK$ and EXTM$ extend task directives would not attempt a task extend if the task had memory resident overlays. This restriction has been removed as described in the following paragraphs. (See Section 5.3.32, page 5–84 in the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual* for a description of the EXTK$ directive, and Section 2–8, page 2–17 of the *RSX–11M–PLUS Version 4.3 Release Notes* for a description of the EXTM$ directive.)

The restriction is overridden if the task is built with the new Task Builder /EX task file switch. Using this switch when performing a task build will permit extension of the task when a valid EXTK$ or EXTM$ directive is issued. Normal physical allocation of memory positions the root first, then any memory resident overlays, followed by any disk resident overlays. When the /EX switch is used, the Task Builder modifies the physical allocation of memory so that the root of the task is positioned after all memory resident overlays, but before any disk resident overlays, thus allowing the task to be extended, since any extension is done to the root. For ID tasks, only the D-space root (along with the D-space stack and header) is shifted, since only the D-space root is extended.

Virtual memory is also reallocated to facilitate run-time task extensions over APR boundaries as follows: after all allocation of APRs has been carried out (for the task, shared regions, VSECTS, etc.), the task overlays are shifted up as high as possible, to allow for run-time task extension. This shift is done in one contiguous block, and applies to both memory resident and disk overlays. Note that a disk overlay can only occur up-tree from a memory resident overlay, and will share the same window as the memory resident overlay.

As an example, suppose a task is built which normally uses APR 0 for the root, APRs 1 and 2 for memory resident (and possibly disk resident) overlays, and APR 7 for a library. If the /EX switch is used, APR 0 will continue to be allocated for the root, and APR 7 will be allocated for the library as before. However, the task overlays will now be allocated to APRs 5 and 6, leaving room for task extension into APRs 1 through 4.

1.3.2 PLAS Directive

A new facility has been added to the Create Regions (CRRG$) directive.

If RS.RES is set in the region definition block prior to executing the CRRG$ directive, the region will be created only if it can immediately be made resident (without necessitating checkpointing of other resident regions). This includes checking to see that it will fit in the currently available holes. If space is not currently available, the directive will fail and return a status of IE.ALG (-84.)

This allows an application to create a shared region without the complication that the region may be permanently or semi-permanently locked out of memory by other regions, even though its size can be accommodated in the partition requested.

1.3.3 Support for PDP–11 C Compiler

DCL will support PDP–11 C Version 1.0. When PDP–11 C is installed, issuing "CC" along with any PDP–11 C command qualifiers or files will invoke the PDP–11 C compiler.

1.3.4 Backup and Restore Utility (BRU)

1.3.4.1 Compatibility with VAX–11 RSX and VAX CoProcessor/RSX

For compatibility with the VAX–11 RSX and VAX CoProcessor/RSX versions of BRU, BRU was modified to unload (IO.STC with VV$UNL set) disks used in multi-volume disk operations before prompting the user to mount the next volume.

1.3.4.2 Enhancement to BRU and the MU Device Driver

BRU has been enhanced to improve its method of verifying or comparing data backed up from a disk to an MU-type tape device, such as the TK50. MUDRV, the driver for the MU-type devices, has also been enhanced to direct the hardware to perform a compare-host function.

Previously, BRU used a single buffer to hold the data from the disk and another buffer to hold the data from the magnetic tape that it was comparing or verifying. This caused the tape to stop and restart frequently while data was being loaded into BRU's single buffer.

Now BRU uses both of its buffers to hold data from the disk, and a buffer created by the compare-host function in the hardware holds the data from the tape. BRU can now load data into one buffer while the other is in use, thus requiring fewer starts and stops of the tape. These changes increase the efficiency of compare and verify operations.

The new behavior is the default for MU-type devices only. You may override this default by using the SINGLEBUFFER option added to the existing BRU qualifiers /COMPARE and /VERIFY, which ensures that compare and verify operations are done as they were previously. For a complete description of the new options, see Section 1.3.4.3.

1.3.4.3 Options for BRU Qualifiers /COMPARE and /VERIFY

The BRU qualifiers /COMPARE and /VERIFY have two options: DOUBLEBUFFER and SINGLEBUFFER. The options can be used only with MU-type devices when verifying or comparing information from a disk to a tape. The input device must be a disk and the output device must be a tape or tapes.

Notes

1. BRU does not support the /COMPARE:DOUBLEBUFFER qualifier for backup sets that contain more than one tape. Specify the /COMPARE:SINGLEBUFFER qualifier if you must compare a backup set that consists of more than one tape.
2. During a verify operation with both MU-type and non-MU-type output devices, you must use the /VERIFY:SINGLEBUFFER qualifier.

Specifying the DOUBLEBUFFER option may improve the performance of streaming tape devices such as the TK50 cartridge tape device by double buffering I/O operations. Specifying the SINGLEBUFFER option will cause BRU to single buffer I/O operations, which was the default action for versions of BRU prior to Version 4.1.

The formats of the /COMPARE qualifier and the /VERIFY qualifier are shown next.

Formats

/COMPARE[:option]

and

/VERIFY[:option]

Parameter

option

Specifies one of the following two options:

SINGLEBUFFER — If you use the SINGLEBUFFER option for a compare or a verify operation from a disk to a tape, BRU will print the first error found and continue comparing or verifying. It prints any other errors found.

The SINGLEBUFFER option is the default for devices that are not MU-type devices.

DOUBLEBUFFER The DOUBLEBUFFER option can only be used on MU-type devices. For most system configurations with MU-type devices, a compare or verify operation from disk to tape will be more efficient if you use the DOUBLEBUFFER option. However, the error messages that are reported when you use the DOUBLEBUFFER option are less explicit than the messages reported when you use the SINGLEBUFFER option. As soon as an error occurs, BRU reports the actual error. If there is any subsequent data in the block, BRU cannot accurately compare or verify the integrity of the remainder of that block. All the remaining file IDs or file names will be listed as potential errors.

You must use an MU-type device drive that supports the compare host-data function. The DOUBLEBUFFER option is the default for MU-type devices.

Examples

```
BRU>
```

Specifies a compare operation with a list of all actual errors found in the data block or header block reported.

```
BRU>
```

Specifies a compare operation with improved performance. As soon as an error occurs, BRU reports the actual compare error. If there is any subsequent data in the block, BRU cannot accurately compare the remainder of that block. All the remaining file IDs or file names will be listed as potential errors.

```
BRU>
```

Specifies a verify operation with a list of all actual errors found in the data block or header block reported.

```
BRU>
```

Specifies a verify operation with improved performance. As soon as an error occurs, BRU reports the actual verify error. If there is any subsequent data in the block, BRU cannot verify the integrity of the remainder of that block. All the remaining file IDs or file names will be listed as potential errors.

1.3.4.4 BRU Error Messages

The following error messages may be displayed during a backup, restore, or compare operation:

BRU-*FATAL*-Device conflict

Explanation: This error message can be caused by the following problems:

- You specified the /COMPARE:DOUBLEBUFFER or /VERIFY:DOUBLEBUFFER qualifier without all MU-type output devices.
- You specified both tape and disk drives as part of either the input device specification (for a restore operation) or the output device specification (for a backup operation).

User Action: If you want to mix types of output magnetic tapes (non-MU with MU), you must specify either the /COMPARE:SINGLEBUFFER qualifier or the /VERIFY:SINGLEBUFFER qualifier.

BRU-*WARNING*-Potential data record verify error
File ID number LBN number

Explanation: A data block on the input disk device did not match a data block on the output tape device. As soon as an error occurs, BRU cannot accurately compare or verify the integrity of the remainder of that data block. All the remaining file IDs will be listed as potential errors.

User Action: Repeat the backup operation. If the backup operation fails again, repeat the operation with a different disk or tape device.

To obtain a list of all actual errors in the data block, repeat the compare operation, specifying the /COMPARE:SINGLEBUFFER qualifier.

BRU-*WARNING*-Potential file header record verify error (directory) filename.type;version

Explanation: A header record on the input disk device did not match a header record on the output tape device. As soon as an error occurs, BRU cannot accurately compare or verify the integrity of the remainder of that header block. All the remaining file names will be listed as potential errors.

User Action: Repeat the backup operation. If the backup operation fails again, repeat the operation with a different disk or tape device.

To obtain a list of all actual errors in the header block, repeat the verify operation, specifying the /COMPARE:SINGLEBUFFER qualifier.

BRU-*WARNING*-Potential file ID area or data record verify error
File ID number LBN number

Explanation: The file ID area of a data block on the input disk device did not match the file ID area of a data block on the output tape device. As soon as an error occurs, BRU is unable to accurately compare or verify the integrity of the remainder of the data block.

User Action: Repeat the backup operation. If the backup operation fails again, repeat the operation with a different disk or tape device.

To obtain a list of all actual errors in the file ID area of a data block, repeat the verify operation, specifying the /COMPARE:SINGLEBUFFER qualifier.

BRU-*FATAL*-Continuation volumes can not be double buffered

Explanation: BRU does not support the /COMPARE:DOUBLEBUFFER qualifier for backup sets that contain more than one tape.

User Action: Specify the /COMPARE:SINGLEBUFFER qualifier if you must compare a backup set that consists of more than one tape.

BRU-*WARNING*-Double buffered compare or verify not supported

Explanation: The tape driver does not support /COMPARE:DOUBLEBUFFER or /VERIFY:DOUBLEBUFFER the qualifier.

User Action: No user action is required. BRU continues the verify or compare operation in the single buffer mode.

1.3.5 VFY Warning Message

An abort AST has been added to VFY. If the user requests that VFY be aborted while a /UP, /RE, /DE, or /DH switch is being processed, a message warns the user that disk corruption may occur if the abort occurs at this time. If the user issues a second request to abort VFY, then VFY will be aborted.

1.3.6 Load Rates for Writes

The RMD C and D displays now include percentages on load rates for writes. Also, the format of these displays have been slightly modified to provide easier reading.

```
>
RSX-11M-PLUS V4.3  BL60      Cache Statistics (General)       17-JUL-89 12:56:54

Cache Region Name: CACHE     Region Size: 47440 (2500. disk blocks)

Device       Total  Hit Fail Load      Total  Hit Fail Load Defer     Total Cache
Name         Reads Rate Rate Rate     Writes Rate Rate Rate Rate    I/O Ops  Used
DU0:      304725.  97%   0%   1%    107309.  82%   0%   0%   0%    412034.  95%
DU3:           0.   0%   0%   0%         0.   0%   0%   0%   0%         0.   0%

Total     304725.  97%   0%   1%    107309.  82%   0%   0%   0%    412034.  95%

>
RSX-11M-PLUS V4.3  BL60      Cache Statistics (Detailed)      17-JUL-89 12:57:04

Device Name:  SY0:                Region Name: CACHE     Region Size: 47440
Cache Status: Active              Requests Being Cached: Dir,Ovr,Vir,Rdh

             Virtual   Readahead  Directory     Logical     Overlay       Total
Reads        122764.          0.     40793.          0.     141176.     304733.
Read Hit Rate    96%          0%        94%          0%         99%         97%
Read Load Rate    1%          0%         3%          0%          0%          1%

Read Overlap      0%          0%         2%          0%          0%          0%
Extent Too Big    0%          0%         0%          0%          0%          0%
Max Extent Size 127.        127.         3.          1.        127.

Writes        57203.                 49792.        314.                 107309.
Write Hit Rate    90%                   72%          1%                     82%
Write Overlap     0%                     0%          0%                      0%

Total I/O    179967.          0.     90585.        314.     141176.     412042.
     Failure Rates (as a % of Total):      Deferred Write Rate   0%
       Primary Pool Allocation    0%       Write Load Rate       0%
       Cache Pool Allocation      0%
       Read Load                  0%
```

1.3.7 PIP Wildcard Support

PIP has been modified to use the embedded wildcard capability of the F11ACP instead of processing embedded wildcards internally. This makes PIP 5 to 10 times faster when doing lookups of files with embedded wildcards. In addition, PIP has been modified to accept embedded wildcards in a directory specification. For example:

```
>

Directory lb:[RMS001001]
20-JUL-89 09:28

RMSRESAB.TSK;9      48.     C  30-JUN-89 13:57
RMSLBL.TSK;9        19.     C  30-JUN-89 13:57
RMSLBM.TSK;9        31.     C  30-JUN-89 13:57

Total of 98./98. blocks in 3. files


Directory lb:[RMS001054]
20-JUL-89 09:28
```

```
RMSCNV.TSK;5        176.    C  06-JUL-89 11:33
RMSIFL.TSK;5        136.    C  06-JUL-89 11:34
RMSRST.TSK;5        191.    C  06-JUL-89 11:34
RMSBCK.TSK;5        187.    C  06-JUL-89 11:34
RMSDES.TSK;5        248.    C  06-JUL-89 11:34
RMSDSP.TSK;5        183.    C  06-JUL-89 11:33
RMSDEF.TSK;5        139.    C  06-JUL-89 11:33

Total of 1260./1260. blocks in 7. files

Grand total of 1358./1358. blocks in 10. files in 2. directories
```

1.3.8 HELLO

The HELLO task now displays login attempts on the console. If the account name or UIC is found in the user account file but the failure was due to an invalid password, the failure will be reported as:

```
hh:mm:ss  Login failure USERNAME        [ggg,mmm] TTnn:
```

If the failure results from an invalid account name or UIC, the report will be:

```
hh:mm:ss  Login failure INVALID USER              TTnn:
```

1.3.9 Resource Accounting

Resource Accounting has been expanded to include a new qualifier to the START and SET commands. The qualifier is:

```
/INTRUSION:yes/no
```

When disabled, invalid login transaction blocks include the user's name and UIC, as found in the user account file, if the login failure results from a password discrepancy. If the user name or UIC is not found in the account file, the invalid login transaction block will simply list INVALID USER as the user name and the UIC field will be blank.

When enabled, invalid login transaction blocks include the user's responses to the Account: and Password: prompts. The system manager can use this information to determine which patterns are being used in an intrusion attempt.

The default is /INTRUSION:no

1.4 Software Problems Corrected in Version 4.3

1.4.1 Executive Support for Volume Valid

The Executive's support for volume valid has been changed to allow the setting of the hardware and software volume valid bit (a QIO with a function code of IO.STC and a second parameter value of VV$SET) on any volume which is mounted foreign without an ACP. Previously, only hardware volume valid could be set on volumes where software volume valid was already set.

This change is primarily intended to ease the handling of multi-volume data sets on foreign mounted devices, by allowing an application (such as the RSX backup and restore utility, BRU) to clear hardware volume valid prior to requesting the change of volumes, and then to reset it after the volume has been changed.

1.4.2 Device Driver

An offline DR: device that has a defined system data structure, for an obscure circumstance, might generate an unsolicited interrupt. The device driver, DRDRV, did not properly handle the unsolicited interrupt resulting in yet another unsolicited interrupt. The system would continuously service these unsolicited interrupts and hang. This problem has been corrected.

1.4.3 Terminal Driver

If an input interrupt occurred while executing ACD code called by the terminal driver, a loss of context would occur. This was caused by the terminal driver's use of the UCB word, U.AFLG. This problem has been corrected.

The terminal driver used to time-out if a large DMA write request was executed. This problem has been fixed.

1.4.4 BRU

In previous releases, restoring a backupset to a mounted disk with the /NOINI switch would occasionally cause the error message "BRU–*WARNING*–Error accessing file." This error would occur when the internal table of file IDs built by BRU when restoring a saveset to a mounted disk would fill up during the restore. Because this table is allocated one entry for each retrieval pointer, and zero block files contain no retrieval pointer information, BRU would zero the counter and then overwrite the file IDs at the beginning of the table. This restored data to the wrong file ID and resulted in this error.

The restore pass of BRU has been modified to exclude the file IDs of zero block files in the internal file ID table. This corrects this problem.

1.4.5 EXEC

A bug has been fixed in the EXEC where the use count of virtual terminals would be incremented based on an APR 6 mapping which was not always the Task Control Block (TCB). This prevented virtual terminals from being eliminated when they should. This problem has been fixed.

1.4.6 Data Caching

Data caching would crash the system when readahead support was turned on with a virtual disk driver. This problem has been fixed.

1.4.7 Installing EDT

If EDT was installed first with an external header (/XHR=YES or /NORESIDENT_HEADER) and then with a pool-resident header (/XHR=NO or /RESIDENT_HEADER), the task failed by trapping at location 6. This problem has been corrected.

1.4.8 Installing VMR with Deferred Binding

If a task was built against clustered resident libraries and then, at run time, attempted to create and map an address window using the CRAW$ directive, the directive would return a W.NSTS word with the WS.UNM bit incorrectly set. This problem occurred only if the task was installed from VMR with deferred binding enabled. MCR INSTALL did not display this effect. This problem has been corrected.

1.4.9 SET /INQUIRE Command

The MCR command SET /INQUIRE=TI: would hang on a VT300-series terminal when logged in via a SET /HOST command. A carriage return would have to be typed at the keyboard for the command to complete successfully. If the VT300 terminal was logged in directly rather than through a SET /HOST, the command worked correctly. This problem has been corrected.

1.4.10 SORT Command

Previously, DCL did not allow use of the /INDEXED_SEQUENTIAL qualifier to the input and the output file for the DCL SORT command. DCL has been modified to allow use of the /INDEXED_SEQUENTIAL qualifier to both the SORT input and output files.

1.4.11 Task Builder

The Task Builder would issue the fatal error message "Work file I/O error" when a task with an extremely large number of global symbols in the root was built with one or more undefined symbols. This problem has been fixed.

1.4.11.1 Bitmask Option

The bitmask option on the TKB library option commands (LIBR, RESLIB, and CLSTR) did not work with a value of 000. This prevented I&D tasks linked to libraries built with an RSX–11M–PLUS Version 4.1 or later Task Builder from accessing the library data space APRs for task use. This problem has been fixed.

1.4.11.2 Overlaid Libraries

Previously, manually-loaded overlaid libraries were not supported for use with I&D tasks. Instead, the overlaid library had to use autoload to load its overlays. If you tried to link an I&D task to an overlaid library that required manual-loading, the following error would occur:

```
TKB  -- *FATAL* -- Module name contains incompatible autoload vectors
```

Support has been added for manually-loaded overlaid libraries.

1.4.12 LP: Device

An open request on the LP: device that failed because of insufficient buffer space in the File Storage Region (FSR) for the allocation of buffers for that file would result in the spooling of a nonexistent file. This problem has been fixed.

1.4.13 Replace Control Task (RCT) Memory Allocation

The replacement control task, RCT..., was not automatically fixed in memory. This made a deadlock possible, in the case where RCT... was not resident and ...LDR issued I/O to a DU: device that could not be completed without intervention by the RCT... task. The request for RCT... was queued to ...LDR but could not be processed until the outstanding I/O request completed. The solution was to fix RCT... in memory so it would not require processing by the loader. This correction has been added to the VMR commands produced by SYSGEN.

1.4.14 HELLO and HELP Overlays

The overlay structure of the HELLO/HELP tasks (HEL.TSK and HELRES.TSK) has been modified to improve maintainability. Formerly, there was insufficient task address space to build either version to include the On-Line Debugging Tool (ODT). HELLO now includes two overlay segments plus a third segment that contains common data and routines, acting as a "root" for these two. Data and routines needed only by HELLO have been moved from the task root (HELROT) to one of the HELLO segments. The HELP overlay segment is not affected by this modification.

If you have added "user-provided" routines to your HELLO task, you will need to rebuild using the new overlay structure defined in HELBLD.ODL or HELRESBLD.ODL.

1.4.15 Queue Manager (QMG)

The print job attributes "Print adjacent to prior job" and "Print job should be held" have been included in [1,20]QMGBLD.BLD as being valid options for $JATDF. This global symbol determines the selection of default qualifiers for print jobs which are spooled by the PRINT$ macro or the .PRINT subroutine in an application, or by using the /SP switch in a command such as MAC or TKB.

Note that if QMG is rebuilt to include these options they will apply to all such spooled print jobs. The PRINT$ macro, .PRINT subroutine, and /SP switch have no provision to modify the task-built defaults.

1.4.16 Set Password Task (PSW)

The PSW (set password task) has been fixed to return the correct exit status.

1.4.17 VMR LOAD and UNLOAD Commands

An invalid VMR LOAD or UNLOAD command would cause all subsequent commands of the same type to fail. This problem has been fixed.

1.5 Software Problems and Restrictions in Version 4.3

1.5.1 Error Log File Reporting

RPT reports the version number of the LOG.ERR file in octal instead of decimal in the "Final Selection" area of the error log report.

1.5.2 Report Generator Task History Qualifier

The Report Generator Task /HISTORY qualifier does not always work when a date/time range is specified. The history summary information within the Report Generator (RPT) is valid only when there is a MOUNT (or RESET) followed by device activity followed by a DISMOUNT (or RESET) with no activity outside that interval. It is likely that devices that are mounted before Error Logging is activated and that remain mounted for the duration, or are left mounted when Error Logging is turned off, will not be properly displayed in the history summary. No record will be created for those devices in the history summary database even if they reported errors during the selection range. This occurs because those devices were not followed by a RESET record within the specified date/time range to force updates of the counts. The QIO counts may also be wrong because they were not updated after the last MOUNT record. This problem may be fixed in a later release.

1.6 Supplementary Software Information

1.6.1 Disk Save and Compress Utility

The Disk Save and Compress Utility (DSC) will not be included on the software kit after this release of RSX–11M–PLUS Version 4.3. Any data currently saved on media by the DSC utility will have to be restored using the DSC utility and then resaved using BRU. Future releases of RSX–11M–PLUS will require customers to use the save and compression features of the Backup and Restore Utility (BRU).

1.6.2 File Control Services Changes Supplementary Information

File Control Services (FCS) has been modified to support VMS ancillary control process (ACP) functions that are needed for compatibility with future RSX products and versions of VAX–11 RSX. These modifications increase the size of the FCS code that is included in the task image. Because the increase in FCS code size may affect the building of some large tasks, FCS routine versions that do not have VMS ACP support are included in the object library [1,1]NOVACPLIB.OLB and in the concatenated object module [1,1]FCSNOVACP.OBS.

Some large tasks that have a complicated Overlay Description Language (ODL) may be affected by the internal reorganization of specific FCS modules. This situation is indicated by "multiply defined symbol" errors occurring in ODL structures that were previously valid. You must correct the situation by revising the ODL structure.

The object library [1,1]NOVACPLIB.OLB can be used to build individual tasks that do not have VMS ACP support. The default routines in the system library remain unchanged. If you replace the routines in the system library (SYSLIB) with the routines from the concatenated object module [1,1]FCSNOVACP.OBS, you will affect all the tasks that are built using the system library, and you will be unable to build tasks with VMS ACP support.

If this support is not desired, it can be removed from the system library. The concatenated object module [1,1]FCSNOVACP.OBS is included on the kits and contains the affected modules with the support removed. To remove this support from your system, replace [1,1]FCSNOVACP.OBS in the system library by using the following Librarian Utility Program (LBR) command:

```
LBR>
```

If you also want to remove extended logical name support from the system library by replacing [1,1]FCSNOLOG.OBS in the system library, you must first replace FCSNOVACP.OBS. Some of the same modules are affected by both extended logical name support and VMS ACP support, and the modules are included in both of the concatenated object modules. If FCSNOLOG.OBS is replaced first, some modules in the system library will contain extended logical name support when FCSNOVACP is replaced because the modules in FCSNOVACP.OBS contain extended logical name support.

1.6.3 TKB Restrictions and Supplementary Information

The following section describes TKB restrictions and supplementary information.

1.6.3.1 OTS Fast Map Restriction

The Object Time System (OTS) Fast Map routine uses the RSX–11M–PLUS fast-mapping facility, which means that the task must not use the IOT instruction for any purpose except fast mapping. (For more information on the fast-mapping facility, see the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual.*)

Caution

Be sure to use both TKB and the system library supplied for your system to ensure that the correct OTS modules are incorporated in your task. If you use the Version 4.1 TKB with an incompatible (older) version of the system library to build an autoloaded overlaid task, the following new fatal error message is issued:

```
TKB -- *FATAL* -- Incompatible OTS module
```

Conversely, if you use the Version 4.1 system library with an older version of TKB, the OTS module FSTMAP is automatically included in any task built, but the module will not be accessible. The FSTMAP module will not affect the execution or performance of your task.

Chapter 2

Corrections to Documentation

This chapter corrects errors and omissions in the RSX–11M–PLUS and RMS–11 documentation sets. The section titles in this chapter are the titles of manuals that require correction.

2.1 RSX–11M–PLUS Batch and Queue Manual

In Section 2.6.1, page 2–15 of the *RSX–11M–PLUS Batch and Queue Operations Manual*, please add the following description of the /ALL qualifier to the SHOW QUEUE command:

/ALL
Displays information on all entries in all queues.

2.2 RSX–11M–PLUS Command Language Manual

Please add the following information to the *RSX–11M–PLUS Command Language Manual*:

- In Section 3.8.1, page 3–41, please note that the DCL command SET TERMINAL/INQUIRE supports the VT300-series terminals in VT200 mode only.
- In Section 3.8.1, page 3–42, please note that the DCL command SET TERMINAL/*model* supports the VT300-series terminals in VT200 mode only. You cannot specify VT3xx_SERIES as a terminal type.
- The following listed changes should be made to the /HEADERS:n and /MAXIMUM_FILES:n qualifiers in Section 5.11, which is the description of the DCL INITIALIZE command. Please make the following changes:
  - In place of all the text that follows the table in the explanation of the /HEADERS:n qualifier, please add the following text and table:

    The value used for /HEADERS:n is derived from the maximum number of file headers specified by /MAXIMUM_FILES:n, with consideration for pre-extending the index file if the maximum number of file headers will require a multiheader index file (see the description of /MAXIMUM_FILES:n for more information on

16,777,216 = 2^24

multiheader index files). You can use Table 2–1 to determine an appropriate value to use for the /HEADERS:n parameter.

Table 2–1: Number of Index File Headers

Maximum Number of Files	Disk Size	Number of File Headers
Maximum files less than 25588	Disks less than 209 Mb	Headers=maximum/2
Maximum files greater than 25588 and less than 51176	Disks between 210Mb and 419Mb	Headers=25593
Maximum files greater than 51176 and less than 51693	Disks between 420Mb and 423Mb	Headers=25846
Maximum files greater than 51693 and less than 65500	Disks greater than 423Mb	Headers=51693

- Please replace the second paragraph and formula after the /MAXIMUM_FILES:n qualifier with the following paragraph, table, paragraph, and formula:

  The maximum number of files varies according to disk size. You may want to use the values in Table 2–2 as the default approximations:

Table 2–2: Default Number of Files—Approximation

Disk Size	Approximate Number of Files
Up to 64Mb	Calculate the maximum number of files using the formula following the table
From 64Mb to 532Mb	Specify the number of blocks divided by 16 as the maximum number of files
Greater than 532Mb	Specify 65500 as the maximum number of files

The default used for the maximum number of files is the theoretical maximum possible number of one-block files, divided by eight. The following formula calculates the theoretical maximum number of files:

```
MAXIMUM FILES = ((N-(((N+4095.)/4096.)+9.))*127.)/258.
```

- In Section 5.12.5.1, Section 5.12.5.2, and Chapter 9, pages 5–94, 5–96, and 9–6, the format of the BACKUP qualifiers /CREATED and /MODIFIED is incorrect. The correct formats are shown next.

512 × 1024
1020 MB
512×1024 MB
8

Formats

BACKUP/CREATED/arg

and

BACKUP/MODIFIED/arg

Where *arg* can be specified as either BEFORE:(dd-mmm-yy hh:mm[:ss]) or AFTER:(dd-mmm-yy hh:mm[:ss]).

- In Section 5.12.6, page 5–104, the second example is incorrect. The correct example is shown next.

  ```
  $
  ```

  This example backs up all files on the fixed disk that were modified after 5:00 P.M. on May 14, 1987. After all the files have been copied onto the tape, BACKUP verifies the tape. If files on the tape do not verify, BACKUP returns an error message.

- The following section describes an enhancement and lists restrictions for the Backup and Restore Utility (BRU).

  The behavior of the BRU /ERROR qualifier with restore operations is unchanged. However, the qualifier can now be used with double-buffered compare and verify operations on data backed up from a disk to an MU-type device. The default number of errors is 25.

  The BRU Utility has the following restrictions:

  - The /LENGTH qualifier does not apply to cartridge tape devices, such as the TK25 and TK50, and does not function properly if used on those devices.
  - The /NEW_VERSION and /VERIFY qualifiers are mutually exclusive. They cannot be specified together in the same command line.
  - You must specify /REWIND/APPEND when placing the first backup set on a magnetic tape that contains a bootable system image.
  - The /TAPE_LABEL qualifier is ignored during a restore operation if there is a bootable system image at the beginning of the tape.

- In Section 5.13.1 of the manual, on page 5–132 immediately before the Examples heading, please add the following text:

  The DCL commands SHOW ASSIGNMENTS and SHOW LOGICALS can display at your terminal all logical assignments for a specific logical name.

  ### Formats

  ```
  SHOW ASSIGNMENTS   [logicalname][/qualifier]
  ```

  and

  ```
  SHOW LOGICALS   [logicalname][/qualifier]
  ```

  ### Parameter

  logicalname

  Specifies the logical name. You can also specify a portion of a logical name and either the generic wildcard character asterisk (*) or the specific wildcard character percent sign (%). The asterisk can represent any number of characters. The percent sign can represent only one character.

  If you specify quotation marks around the logical name, the SHOW ASSIGNMENTS command interprets the quotation marks literally, while the SHOW LOGICALS command removes the quotation marks.

Qualifiers

/ALL
/GLOBAL
/GROUP[:nG]
/LOCAL
/LOGIN
/SYSTEM
/TERMINAL[:ttnn:]

Refer to the *RSX-11M-PLUS Command Language Manual* for information on qualifiers.

- The display format of logical assignments has been modified to be more readable. The display specifies the logical name table and then lists the logical name assignments for the issuing terminal.

Examples

```
$
(Session Login Logical Table for VT2:)

     "DCM"         =    "DU:[CACHE]"
     "ER"          =    "EDT/RO "
     "GO"          =    "SET /DEF="
     "HOME"        =    "SYS$LOGIN"
     "MAP"         =    "LB:[1,54]RSX11M.MAP"
     "R"           =    "RECALL"
     "REVIEW"      =    "DU1:[REVIEWS]"
     "SYS$LOGIN"   =    "DU:[USER]" [Final]
     "SYS$PROMPT"  =    "Jim>"

(Session Local Logical Table for VT2:)

     "SYS$CLI"     =    "DCL"

$
```

Displays all the login and local assignments for the issuing terminal.

```
$
(Session Login Logical Table for VT2:)

     "SYS$LOGIN"   =    "DU:[USER]" [Final]
     "SYS$PROMPT"  =    "Jim>"

(Session Local Logical Table for VT2:)

     "SYS$CLI"     =    "DCL"

$
```

Displays the login and local assignments that begin with the letter S for the issuing terminal.

```
$
(Session Local Logical Table for VT2:)

     "SYS$CLI"     =    "DCL"

$
```

Displays the login and local assignments that match the string SYS$CLI for the issuing terminal.

```
$

$
```

Displays no matching login and local assignments for the string "SYS$CLI" for the issuing terminal.

```
$

(System Logical Table)

     "LI"         =   "LB:" [Final]
     "SS"         =   "LB:" [Final]
     "WK"         =   "LB:" [Final]

(Group 7 Logical Table)

     "GROUP"      =   "DR5:[7,43]"

(Session Login Logical Table for VT2:)

     "DCM"        =   "DU:[CACHE]"
     "ER"         =   "EDT/RO "
     "GO"         =   "SET /DEF="
     "HOME"       =   "SYS$LOGIN"
     "MAP"        =   "LB:[1,54]RSX11M.MAP"
     "R"          =   "RECALL"
     "REVIEW"     =   "DU1:[REVIEWS]"
     "SYS$LOGIN"  =   "DU:[USER]" [Final]
     "SYS$PROMPT" =   "Jim>"

(Session Local Logical Table for VT2:)

     "SYS$CLI"    =   "DCL"

$
```

Displays all the system, group, login, and local assignments for the issuing terminal.

```
$

(System Logical Table)

     "SS"         =   "LB:" [Final]

(Session Login Logical Table for VT2:)

     "SYS$LOGIN"  =   "DU:[USER]" [Final]
     "SYS$PROMPT" =   "Jim>"

(Session Local Logical Table for VT2:)

     "SYS$CLI"    =   "DCL"

$
```

Displays the system, group, login, and local assignments that begin with the letter S for the issuing terminal.

- Section 8.1.3, page 8–5, states that the DCL command SET SYSTEM/CRASH_DEVICE is valid only for pregenerated operating systems. This is no longer true; the command is now valid for all RSX–11M–PLUS operating systems.

2.3 RSX–11M–PLUS Indirect Command Processor Manual

Please make the following changes to Chapter 2 of the *RSX–11M–PLUS Indirect Command Processor Manual*:

- Change the final sentence in Section 2.4.2 to read as follows:

  See Section 2.4.6.1 for more information.

- In Section 2.6.16, page 2–61 of the manual, remove the second sentence of the third paragraph. In its place, please add the following text:

  The .FORM directive in the Indirect Command Processor (Indirect) includes support for VT200-series terminals in VT200 mode.

- Delete the following text in Section 2.6.22:

  Task not installed in system (.XQT, .WAIT)

- Replace the example in Section 2.6.26 with the following:

  The following example is from an interactive terminal session:

  ```
  >
  AT.>
  AT.>
  AT.>
  AT.>
  >;1
  AT.>
  >;2
  AT.>
  >;                          (null substring)
  AT.>
  AT.>
  >;1
  AT.>
  >;2,
  AT.>
  >@ <EOF>
  >
  ```

- An initialization test has been added in the Indirect Command Processor. This test assures that the allocation of buffers and file structures does not extend into the internal string storage region. A failure is reported as

  ```
  AT. -- Initialization error, code 6.
  ```

  Change the explanation of this error code, in Appendix A.2 by replacing the last sentence. The resulting description reads as:

  6. Unable to allocate enough space for command and data I/O buffers.

     For privileged Indirect tasks, Indirect was not installed with a large enough increment value. The system manager should remove and reinstall Indirect with a larger increment or in a larger partition.

For the nonprivileged Indirect task, the Executive directive Extend Task failed to return sufficient space for Indirect to allocate the buffers or Indirect was installed with an excessively large increment value. If it was installed with an increment value, the system manager should remove and reinstall Indirect with a smaller increment. Note that it is unnecessary to install the nonprivileged Indirect with an increment value, unless checkpointing is disabled or checkpoint space has not been allocated, because it will extend itself as required to the extent that address space is available.

2.4 RSX-11M-PLUS MCR Operations Manual

Please make the following corrections to the *RSX-11M-PLUS MCR Operations Manual*:

- In Section 3.6, page 3–12, the descriptions of the ALT keywords /RPRI and /TERM are incorrect. Delete the second paragraph, which is in parentheses, of the /RPRI keyword description. Also, the first sentence of the /TERM keyword description should read as follows:

  "Alters the priority of a task not initiated from the issuing terminal."

- In Section 3.7.7 of the manual, on page 3–22, immediately before the heading examples, please add the following text:

  The MCR command ASSIGN (ASN) can display at your terminal all logical assignments for a specific logical name.

Format

```
ASN  [logicalname][/keyword]
```

Parameter

logicalname

Specifies the logical name. You can also specify a portion of a logical name and either the generic wildcard character asterisk (*) or the specific wildcard character percent sign (%). The asterisk can represent any number of characters. The percent sign represents only one character.

If you specify quotation marks around the logical name, the ASN command interprets the quotation marks literally, while the DFL command removes the quotation marks.

Keywords

/ALL
/GBL
/GR[=n]
/SYSTEM
/TERM[=ddnn:]

Refer to the *RSX-11M-PLUS MCR Operations Manual* for information on keywords.

The display format of logical assignments has been modified to be more readable. The display specifies the logical name table and then lists the logical name assignments for the issuing terminal.

Examples

```
>

(Session Login Logical Table for VT2:)

     "DCM"         =   "DU:[CACHE]"
     "ER"          =   "EDT/RO "
     "GO"          =   "SET /DEF="
     "HOME"        =   "SYS$LOGIN"
     "MAP"         =   "LB:[1,54]RSX11M.MAP"
     "R"           =   "RECALL"
     "REVIEW"      =   "DU1:[REVIEWS]"
     "SYS$LOGIN"   =   "DU:[USER]" [Final]
     "SYS$PROMPT"  =   "Jim>"

(Session Local Logical Table for VT2:)

     "SYS$CLI"     =   "DCL"

>
```

Displays all the login and local assignments for the issuing terminal.

```
>

(Session Login Logical Table for VT2:)

     "SYS$LOGIN"   =   "DU:[USER]" [Final]
     "SYS$PROMPT"  =   "Jim>"

(Session Local Logical Table for VT2:)

     "SYS$CLI"     =   "DCL"

>
```

Displays the login and local assignments that begin with letter S for the issuing terminal.

```
>

>
```

Displays no matching login and local assignments that match the string “SYS$CLI” for the issuing terminal.

```
>

(System Logical Table)

     "LI"          =   "LB:" [Final]
     "SS"          =   "LB:" [Final]
     "WK"          =   "LB:" [Final]

(Group 7 Logical Table)

     "GROUP"       =   "DR5:[7,43]"

(Session Login Logical Table for VT2:)
```

```
     "DCM"         =   "DU:[CACHE]"
     "ER"          =   "EDT/RO "
     "GO"          =   "SET /DEF="
     "HOME"        =   "SYS$LOGIN"
     "MAP"         =   "LB:[1,54]RSX11M.MAP"
     "R"           =   "RECALL"
     "REVIEW"      =   "DU1:[REVIEWS]"
     "SYS$LOGIN"   =   "DU:[USER]" [Final]
     "SYS$PROMPT"  =   "Jim>"

(Session Local Logical Table for VT2:)

     "SYS$CLI"     =   "DCL"

>
```

Displays all the system, group, login, and local assignments for the issuing terminal.

- In Section 3.22.2 of the manual, on page 3–57 immediately before the heading Examples, please add the following text:

The MCR command DEFINE LOGICALS (DFL) can display at your terminal all logical assignments for a specific logical name.

Format

DFL [logicalname][/keyword]

Parameter

logicalname

Specifies the logical name. You can also specify a portion of a logical name and either the generic wildcard character asterisk (*) or the specific wildcard character percent sign (%). The asterisk can represent any number of characters. The percent sign represents only one character.

If you specify quotation marks around the logical name, the ASN command interprets the quotation marks literally, while the DFL command removes the quotation marks.

Keywords

/ALL
/GBL
/GR[=n]
/SYSTEM
/TERM[=ddnn:]

Refer to the *RSX–11M–PLUS MCR Operations Manual* for information on keywords.

The display format of logical assignments has been modified to be more readable. The display specifies the logical name table and then lists the logical name assignments for the issuing terminal.

Examples

```
>
(Session Local Logical Table for VT2:)

     "SYS$CLI"    =   "DCL"

>
```

Displays the login and local assignments that match the string SYS$CLI for the issuing terminal.

```
>
(System Logical Table)

     "SS"         =   "LB:" [Final]

(Session Login Logical Table for VT2:)

     "SYS$LOGIN"  =   "DU:[USER]" [Final]
     "SYS$PROMPT" =   "Jim>"

(Session Local Logical Table for VT2:)

     "SYS$CLI"    =   "DCL"

>
```

Displays the system, group, login, and local assignments that begin with the letter S for the issuing terminal.

- The following listed changes should be made to the /INF and /MXF keywords in Section 3.29, which is the description of INITIALIZE VOLUME command (INI). Please make the following changes:
  - In place of the text that explains the /INF keyword, please add the following in place of the first three paragraphs, table, and fourth paragraph:

    Specifies the number of file headers to allocate initially in the index file. The five system files (INDEXF.SYS, BITMAP.SYS, BADBLK.SYS, CORIMG.SYS, and 000000.DIR) are not included in the value for INF.

    The value used for /INF is derived from the maximum number of file headers specified by /MXF, with consideration for pre-extending the index file if the maximum number of file headers will require a multiheader index file (see the description of /MXF for more information on multiheader index files). You can use Table 2–3 to determine an appropriate value to use for the /INF parameter.

Table 2–3: Number of Index File Headers

Maximum Number of Files	Disk Size	Number of File Headers
Maximum files less than 25588	Disks less than 209 Mb	Headers=maximum/2

(continued on next page)

Table 2–3 (Cont.): Number of Index File Headers

Maximum Number of Files	Disk Size	Number of File Headers
Maximum files greater than 25588 and less than 51176	Disks between 210Mb and 419Mb	Headers=25593
Maximum files greater than 51176 and less than 51693	Disks between 420Mb and 423Mb	Headers=25846
Maximum files greater than 51693 and less than 65500	Disks greater than 423Mb	Headers=51693

- Please replace the second paragraph and formula after the /MXF keyword with the following paragraph, table, and formula:

  The maximum number of files varies according to disk size. You may want to use the values in Table 2–4 as the default approximations:

Table 2–4: Default Number of Files—Approximation

Disk Size	Approximate Number of Files
Up to 64Mb	Calculate the maximum number of files using the formula following the table
From 64Mb to 532Mb	Specify the number of blocks divided by 16 as the maximum number of files
Greater than 532Mb	Specify 65500 as the maximum number of files

The default used for the maximum number of files is the theoretical maximum possible number of one-block files, divided by eight. The following formula calculates the theoretical maximum number of files:

```
MXF = ((N-(((N+4095.)/4096.)+9.))*127.)/258.
```

- In Section 3.31, page 3–103, the description and format of the MCR command LOAD is incomplete. Please add the following sentence to the end of the second paragraph:

  “The Load command also loads the extended Executive partitions into memory.”

  The correct formats of the Load command are shown next.

Formats

```
LOA[D]  dd:[/keyword(s)]
```

and

```
LOA[D]  /EXP=expname[/keyword(s)]
```

- In Section 3.34, page 3–128, the formats of the OPE keywords /TASKD and /TASKI are incorrect. The correct formats are shown next.

Formats

/TASKD=taskname

and

/TASKI=taskname

- In Section 3.34, the second sentence in the first paragraph on page 3–129 is incorrect; it should read as follows:

  "You are limited to the specified memory region (TASK or REG)."

  Also, the second, third, and fourth examples on page 3–129 are incorrect. The correct examples are shown next.

  ```
  >
  00000000/50712
  ```

  This command opens the region TSTREG at location 0 and displays in octal the current value at the location.

  ```
  >
  00000000/50712 % ESC MCR
  ```

  This command opens location 0 of TSTREG, displays the current value in octal, and then displays the Radix-50 value in ASCII format.

  ```
  >
  00000000/50712 % ESC MCR %DCL RET
  00000002/00000 RET
  00000004/14604
  ```

  This command sequence displays the current value at location 0 in octal and then Radix-50 format, and then changes the value. The new value is also in Radix-50 format. Pressing the RETURN key enters the new value into location 0 and displays it in octal, then opens the next location in memory and displays its contents in octal.

- In Section 3.40, pages 3–138, 3–140, and 3–141, the format of the RUN keyword /UIC is incorrect in Format 3 and Format 4. The correct format is shown next.

Format

/UIC=[g,m]

- In Section 3.40, page 3–139, the definition of the keyword /UIC is incorrect. The correct definition is shown next.

/UIC

(Privileged keyword.) The User Identification Code (UIC) under which the task will be requested to run. This UIC also determines which files the task can access.

The UIC has the format [g,m], where the variables *g* and *m* specify octal numbers between 1 and 377 that represent the group and member numbers, respectively. The square brackets are required syntax.

When you specify /UIC, the UIC that you specify becomes the default and protection UICs for the task.

When you do not specify /UIC, the terminal's UIC becomes the default and protection UICs for the task. This is the default.

- In Section 3.40, page 3–139, the definitions of the parameters *taskname* and *dtime* are incorrect. The correct definitions are shown next.

  taskname
  Specifies a 1- to 6-character name of an installed task.

  dtime
  Specifies a delta time. Delta time is an increment from the current time.

- In Section 3.40, pages 3–140 and 3–141, the definition of the parameter *taskname* is incorrect. The correct definition is shown next.

  taskname
  Specifies a 1- to 6-character name of an installed task.

- In Section 3.40, page 3–142, the definition of the parameter *$* is incorrect. The correct definition is shown next.

  $

  When you specify a dollar sign ($) in the command line, the parameter *dev* defaults to LB:, and [g,m] defaults to the current library UIC (usually, [3,54]). If the task cannot be found in the library UIC, INSTALL searches the system UIC on device LB: (usually, [1,54]).

  If you do not specify a dollar sign ($), RUN searches for the task image file in the UIC (on device SY) to which the terminal requesting the task is set.

  Note

  If LB has been reassigned to another device and the system UIC for that device contains privileged tasks built for another system, your system will fail.

- In Section 3.40, pages 3–144 and 3–145, the descriptions of the RUN command keywords /PRI=number and /UIC are incorrect. The correct description is shown next.

  /PRI=number
  Specifies the priority of the task. The value range is 1 to 250_{10}, where 250 is the highest priority. Standard number conventions apply: octal by default, decimal if followed by a period.

  If a nonprivileged user specifies a priority that is greater than 50_{10}, the priority is set to 50_{10}.

  The default is /PRI=50.

/UIC=(g,m)

(Privileged keyword.) Specifies the User Identification Code (UIC) under which the task will be requested to run. The square brackets are required syntax.

If you specify a UIC, the UIC becomes the default UIC and protection UIC for the task. If you do not specify a UIC, your default UIC and protection UIC become the default UIC and protection UIC for the task.

The default is the UIC of the terminal that issues the RUN command.

- In Section 3.42, page 3–154 and page 3–169, the format of the MCR command SET /INQUIRE is incorrect. The correct format is shown next.

  Format

  /INQUIRE=term

- In Section 3.42, page 3–161, the format of the MCR command SET /CRASHDEV is incorrect. The correct format is shown next.

  Format

  SET /CRASHDEV=ddnn:[CSRaddr]

  or

  SET /CRASH_DEVICE=ddnn:[CSRaddr]

  Also, the section states that the MCR commands SET /CRASHDEV and SET /CRASH_DEVICE are valid only for pregenerated operating systems. This is no longer true; the commands are now valid for all RSX–11M–PLUS operating systems.

- In Section 3.42, page 3–163 and page 3–164, the description of the MCR command SET /DPRO is incorrect. The text incorrectly states that the parameter *protection-spec* has two formats. The parameter *protection-spec* can only be specified in the format shown next.

  Format

  [RWED,RWED,RWED,RWED]

  In Section 3.42, page 3–164, the first paragraph under the bulleted items is incorrect. The paragraph should read as follows:

  The /DPRO keyword establishes a default file protection for your current session at the terminal. When you start a new terminal session, it resets your file protection to the system default protection (or to the default protection in your account file if one was specified when your account was created). To establish a default protection code for all of your future sessions at the terminal, specify the code as a command in your login command file (LOGIN.COM).

- In Section 3.42, page 3–169, add the following information:

  The MCR command SET/INQUIRE supports the VT300-series terminals in VT200 mode only.

- In Section 3.42, page 3–185, add the following information:

The MCR command SET/TERM supports the VT300-series terminals in VT200 mode only. You cannot specify VT3xx as a terminal type.

- In Section 3.42, page 3–197, the description of the sixth example is incorrect. The correct example is shown next.

  ```
  >
  SECPOL=285.:640.:44%
  ```

  Displays the amount of available secondary pool.

- Immediately before Section 12.1.4.5 on page 12–11 of the manual, please enter the following text:

  The Peripheral Interchange Program (PIP) qualifier /DD must be used with the ampersand (&) character.

2.5 RSX–11M–PLUS and Micro/RSX Crash Dump Analyzer Reference Manual

Please make the following correction to the *RSX–11M–PLUS and Micro/RSX Crash Dump Analyzer Reference Manual*:

- Section 1.1.1 specifies that, for RSX–11M–PLUS operating systems that are not pregenerated, you must select the crash dump driver during system generation. This is no longer true; RSX–11M–PLUS now supports loadable crash dump drivers.

  You can select loadable crash dump support during system generation and choose a loadable crash driver during the system startup procedure. Crash dump devices can be DU-, DL-, MU-, MS-, and MM-type devices. Refer to the *RSX–11M–PLUS System Generation and Installation Guide* for more information on including loadable crash support in your system if it is not pregenerated.

  In addition, for systems that have loadable crash dump support included, you no longer must perform another SYSGEN to change the crash dump device or unit number; you can use either the DCL command SET SYSTEM/CRASH_DEVICE or the MCR command SET /CRASHDEV. The previous commands are now valid for all RSX–11M–PLUS operating systems. Refer to the *RSX–11M–PLUS Command Language Manual* and the *RSX–11M–PLUS MCR Operations Manual* for more information on the commands.

- Please add the following text to Section 1.1.2 of the manual:

  Crash drivers on RSX–11M–PLUS systems with 4Mb of memory have the following functions:

  — The MU and DU crash drivers on 4Mb systems keep track of the amount of memory dumped and terminate the dump when 2044K words of memory have been dumped.

  — The DL crash driver on 4Mb systems dumps 2044K words of memory.

- Section 1.1.2.1 specifies an incorrect format for the MCR command SET /CRASHDEV (or SET /CRASH_DEVICE). The correct format is shown next.

Format

```
SET /CRASHDEV=ddnn[:CSRaddr]
```

or

```
SET /CRASH_DEVICE=ddnn[:CSRaddr]
```

- Section 1.2.1 specifies that, for systems that are not pregenerated, transferring processor control to the crash dump driver depends on whether you built the Executive Debugging Tool (XDT) into your system during system generation. The information in this section is generally incorrect because RSX–11M–PLUS Version 4.0 included loadable XDT support for all RSX–11M–PLUS systems and RSX–11M–PLUS Version 4.1 included loadable crash dump support for all RSX–11M–PLUS operating systems.

  Loadable XDT support enables you to load XDT when you want to use it for debugging. System performance is improved when XDT is not part of the system. Features such as instruction decoding and automated searching of symbol addresses listed in the Executive map are included in the loadable version of XDT. Refer to the *RSX-11M-PLUS and Micro/RSX Crash Dump Analyzer Reference Manual* for more information on loadable XDT support.

2.6 RSX–11M–PLUS and Micro/RSX Debugging Reference Manual

In the *RSX-11M-PLUS and Micro/RSX Debugging Reference Manual*, Section 1.2.3.3, please remove the second paragraph. For information on how to write to supervisor-mode libraries, refer to the *RSX-11M-PLUS Release Notes*, Section 1.17.1.

2.7 RSX–11M–PLUS and Micro/RSX Error Logging Manual

Please make the following corrections to the *RSX-11M-PLUS and Micro/RSX Error Logging Manual*:

- In Table 2–2, page 2–8, the following change must be made to the device entry for the control file module ETSV05:

TSV05 /TK25	ETSV05

- In Section 3.3.3.4, the /HISTORY qualifier does not always work as described in the following text:

  "RPT generates a summary report sorted by device error history. It displays the hard and soft error count and QIO count for every volume on each device."

  The use of a date/time range may make the history summary more prone to error. The history summary information within the Report Generator (RPT) is valid only when there is a MOUNT (or RESET) followed by device activity followed by a DISMOUNT (or RESET) with no activity outside that interval. It is likely that devices that are mounted before Error Logging is activated and that remain mounted for the duration, or are left mounted when Error Logging is turned off, will not be properly displayed in the history summary. No record will be created for those devices in the history summary database even if they reported errors during the selection range. This occurs because those devices were not followed by a RESET record within the specified date/time range to force updates of the counts. The QIO counts may also be wrong because they were not updated after the last MOUNT record.

- In Example A–1, page A–4, the following change must be made to the list of acceptable device names:

```
;       TSV05 or TK25
```

2.8 RSX–11M–PLUS and Micro/RSX Executive Reference Manual

Please make the following corrections to the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual*:

- In Section 2.1, page 2–1, the tenth bullet in the list is incorrect. The following is the correct text:

  The execution of the round-robin scheduling algorithm at the end of a round-robin scheduling interval if the eligibility of the current task has changed.

- In Section 3.7.2, page 3–21, the last sentence in the paragraph is incorrect. The following is the correct text:

  R0 is returned as the status (IS.SUC or IE.ALG). If a length change was requested (for example, if the high bit of R0 was set) and the new length was set by default (for example, R2 was set to 0), R2 is returned as the new length of the mapping. In all cases R2 and R3 are destroyed by the fast-map call.

- In Section 5.13, page 5–40, the macro call for CMKT$ is incorrect. The correct macro call is shown next.

  Macro Call:

```
CMKT$    [[efn],[ast]]
```

  where:

efn	=	Event flag number
ast	=	Mark time AST address

  Macro Expansion:

```
CMKT$    52.,MRKAST
.BYTE    27.,3          ;CMTK$ MACRO DIC, DPB SIZE = 3 WORDS
.WORD    52.            ;EVENT FLAG NUMBER 52
.WORD    MRKAST         ;ADDRESS OF MARK TIME REQUEST AST ROUTINE
```

Note

The above example will cancel only the Mark Time requests that were specified with efn 52 or the AST address MRKAST. If no ast or efn parameters are specified, all Mark Time requests issued by the task are cancelled and the DPB size equals 1.

- In Section 5.36, page 5–95, additional symbols have been added to the FEAT$ directive. Please add the following symbols to Table 5-1, System Feature Symbols:

Table 5–1: System Feature Symbols

Symbol	Value	Meaning
FE$NCT	63.	System has NCT support
FE$LSD	64.	System has LUT scan in LOADR disabled
FE$PC3	65.	System supports Professional 3xx series personal computers
FE$DFB	66.	System supports deferred binding
FE$RTB	67.	Run-time binding
FE$ODB	68.	Deferred binding is overridable
FE$XDJ	69.	XDT is doing I/O via TPRs on KXJ
FE$NSY	70.	No local system disk
FE$NCO	71.	No local console
FE$RTK	72.	Remote task services
FE$RDR	73.	Remote directory storage
FE$RLG	74.	Remote logical support
FE$LDR	75.	Remote load/overlays
FE$VTL	76.	VT logins disabled
FE$ANT	77.	Automatic network startup by save
FE$NRT	78.	Network remote system
FE$EXE	79.	Task file names default to .EXE
FE$CMO	80.	Resident overlays default to 512. byte alignment

- In Section 5.64, page 5–188, the macro expansion for the RLON$ and RLOG$ directives is incorrect. The correct macro expansion is shown next.

```
RLON$ MOD,TBMSK,STATUS,LNS,LNSSZ,ENS,ENSSZ,RSIZE,RTBMOD
.BYTE     207.,10.      ;RLON$ MACRO DIC, DPB SIZE = 10(10) WORDS
.BYTE     14.           ;SUBFUNCTION VALUE (RLOG$ = 10(10))

.BYTE     MOD           ;LOGICAL NAME MODIFIER
.WORD     TBMSK         ;LOGICAL NAME TABLE INHIBIT MASK

.WORD     LNS           ;LOGICAL NAME STRING ARRAY
.WORD     LNSSZ         ;SIZE (IN BYTES) OF LOGICAL NAME STRING

.WORD     ENS           ;RETURNED EQUIVALENCE NAME ARRAY
.WORD     ENSSZ         ;SIZE (IN BYTES) OF EQUIVALENCE NAME

.WORD     RSIZE         ;LOCATION OF SIZE FOR RETURNED EQUIVALENCE NAME
```

```
.WORD     RTBMOD        ;LOCATION OF LOGICAL TABLE NUMBER (LOWER BYTE) AND
                        ;MODIFIER VALUE OF LOCATED LOGICAL NAME (HIGHER BYTE)

.WORD     STATUS        ;LOCATION OF LOGICAL NAME STATUS
```

- In Section 5.98, page 5–281, the reference to the CINT$ section in the second sentence in the first paragraph is incorrect. Replace that sentence with the following:

  For information on mapping the subroutines, see Note 2 for the description of the CINT$ (Connect to Interrupt) directive.

- In Section 5.98, page 5–282, the calculation of the addresses in Notes 3 and 4 is incorrect. The calculation should be as follows:

  n = n + 120000 + (base & 177700)

- In Section 5.99, page 5–284, an additional symbol has been added to the TFEA$ directive. Please add the following symbol to Table 5-2, Task Feature Symbols:

  Table 5–2: Task Feature Symbols

Symbol	Value	Meaning
T4$DFB	41.	Task has deferred binding

- In Section 5.100, page 5–288, the macro expansion for the TLON$ and TLOG$ directives is incorrect. The correct macro expansion is shown next.

```
TLON$ MOD,TBMSK,STATUS,LNS,LNSSZ,ENS,ENSSZ,RSIZE,RTBMOD
.BYTE     207.,10.      ;TLON$ MACRO DIC, DPB SIZE = 10(10) WORDS
.BYTE     13.           ;SUBFUNCTION VALUE (TLOG$ = 9(10))

.BYTE     MOD           ;LOGICAL NAME MODIFIER
.WORD     TBMSK         ;LOGICAL NAME TABLE INHIBIT MASK

.WORD     LNS           ;LOGICAL NAME STRING ARRAY
.WORD     LNSSZ         ;SIZE (IN BYTES) OF LOGICAL NAME STRING

.WORD     ENS           ;RETURNED EQUIVALENCE NAME ARRAY
.WORD     ENSSZ         ;SIZE (IN BYTES) OF EQUIVALENCE NAME

.WORD     RSIZE         ;LOCATION OF SIZE FOR RETURNED EQUIVALENCE NAME

.WORD     RTBMOD        ;LOCATION OF LOGICAL TABLE NUMBER (LOWER BYTE) AND
                        ;MODIFIER VALUE OF LOCATED LOGICAL NAME (HIGHER BYTE)

.WORD     STATUS        ;LOCATION OF LOGICAL NAME STATUS
```

- In Sections 5.104, 5.105, and 5.106, the definition of the FORTRAN subroutine call parameter bufadr is incorrect. The correct definition is as follows:

  bufadr
  Specifies an array containing data to be sent (must be word aligned (INTEGER*2)).

  Please note that the definition of the macro call parameter bufadr is correct and should not be changed.

- The following sections describe an Executive modification, list a restriction, and provide supplementary information.

Modification to the IOSUB.MAC Module

A modification was made to the IOSUB.MAC module to prohibit both UNIBUS Mapping Registers (UMRs) and error logging processing for IO.ATT and IO.DET functions.

Shared Regions Restriction

You cannot build a shared region that contains I- and D-space. Shared regions that are built with the /ID switch will not operate correctly. As stated in Section 10.19 of the *RSX–11M–PLUS and Micro/RSX Task Builder Manual*, you cannot use the /-HD switch and /ID switch in the same build.

Extend Task Directive

The Extend Task directive with the Active Page Register (APR) protection mask enhancement instructs the system to modify the size of the issuing task by a positive or negative increment of 32-word blocks. If the directive does not specify an increment value or if it specifies an increment value of zero, the Executive makes the issuing task's size equal to its installed size. The issuing task must be running in a system-controlled partition and cannot have any outstanding I/O when it issues the directive. The task must also be checkpointable to increase its size; if necessary, the Executive checkpoints the task and then returns the task to memory with its size modified as directed.

In a system that supports the memory management directives, the Executive does not change any current mapping assignments if the task has memory-resident overlays unless the task was built with the Task Builder /EX switch. Please refer to Section 1.3.1. However, if the task does not have memory-resident overlays, the Executive attempts to modify, by the specified number of 32-word blocks, the mapping of the task to its task region.

If the issuing task is checkpointable but has no preallocated checkpoint space available, a positive increment may require dynamic memory and extra space in a checkpoint file sufficient to contain the task.

The Extend Task directive with the APR protection mask enhancement enables you to specify a mask parameter that will prevent the extension from changing the default mapping of the task's APR mapping. This enables you to extend the data-space window of a task without changing the mapping of APRs that default to overmapping a library in instruction space.

There are several constraints on the size to which a task can extend itself using the Extend Task directive enhancement. These constraints are as follows:

- No task can extend itself beyond the maximum size set by the MCR command SET /MAXEXT or the DCL command SET EXTENSION_LIMIT or the size of the partition in which it is running. (See the *RSX–11M–PLUS MCR Operations Manual* or the *RSX–11M–PLUS Command Language Manual*.)
- A task that does not have memory-resident overlays cannot extend itself beyond 32K minus 32 words.
- A task that has preallocated checkpoint space in its task image file cannot extend itself beyond its installed size.

- A task that has memory-resident overlays cannot reduce its size below the highest window in the task partition.

Format

EXTM$ [inc],mask

Parameters

inc

Specifies a positive or negative number equal to the number of 32-word blocks by which the task size is to be extended or reduced.

mask

Specifies a mask of APRs to be protected. Bit 0 represents APR0 and bit 1 represents APR1. For example, if you specify 340, then APRs 7, 6, and 5 are protected.

Macro Expansion

```
EXTM$   40,340
.BYTE   89.,3           ;EXTK$ MACRO DIC, DPB SIZE = 3 WORDS
.WORD   40              ;EXTEND INCREMENT, 40(8) BLOCKS (1K WORDS)
.WORD   340             ;APR PROTECTION MASK
```

Local Symbol Definition

E.XTIN Extend increment (2)

DSW Return Codes

IS.SUC Successful completion.

IE.UPN Insufficient dynamic memory or insufficient space in a checkpoint file.

IE.ITS The issuing task is not running in a system-controlled partition.

IE.ALG The issuing task attempted to reduce its size to less than the size of its task header; the task tried to increase its size beyond 32K words or beyond the maximum set by the MCR command SET /MAXEXT or DCL command SET EXTENSION_LIMIT; the task tried to increase its size to the extent that one virtual address window would overlap another; the task has memory-resident overlays and it attempted to reduce its size below the highest window mapped to the task partition; or the extend would unmap a protected APR.

IE.RSU Other tasks are attached to this task partition.

IE.IOP I/O is in progress for this task partition.

IE.CKP The issuing task is not checkpointable and specified a positive integer.

IE.NSW	The task attempted to extend itself to larger than the installed size (when checkpoint space is allocated in the task).
IE.ADP	Part of the Directive Parameter Block (DPB) is out of the issuing task's address space.
IE.SDP	Directive Identification Code (DIC) or DPB size is invalid.

General Information Directive

The general information directive function Get Device Information Junior (GI.DVJ) returns information about a particular device. The device for which information is returned is determined by first performing a logical assignment (if required) and then following any redirection assignments. Device assignments are checked if the high bit in the flag's byte is clear; otherwise, no check of device assignments is made.

Format

```
GIN$   GI.DVJ, buf, siz, dev, unt
```

Parameters

GI.DVJ
Specifies GIN$ function code (18).

buf
Specifies address of buffer to receive the unit information.

siz
Specifies size of buffer. Buffer size is 1 word.

dev
Specifies device name (if blank, use task's TI:).

unt
Specifies device unit number (if high bit clear, follow assignments).

Buffer Format

-1	Device is not a mass-storage and mountable device.
-2	Device is not mounted for issuer.
Bit 0	READ access is prohibited.
Bit 1	WRITE access is prohibited.
Bit 2	CREATE access is prohibited.
Bit 3	DELETE access is prohibited.
Bit 4	Device is mounted public.

Bit 5 Device is mounted private (allocated).

Bit 6 Device is mounted foreign.

Bit 7 Foreign device has ACP.

Macro Expansion

```
GIN$    GI.DEV, DVBUF, DVSIZ, "TT, 1
.BYTE   169.,6
.WORD   GI.DEV
.WORD   DVBUF
.WORD   DVSIZ
.WORD   "TT
.WORD   1
```

DSW Return Codes

IS.SUC Successful completion.

IE.ADP Part of Directive Parameter Block (DPB) is out of task's address space.

IE.IDU The specified device does not exist, or device is a virtual terminal and issuing task is not parent or offspring.

IE.SDP Invalid function code or the DPB size is invalid.

Notes

1. If the task has the "slave" attribute, logical assignments are not checked regardless of the setting of the high bit in the fourth parameter word.
2. If bit 4 and bit 5 are off, the device may be mounted semiprivate; that is, the device may be mounted by the user but not allocated.

Creating Region Names

RSX–11M–PLUS Version 4.1 allows a region to create a region with the name GEN. In addition, RSX–11M–PLUS allows regions in region GEN to be created even if a common named GEN is installed in any main partition.

The Get Partition Parameters directive (GPRT$) uses the subroutine $SRNAM, which is used by many parts of the executive to look up region and partition names. GPRT$ looks at the main partition list first; then it looks at the common block directory.

The TKB command shown next performs a GPRT$ directive for the main partition GEN.

>

Any application that uses GPRT$ to get the parameters of a named common region gets the parameters of a main partition if the name of the common region is not unique with respect to the main partition name. All applications that use $SRNAM must be changed to attach to the region by name and then to use the Get Region Parameters directive (GREG$) to obtain the region parameters.

2.9 RSX-11M-PLUS and Micro/RSX Guide to Writing an I/O Driver

Please make the following corrections to the *RSX-11M-PLUS and Micro/RSX Guide to Writing an I/O Driver:*

- Section 1.5.5, page 1–21, illustrates how to create an executive entry point vector table. It does not indicate that there are any differences between creating a vector table for a driver and creating a vector table for a privileged task. However, the length is calculated differently in each case. Use the following statement when determining the length of the vector table for a privileged task:

  ```
  EXEVCL=<<.-EXEVEC>/2>
  ```

  Use the following statement when determining the length of the vector table for a driver:

  ```
  EXEVCL=<<<.-EXEVEC>/2>-1>
  ```

- In Section 1.5.6, page 1–22, the table entitled Callable Routine for Converting Executive References in a Driver is incorrect. The correct table is as follows:

  ```
  MOV     KINAR6,-(SP)      ; SAVE KINAR6
  MOV     @#112,R0          ; GET ADDRESS OF TABLE OF ENTRIES
  MOV     (R0),R0           ; GET ADDRESS OF APR BIAS -- FIRST WORD IN TABLE
  MOV     (R0),KINAR6       ; MAP COMMON THROUGH I-SPACE APR 6
  MOV     #EXEVEC,R3        ; POINT TO VECTOR
  MOV     #EXEVCL,R2        ; SPECIFY LENGTH OF VECTOR
  CALL    @#140004          ; TRANSLATE THE VECTOR
  MOV     (SP)+,KINAR6      ; RESTORE KINAR6
  ```

- Please add the following information to Section 7.4.16 of the manual:

  If you build a vectored driver for the system macro call GTPKT$, the following symbol must be defined in the driver code:

  ```
  VC$xx = 0
  ```

  The parameter xx represents the 2-character device mnemonic.

- Please add the following information to Section 7.4.19 of the manual:

  If you build a vectored driver for the system macro call INTSV$, the following symbol must be defined in the driver code:

  ```
  VC$xx = 0
  ```

  The parameter xx represents the 2-character device mnemonic.

- Please make the following correction to Section 8.3, page 8–24 and 8–25:

  The example of the sample driver BMDRV.MAC is incorrect because the offset I.PRM+16 must be cleared if you cannot do buffered I/O. Replace the section of code in the example with the following code:

```
;       ****************************************************************
;       *                                                              *
;       *      CONVERT TO BUFFERED I/O REQUEST                         *
;       *                                                              *
;       ****************************************************************
        MOV    R5,R3           ; COPY I/O PACKET ADDRESS BACK
;       +--------------------------------------------------------------+
;       |                                                              |
;       |      THE INPUT PARAMETERS FOR $INIBF ARE:                    |
;       |                                                              |
;       |      R3 = ADDRESS OF THE I/O PACKET TO BUFFER                |
;       |                                                              |
;       |      NO OUTPUT PARAMETERS.                                   |
;       |                                                              |
;       +--------------------------------------------------------------+
        CALL   $INIBF          ; INITIALIZE BUFFERED I/O
        BR     45$             ; SKIP CLEARING OF BUFFER ADDRESS
40$:    CLR    I.PRM+16(R3)    ; INDICATE NO BUFFERED I/O
;       ****************************************************************
;       *                                                              *
;       *      QUEUE THE CLOCK BLOCK                                   *
;       *                                                              *
;       ****************************************************************
45$:    MOV    I.PRM+14(R3),R0 ; GET ADDRESS OF CLOCK BLOCK
```

2.10 RSX–11M–PLUS and Micro/RSX I/O Drivers Reference Manual

Please make the following corrections to the *RSX–11M–PLUS and Micro/RSX I/O Drivers Reference Manual*:

The values listed for the RX33 drive in Table 4-1, page 4–2 are incorrect. The correct characteristics for the RX33 drive are as follows:

Drive	Revolutions Per Minute	Sectors	Tracks	Cylinders	Bytes/Drive	Decimal Blocks
RX33	3600	15	2	80	1,228,800	2400

Information on message-oriented communication drivers was inadvertently removed from the *RSX–11M–PLUS and Micro/RSX I/O Drivers Reference Manual.* This information is applicable only to RSX–11M–PLUS operating systems. Refer to Appendix B, Section B.3.10 for information on message-oriented I/O function codes.

The information shown next should be added to the manual.

Introduction to Message-Oriented Communication Drivers

Message-oriented communication line interfaces usually link two separate but complementary computer systems. One system must serve as the transmitting device and the other as the receiving device. Message-oriented communication line interfaces are used to transfer large blocks of data.

While character-oriented interfaces can only be accessed indirectly through the terminal driver, the DMC11 and DUP11 synchronous line interfaces allow I/O requests to be queued directly for them. These devices have drivers of their own and can be accessed by means of logical device names. You can use these names in assigning Logical Unit Numbers (LUNs) with the Assign LUN system directive at task build or with the MCR command REASSIGN.

DMC11 Synchronous Line Interface

The DMC11 synchronous line interface provides a direct memory access interface between two PDP–11 computer systems using the DDCMP line protocol, thus delivering high throughput and reliability while simplifying programming. The DMC11 supports nonprocessor request (NPR) data transfers of up to 8K words at rates of 1,000,000 baud for local operation (over coaxial cable) and 19,200 baud for remote operation (using modems). Both full- and half-duplex modes are supported. The DMC11 synchronous line interface also implements remote load detect, allowing it to reinitialize a halted computer system.

DUP11 Synchronous Line Interface

The DUP11 synchronous line interface is a single-line communications device that provides a program-controlled interface between the PDP–11 and a serial synchronous line. The PDP–11 can be interfaced with a high-speed line to perform remote batch processing, remote data collection, and remote concentration applications. Modem control is a standard feature of the DUP11 and allows using the device in a switched or dedicated configuration. The DUP11 transmits data at a maximum rate of 9600 baud; this rate is limited by modem and data set interface level converters.

The DUP11 can be programmed to accept any sync character that you define. The DUP11 incorporates hardware to perform a cyclic redundancy check (CRC).

Get LUN Information Macro

Word 2 of the buffer filled by the Get LUN Information system directive (the first characteristics word) contains the following information for message-oriented communication interfaces. A bit setting of 1 indicates that the described characteristic is true for the interfaces described in this section.

Bit	Setting	Meaning
0	0	Record-oriented device
1	0	Carriage-control device
2	0	Terminal device
3	0	File-structured device
4	0	Single-directory device
5	0	Sequential device
6	0	Mass-storage device
7	0	User-mode diagnostics supported

Bit	Setting	Meaning
8	0	Device supports 22-bit direct addressing
9	0	Unit software write locked
10	0	Input spooled device
11	0	Output spooled device
12	0	Pseudo device
13	1	Device mountable as a communications channel
14	0	Device mountable as a Files-11 volume
15	1	Device mountable

Words 3 and 4 are undefined, and word 5 has a special meaning for the DUP11 interface. Byte 0 of word 5 contains the number of sync characters to be transmitted before a synching message (for example, after line turn-around in a half-duplex operation), and byte 1 is a sync counter.

QIO$ Macro

The following sections summarize the standard and device-specific functions of the QIO$ macro that are valid for the communication interfaces.

Standard QIO$ Functions

The standard functions of the QIO$ macro that are valid for the communication devices are shown next.

Format	Function
QIO$C IO.ATT, . . .	Attach device [1]
QIO$C IO.DET, . . .	Detach device
QIO$C IO.KIL, . . .	Cancel I/O requests
QIO$C IO.RLB, . . . ,<stadd,size>	Read logical block (stripping sync)
QIO$C IO.WLB, . . . ,<stadd,size>	Write logical block (preceded by syncs)

[1] Only unmounted channels may be attached. An attempt to attach a mounted channel results in an IE.PRI status return in the I/O status doubleword.

stadd
The starting address of the data buffer (may be on a byte boundary).

size

The data buffer size in bytes (must be greater than 0).

Device-Specific QIO$ Functions

The specific functions of the QIO$ macro that are valid for the communication line interfaces are shown next.

Format	Function
QIO$C IO.FDX	Set device to full-duplex mode
QIO$C IO.HDX, . . . ,<stat,mode>	Set device to half-duplex mode
QIO$C IO.INL, . . .	Initialize device and set device characteristics
QIO$C IO.RNS, . . . ,<stadd,size>	Read logical block, without stripping sync characters (transparent mode); for DMC11, treated like IO.RLB. Not supported on DUP11.
QIO$C IO.SYN, . . . ,<syn>	Specify sync character; not applicable to DMC11
QIO$C IO.TRM, . . .	Terminate communication, disconnecting from physical channel
QIO$C IO.WNS, . . . ,<stadd,size>	Write logical block without preceding sync characters (transparent mode); for DMC11, treated like IO.WLB

stat

Specifies the station assignment (primary or secondary).

mode

Specifies the transmission mode (normal or maintenance).

stadd

Specifies the starting address of the data buffer (may be on a byte boundary).

size

Specifies the data buffer size in bytes (must be greater than 0).

syn

Specifies the sync character, expressed as an octal value.

The device-specific QIO$ functions are described in the following sections.

IO.FDX Function

The QIO$ function IO.FDX sets the mode on a DUP11 or DMC11 unit to full duplex. The IO.FDX function code can be combined (ORed) with the IO.SYN function code, if desired, to set the operational characteristics of the physical device unit.

IO.HDX Function

The QIO$ function IO.HDX sets the mode on a DUP11 or DMC11 unit to half duplex. The IO.HDX function code can be combined (ORed together) with the IO.SYN function code, if desired, to set the operational characteristics of the physical device unit.

Setting half-duplex mode on the DMC11 also involves setting the station assignment (primary/secondary) and may include selecting maintenance mode (MOP) as opposed to normal mode. The station assignment is included in the optional QIO$ parameter p1. A 0 indicates a primary station and a nonzero indicates a secondary station. The DMC11 works properly if both ends are primary stations or if there is one primary and one secondary station. It does not work if both ends are secondary stations. The optional QIO$ parameter p2 selects the mode. A 0 selects normal mode and a nonzero selects MOP mode. A DMC11 in MOP mode cannot communicate with a DMC11 in normal mode.

IO.INL and IO.TRM Functions

The QIO$ functions IO.INL and IO.TRM have the same function code but different modifier bits.

IO.INL initializes a physical device unit for use as a communications link. It turns the device on line, sets device characteristics, and ensures that the appropriate data terminal is ready.

IO.TRM disconnects the device. If the device has a dial-up interface, it also hangs up the line.

IO.RNS Function

The IO.RNS QIO$ function reads a logical block of data without stripping the sync characters that may precede the data.

IO.RLB is a similar function, which is nontransparent, in that it causes the sync characters that precede the data message to be stripped. Use IO.RLB at the start of a segmented data request, in which the block might have the following layout:

```
 ---------------------------------------------------------
|    |    |    |    |    |    |     |     |               |     |
| S  | S  | H  | H  | H  | H  | CS  | CS  |     Data      | CS  |
|    |    |    |    |    |    |     |     |               |     |
 ---------------------------------------------------------
  1    2    3    4    5    6    7     8
```

S

Specifies a sync character.

H

Specifies a header character.

CS
Specifies a validity check character.

You must strip sync characters from the beginning of a data block in this way. Stripping only at the beginning of a read operation allows a later character that happens to have the same binary value as a sync character to be read without stripping. Use IO.RLB to read a logical block with leading sync characters stripped; use IO.RNS to read the block without stripping leading sync characters.

IO.SYN Function

This QIO$ function allows the programmer to specify the sync character to be recognized when an IO.RLB or IO.WLB function is performed. IO.SYN can be combined (ORed together) with IO.HDX or with IO.FDX to set the characteristics of the physical device unit.

IO.WNS Function

This QIO$ function causes a logical block to be written with no preceding sync characters. To ensure that the two systems involved in a communication are synchronized, two or more sync characters are transmitted by one system and received by the other before any other message can be sent.

Use IO.WLB to write a block of data preceded by sync characters.

Use IO.WNS to perform a block transfer without sending sync characters first.

Programming Hints

The following sections contain important information about programming the message-oriented communication interfaces.

Transmission Validation

Because there is no way for the transmitting device to verify that the data block has successfully arrived at the receiving device unless the receiver responds, the transmitter assumes that any message that is clocked out on the line (without line or device outage) has been successfully transmitted. As soon as the receiver is able to satisfy a read request, it returns a successful status code (IS.SUC) in the I/O status block. Of course, only the task receiving the message can determine whether the message has actually been transmitted accurately.

The receiving device should be ready to receive data (with a read request) at the time the transmission is sent.

Redundancy Checking

By the nature of message-oriented communications, only the task that receives a communication can determine whether the message was received successfully. The transmitter simply transfers data, without validation of any kind. It is therefore the responsibility of the communicating tasks that use the device to check the accuracy of the transmission. A simple validity check is a checksum-type longitudinal redundancy check. A better approach to validating data is the use of a cyclic redundancy check (CRC). A CRC can be computed in software or with a hardware device, such as the KG–11 communications arithmetic option.

The DUP11 incorporates hardware to compute a CRC.

Half-Duplex and Full-Duplex Considerations

Because there is a single I/O request queue, only one QIO$ request can be performed at a time. It is therefore not possible, through QIO$s, for a device to send and receive data at the same time. Also, because timeouts are not set for receive functions, a receive QIO$ is terminated only by receiving a message from the remote system, or by issuing an IO.KIL QIO$ for the device. Therefore, if no message is transmitted by the remote system, a receive does not terminate, and no further I/O can be performed on that device until the receive is killed by issuing an IO.KIL QIO$.

You can use both half-duplex and full-duplex lines with the DMC11 and DUP11. The mode is settable by using IO.FDX for full-duplex mode and IO.HDX for half-duplex mode. In half-duplex mode, the modem signal RTS (Request To Send) is cleared after each "transmit message." In full-duplex mode, this signal is always left on. Using full-duplex mode eliminates modem delays in transmission, but requires full-duplex hardware and communication links.

The DMC11 Driver maintains both transmit operations and receive operations separately in its own internal queues. Thus, it is a full-duplex driver. There is no limit on the number of outstanding I/O requests that can be active at any given time. The DMC11 hardware, however, allows a maximum of only seven transmit operations and seven receive operations to be active at any time. The driver gives the first seven transmit operations (or receive operations) directly to the DMC11 and queues the eighth and subsequent transmit operations (or receive operations) internally until the DMC11 acknowledges a successful I/O request. When running on a MicroPDP–11/70, the driver gives only two transmit operations (or receive operations) to the DMC11 because each request requires a UNIBUS mapping register (UMR). The DMC11 driver is assigned five UMRs: one for base table(s), two for active transmit operations, and two for active receive operations.

Low-Traffic Sync Character Considerations

If message traffic on a line is low, each message sent from a communications device should be preceded by a sync train. This enables the controller to resynchronize if a message is "broken" (that is, part or all of it is lost in transmission). Correspondingly, every message received by a communications device under low-traffic conditions, when messages are not contiguous (back-to-back), should be read with an IO.RLB (read, strip sync) function. This requires that the first character in the data message itself not have the binary value of the sync character.

Powerfail with DMC11

The DMC11 currently cannot recover after a power failure because the random-access memory (RAM) in its internal microprocessor is erased when power fails. Any I/O requests outstanding at the time of a power failure return the IE.ABO status. These requests must be reissued after initializing the DMC11 (IO.INL).

Importance of IO.INL

After the type of communication line has been determined, and after IO.SYN has specified the sync character, it is extremely important that IO.INL be issued before any transfers occur. This ensures that appropriate parameters are initialized and that the interface is properly conditioned. Note that IO.INL provides the only means of setting device characteristics, such as sync character. For this reason, you should always use IO.INL immediately prior to the first transfer over a newly activated link.

Tasks sending messages to the DMC11 should begin by terminating and reinitializing the device (IO.TRM, IO.INL). Note that this causes the error IE.CNR to be returned on any I/O outstanding on the other end of the line. IO.INL must be issued after each IO.KIL (which effectively kills the DMC11), after power fail, and upon receipt of any error code.

Programming Example

The following example illustrates the initialization, the setting of device parameters, and the transmission of a block of data on a message-oriented communication device.

```
.MCALL   ALUN$S,QIO$S
.        .
.        .
.        .

ALUN$S   1,"XP,0                              ;USE LUN1 FOR DP11
QIO$S    IO.HDX!IO.SYN,<1,,,,,226>            ;SET DEVICE PARAMETERS
QIO$S    IO.INL,1                             ;PUT DEVICE ON LINE
QIO$S    IO.WLB,1,,,<TXSTS,TXAST,TXBUF,100>   ;SEND A BLOCK
.        .
.        .
.        .

TXAST:   CMPB   IS.SUC&377,@(SP)+             ;WAS DATA CLOCKED OUT
                                              ;SUCCESSFULLY?
                                              ;IF SO, SET UP FOR NEXT
BEQ      10$                                  ;BLOCK
```

Please add the following section to the manual:

Section 4.5.6 DU Driver Enhancements

The DU-type device driver (DUDRV) returns more information in the IO.RSN (Read Serial Number) function. The added information is used by the Bad Block Replacement Control Task (RCT). In addition to the volume serial number, the information returned includes the controller identification number, the hardware and software version numbers of the controller, the unit identifier (device identification number), and the hardware and software version numbers of the device. RCT sends this information to the Error Logger.

A new subfunction, IQ.SEL (Suppress Error Logging), allows you to prevent error messages from being sent to the Error Logger.

In addition, DUDRV does not request RCT unless the drive is set volume valid.

Please add the following sections to the manual:

Section 4.5.7 Modifications to DUDRV and PUCOM for Digital Storage Architecture Requirements

The Digital Storage Architecture (DSA) drivers, DUDRV and PUCOM, were modified as follows:

- The DSA driver DUDRV and the PUCOM partition now implement a controller specific I/O sequence number.

  The sequence number ensures that the DSA command reference number is unique, and it is useful in debugging problems related to DSA drivers.

- Because of the information RCT puts in the system error log file, DUDRV and PUCOM save information about controller identification in their internal data structures. The IO.RSN (Read Serial Number) function returns the controller identification number.

- DUDRV invokes a time delay before it requests the unit status of RA81 devices following a powerfail. Previously, RA81 devices did not always spin up because the driver requested the unit status from the controller too quickly.

  The time delay is defined in the module DSAPRE.MAC.

- DUDRV ensures that error logs are not requested from the DSA controller if error log support was not generated in the system. This behavior reduces the number of response packets that the drivers must process on systems without error log support.

- Information from internal data structures in the DSA driver PUCOM is passed to RCT because of the information logged by RCT in the system error log file. The information is returned by the IO.RSN (Read Volume Serial Number) function code. The format of the information is found in the DSAPRE.MAC module.

- The DSA driver DUDRV correctly determines, on entry into the powerfail recovery routine, if the recovery was for a controller or a unit.

- Offset P.SEQ in the UDADF.MAC module points to a controller-supplied sequence number and is used for RCT processing.

- DUDRV uses a bit definition (UU.SEL) in the UCBDF.MAC module to determine if error packets should be logged during bad block replacement by RCT. In addition, RCT uses the submodifier IQ.SEL in module QIOMAC.MAC to set and reset UU.SEL. The IQ.SEL submodifier only is supported by DUDRV.

Section 6.7 Modifications to MUDRV and PUCOM for Digital Storage Architecture Requirements

The Digital Storage Architecture (DSA) drivers, MUDRV and PUCOM, were modified as follows:

- The DSA driver MUDRV and the PUCOM partition now implement a controller specific I/O sequence number.

  The sequence number ensures that the DSA command reference number is unique, and it is useful in debugging problems related to DSA drivers.

- MUDRV ensures that error logs are not requested from the DSA controller if error log support was not generated in the system. This behavior reduces the number of response packets that the drivers must process on systems without error log support.

- The DSA driver MUDRV correctly determines on entry into the powerfail recovery routine if the recovery was for a controller or a unit.

2.11 RSX–11M–PLUS and Micro/RSX System Management Guide

- In Section 5.5.16, page 5–74, of the *RSX–11M–PLUS and Micro/RSX System Management Guide*, the third example is incorrect. The correct example is as follows:

  ```
  VMR>
  SECPOL=285.:640.:44%
  ```

  Displays the amount of available secondary pool.

- In Section 13.3.4 of the manual, please add the following text:

  The Bad Block Replacement Control Task (RCT) supports the latest Mass Storage Control Protocol (MSCP) disk storage architecture specification. RCT is used with MSCP controllers, such as the UDA–50, which do not perform automatic revectoring of bad blocks. Instead, these controllers rely on the RCT task to perform revectoring for them. Revectoring is the redirection of reference from an unreliable block to a reliable one.

  A controller that performs its own revectoring creates a complete error log report on the I/O it handles and sends the report to the device driver. The device driver, in turn, sends a report to the Error Logger. In this way, all messages on bad blocks appear in the error log file generated when you enter an ANALYZE/ERROR_LOG command.

- In Section 13.4 of the manual, please add the following text:

  When RCT performs revectoring, it creates the error log report and sends it to the Error Logger. A new error message is issued by the Task Termination Notification program (TKTN), which sends the following message to the console terminal whenever a nonrecoverable hardware error occurs:

  ```
  *** ddnn: -- Replace command failure
  ```

  If you receive this message, Digital recommends that you back up the media and note any errors that are reported during the backup operation. You can use the error information to determine if files are corrupted.

  In addition, RCT does not produce error log packets if a device is write-protected and RCT is unable to write to the device.

- The Resource Monitoring Display (RMD) memory display supports the following two setup commands:

  TOP=*n* Specifies the upper limit of the memory display, where *n* is the limit value in K words.

  BOTTOM=*n* Specifies the lower limit of the memory display, where *n* is the limit value in K words.

  The commands allow you to examine a specific portion of system memory and will display details of system memory, including tasks and shared regions, that do not appear in a full display of system memory.

You can enter the commands as part of the MCR command line, or you can enter the commands after you access the setup page. If you specify values for *n* that are invalid, RMD will default to a full display of system memory.

- Please add the following text to Section 15.1.4 of the manual following the DEFER_WRITES option:

  Data caching supports write loading for temporary files.

  I/O requests that can be deferred will result, if necessary, in a write load, which creates a new cache extent. Blocks in a temporary file that have the potential to be deferred will no longer have to be read before a deferred-write operation can occur. This feature will increase the number of blocks that are deferred for a task, and it will increase the performance of the deferred-write support.

- The Account File Maintenance Program (ACNT) commands ADD and MODIFY prompt you for confirmation when you press the RETURN key in response to the password field prompt or the last name (that is, Username) field prompt. Because blank values for account passwords or user names may result in security problems, Digital recommends that you do not create accounts with blank values in these fields. However, when you add or modify account fields, you may inadvertently create blank values for fields if you press the RETURN key in response to the field prompt instead of the ESCAPE key which leaves a field unchanged.

  Also, the password encryption routine has an improved handling of accounts with blank values in the user-name field or in the password field. The routine uses the user-name field and the password field to encrypt the password. Previously, the routine produced deficient results for accounts with a blank value either in the user-name field or in the password field.

  The enhancement improves security, but you are unable to log in to accounts with blank user names until you modify the user-name field, which re-encrypts the password.

2.12 RSX-11M-PLUS and Micro/RSX System Library Routines Reference Manual

Please make the following correction to the *RSX-11M-PLUS and Micro/RSX System Library Routines Reference Manual*:

In Table 6-1 on page 6-14, the text explaining file name string conversion using the X Directive is incorrect. Please replace that text with the following:

Directive	Form	Operation
X (file name string conversion)	%X	Convert Radix-50 filename string in ARGBLK to ASCII string in format "name.typ"; convert version number, if non-zero, to ASCII decimal string if decimal version support is selected in your system. Otherwise the version number is converted to ASCII octal string. If the version number is zero, no version number is put into OUTBLK. Store the results in OUTBLK.

Directive	Form	Operation
	%nX	Convert next n Radix-50 filename strings in ARGBLK to ASCII strings in format "name.typ"; convert version numbers, if non-zero, to ASCII decimal strings if decimal version support is selected in your system. Otherwise version numbers are converted to ASCII octal strings. If a version number is zero, no version is put into OUTBLK for that filename string. Store results in OUTBLK and insert tab between strings.
	%VX	Use the value in the next word in ARGBLK as a repeat count, convert specified number of Radix-50 filename strings to ASCII strings in format "name.typ"; convert version numbers, if non-zero, to ASCII decimal strings if the decimal version support is selected in your system. Otherwise version numbers are converted to ASCII octal strings. If a version number is zero, no version is put into OUTBLK for that filename string. Store results in OUTBLK and insert tab between strings.

KEY	Meaning
ARGBLK =	The argument block containing the binary data to be converted, the addresses of ASCII and extended ASCII characters or the address of a double precision value.
OUTBLK =	The output block in which $EDMSG is to store output.

2.13 RSX-11M-PLUS and Micro/RSX Task Builder Manual

Please make the following changes to the *RSX-11M-PLUS and Micro/RSX Task Builder Manual*:

- Change the fourth line of Figure 4-4 to .NAUTO as shown in the following figure:

```
  --------------------------------
 |        MOV (PC)+,-(SP)         |
 |--------------------------------|
 |  ADDRESS OF PACKET (D-SPACE)   |
 |--------------------------------|
 |          JMP @.NAUTO           |
 |--------------------------------|
 |  PC RELATIVE OFFSET TO .NAUTO  |
  --------------------------------
          I-SPACE PORTION

  --------------------------------
 |  ADDRESS OF SEGMENT DESCRIPTOR |
 |--------------------------------|
 |       ENTRY POINT ADDRESS      |
  --------------------------------
          D-SPACE PORTION
```

- Change all references to FORTRAN IV–PLUS in Chapter 5 to FORTRAN–77. FORTRAN IV–PLUS is no longer supported.
- Change all references to F4PRES in Chapter 5 to F7FRES.
- Change all references to F4P in Chapter 5 to F77.
- Add the following note to Section 5.2.9.2:

  Caution

  This command file example will work only for FMS Version 2.0.

- Replace the TKB command sequence in Section 5.4.2 with the following:

  ```
  TKB>
  TKB>
  Enter Options:
  TKB>
  TKB>
  TKB>
  >
  ```

  Or, if you use the LINK command, use the following command sequence:

  ```
  $
  Option?
  Option?
  Option?
  $
  ```

- Replace FOROTS.OLB with F77FCS.OLB in the second paragraph under the command sequence in Section 5.4.2.
- In Section 11.26, page 11–36, the format of the /INCLUDE qualifier is incorrect. The correct format is shown next.

Format

```
LINK/TAS/MAP/SYM inputfile/INCLUDE:(MOD1,MOD2,...MOD8)
```

Also, please remove note number 3. It is no longer applicable.

In note number 5, the example of the /INCLUDE qualifier is incorrect. The correct example is as follows:

```
$
->
```

- In Section 11.39, page 11–55, the description of the /SAVE qualifier is incorrect. All references to the file ATLINK.CMD should be changed to ATLINK.TMP.

  Also, please remove the clause "and it contains legitimate TKB command syntax" from the second paragraph.

- In Section 12.4, page 12–8, the definitions of the device-name and unit-num8 parameters are incorrect. The correct definitions are shown next.

  device-name
  Specifies a 2-character alphabetic device name followed by a 1- to 3-digit octal unit number.

  unit-num8
  Specifies decimal numbers indicating the logical unit numbers (LUNs). If your task uses more than six logical units, you should use the UNITS option to specify the number of logical units that your task will use.

  Also, the note is no longer applicable; please remove it.

- Please add the following text to Section 12.1.25, page 12–34 of the manual and to Section 12.1.28, page 12–37 of the manual:

  The RESSUP and SUPLIB options have a new parameter code /SW that allows you to write to data space in a supervisor-mode library when mapped to supervisor D-space with the MSDS$ directive. Use the /SW parameter code to specify read-write access when you build a task that links to a supervisor-mode library. You should use the parameter code /SV to specify read-only access.

  Note that, for the /SW parameter code, the supervisor-mode library must be installed with the /RON=NO switch.

  The formats of the /SW parameter code are shown next.

Formats

```
RESSUP   file-specification/[-]SW[apr]
```

and

```
SUPLIB   file-specification/[-]SW[apr]
```

In addition, the library flag word parameter, R$LFLG, has a new flag in label block 0. The definition of the flag is shown next.

Mask	Bit	Flag	Meaning
010000	12	LD$SMV	Include supervisor-mode vectors (1=NO)

- Please add the following text to Section 12.1.34, page 12–44 of the manual:

  For an I- and D-space task build, the task builder automatically assumes the program section specified in the VSECT option is a data program section.

- Please add the following information to Section 12.1.5 on page 12–10, Section 12.1.7 on page 12–13, and Section 12.1.21 on page 12–31 of the manual:

TKB Supplementary Information

TKB now allocates both data space APRs and instruction space APRs for libraries. At offset 404, label block 0 contains the additional field L$BAPR, which contains the data space APRs that the task or library requires. (Instruction space APR allocation information is stored in field $APRMP in psect $$TSKP. For more information on psect $$TSKP, refer to Appendix E in the *RSX-11M-PLUS and Micro/RSX Task Builder Manual*.)

When you build a library by using the TKB switch /LI, the default action is to allocate the corresponding instruction and data space APRs for the library. However, you can reserve specific data space APRs by using the new /LI subswitch /LI[:bitmask]. The appropriate bits in the bit mask should be set to specify the data space APRs that you want.

If you build a task that links to a library that was built with an earlier version of TKB, only the instruction space APRs are allocated for the library. Tasks that link to libraries built with TKB Version 4.1 have the specified library data space APRs reserved. However, you can override the library data space APR reservations by using the new bit mask option with the LIBR, RESLIB, and CLSTR options.

The formats for these options are shown next.

Formats

```
LIBR =name:accesscode[:baseAPR[:bitmask]]
```

and

```
RESLIB =file/accesscode[:baseAPR[:bitmask]]
```

and

```
CLSTR =lib1,lib2...libn:accesscode[:baseAPR[:bitmask]]
```

The appropriate bits in the bit mask should be set to reserve the desired data space APRs. If you do not want to reserve data space APRs for your library, you should specify a bit mask of 000. Note that for clusters any data space APR reservation applies to all the libraries in the cluster. The bit mask for a position-independent code (PIC) library is shifted the same amount as the library. For example, if the bit mask for a

2-APR PIC library is 200 and the library is placed in APR 5 and APR 6, the bit mask is changed to 100.

The bit mask uses the same format as the mask for the new EXTM$ directive: bit 0 represents APR0 and bit 1 represents APR1 (refer to Section 1.13.3.1). For example, if you specify 340, the APRs 7, 6, and 5 are reserved.

- In Appendix B, page B–6, the description of the Label Block Group is incorrect. Please remove the last sentence in the second paragraph, which states the following: "The LBLDF$ macro on your system will have the correct offsets."

  Also, remove the third paragraph and replace it with the following paragraph:

  The LBLDF$ macro defines the label block offsets for a task that is built on a RSX–11M operating system. Tasks that are built on RSX–11M–PLUS systems have an additional eight library entries that are inserted in the label block after the entries at offset L$BLIB. Because of this, the label block offsets from L$BPRI to L$BDMZ must be adjusted by the size of the additional entries. Label block offset LBLDF$ defines the symbol $LBXL (label block extra length), which determines the size of the additional entries. Therefore, for tasks built on RSX–11M–PLUS operating systems (L$BSYS=4), you must add $LBXL to the offsets from L$BPRI to L$BDMZ in order to determine the true offset.

- In Appendix B, Figure B–5: Table Block 0–Task and Resident Library Data is incorrect. The correct figure is shown next.

Label	Offset	
L$BTSK	0	Task
	2	Name
L$BPAR	4	Task
	6	Partition
L$BSA	10	Base Address of Task
L$BHGV	12	Highest Window 0 Virtual Address
L$BMXV	14	Highest Virtual Address in Task
L$BLDZ	16	Load Size in 64-Byte Blocks
L$BMXZ	20	Maximum Size in 64-Byte Blocks
L$BOFF	22	Task Offset into Partition
L$BWND/L$BSYS	24	System I.D. Number of Window Blocks*
L$BSEG	26	Size of Overlay Segment Descriptors
L$BFLG	30	Task Flag Word
L$BDAT	32	Task Creation Date --- Year
	34	--- Month
	36	--- Day
L$BLIB	40	Library/Common
	42	Name
	44	Base Address of Library
	46	Highest Address in First Library Window
	50	Highest Address in Library
	52	Library Load Size (64-Byte Blocks)
	54	Library Maximum Size (64-Byte Blocks)
	56	Library Offset into Region
	60	Number of Library Window Blocks
	62	Size of Library Segment Descriptors
	64	Library Flag Word
	66	Library Creation Date --- Year
	70	--- Month
	72	--- Day
	:	: :
	344	0
L$BPRI**	346	Task Priority
L$BXFR**	350	Task Transfer Address
L$BEXT**	352	Task Extension (64-Byte Blocks)
L$BSGL**	354	Block Number of Segment Load List
L$BHRB**	356	Block Number of Header
L$BBLK**	360	Number of Blocks in Label
L$BLUN**	362	Number of Logical Units
L$BROB**	364	Relative Block of R-O Image
L$BROL**	366	R/O Load Size
L$BRDL**	370	R/O Data Size in 32-Word Blocks
L$BHDB**	372	Relative Block Number of Data Header
L$BDHV**	374	High Virtual Address of Data Window 1
L$BDMV**	376	High Virtual Address of Data
L$BDLZ**	400	Load Size of Data
L$BDMZ**	402	Maximum Size of Data
L$BAPR	404	Library D-Space Reservations
	:	: :
L$BFL2	772	Second Task Flags Word
L$BLRL	774	Label Block Revision Number
	776	AME (Must Be 0)

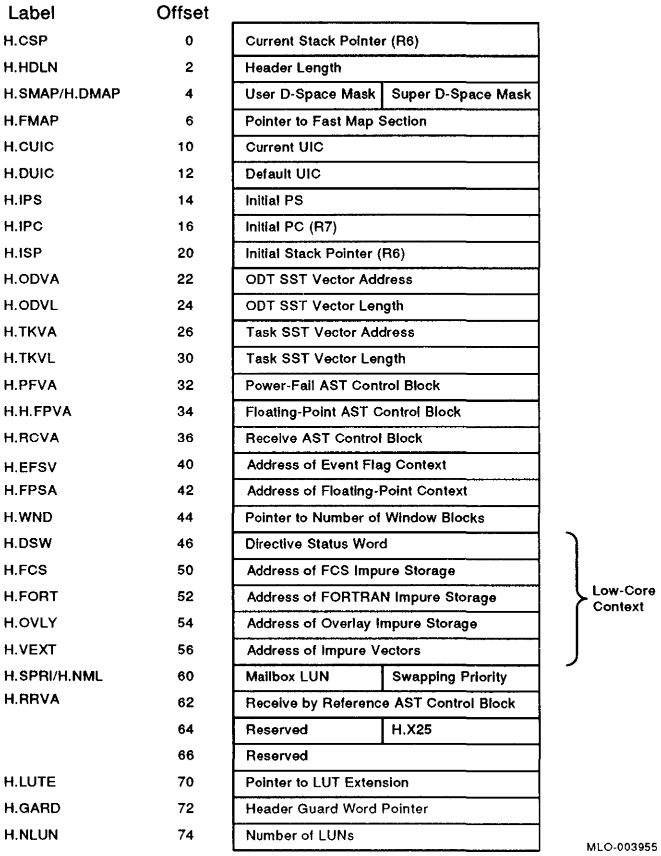
Library Entry Offset		
0		Library Request (Maximum of Seven 14-Word Entries in RSX-11M Systems and Maximum of Fifteen 14-Word Entries in RSX-11M-PLUS Systems)
2	R$LNAM	
4	R$LSA	
6	R$LHGV	
10	R$LMXV	
12	R$LLDZ	
14	R$LMXZ	
16	R$LOFF	
20	R$LWND	
22	R$LSEG	
24	R$LFLG	
26	R$LDAT	
30		
32		

* Less library window blocks.

** If parameter L$BSYS is 4, you must add $LBXL($340_8$) to the offset to determine the true offset.

MLO-003954

- In Appendix B, Figure B–8: Task Header, Fixed Part is incorrect. The correct figure is shown next.



Label	Offset			
H.CSP	0	Current Stack Pointer (R6)		
H.HDLN	2	Header Length		
H.SMAP/H.DMAP	4	User D-Space Mask	Super D-Space Mask	
H.FMAP	6	Pointer to Fast Map Section		
H.CUIC	10	Current UIC		
H.DUIC	12	Default UIC		
H.IPS	14	Initial PS		
H.IPC	16	Initial PC (R7)		
H.ISP	20	Initial Stack Pointer (R6)		
H.ODVA	22	ODT SST Vector Address		
H.ODVL	24	ODT SST Vector Length		
H.TKVA	26	Task SST Vector Address		
H.TKVL	30	Task SST Vector Length		
H.PFVA	32	Power-Fail AST Control Block		
H.H.FPVA	34	Floating-Point AST Control Block		
H.RCVA	36	Receive AST Control Block		
H.EFSV	40	Address of Event Flag Context		
H.FPSA	42	Address of Floating-Point Context		
H.WND	44	Pointer to Number of Window Blocks		
H.DSW	46	Directive Status Word		Low-Core Context
H.FCS	50	Address of FCS Impure Storage		
H.FORT	52	Address of FORTRAN Impure Storage		
H.OVLY	54	Address of Overlay Impure Storage		
H.VEXT	56	Address of Impure Vectors		
H.SPRI/H.NML	60	Mailbox LUN	Swapping Priority	
H.RRVA	62	Receive by Reference AST Control Block		
	64	Reserved	H.X25	
	66	Reserved		
H.LUTE	70	Pointer to LUT Extension		
H.GARD	72	Header Guard Word Pointer		
H.NLUN	74	Number of LUNs		

- In Appendix H, page H–2, the explanation for the "Illegal APR reservation" is not complete. Please make the following correction:

Illegal APR reservation

Explanation: An Active Page Register (APR) specified either with the /LI switch or in a COMMON, LIBR, RESCOM, or RESLIB option is outside the range 0 to 7.

2.14 RSX-11M/M-PLUS and Micro/RSX I/O Operations Reference Manual

Please make the following correction to the table of normal error returns and no error returns in Section 4.1:

Normal Error Return (Carry Bit and F.ERR)	No Error Return
.ASCPP	.RDFDR
.PARSE	.WDFDR
.PRSDV	.RDFUI
.PRSDV	.WDFUI
.PRSFN	.RDFFP
.ASLUN	.WDFFP
.FIND	.RFOWN
.ENTER	.WFOWN
.REMOV	.PPASC
.GTDIR	.MARK
.PGTDID	
.POINT	
.POSRC	
.POSIT	
.XQUIO	
.RENAM	
.EXTND	
.TRNCL	

2.15 RSX-11M/M-PLUS RMS-11 Macro Programmer's Guide

Please make the following corrections to the *RSX-11M/M-PLUS RMS-11 Macro Programmer's Guide*:

- In Section 2.3, the argument for P$BUF is incorrectly given as "bufcount." The correct argument is "iopoolsize," as discussed in Section 2.3.4.
- In Section 5.19, the last paragraph incorrectly describes the use of the FID field in the NAM block. It should read as follows:

  "If this value is nonzero..."

- In Table 6–2, page 6–14, the value for the symbol XB$DAT is incorrectly given as 000003. The correct value is 000002.
- In Appendix A, page A–8, please add the following sentence:

  An attempt to insert a record that is too small to contain the whole primary key field may also cause the error ER$KEY.
- In Section A.1, add the following text to the description of the error ER$MRS:

  Or, the sum of the fixed-length record size and the record overhead exceeds the bucket size. Or, No Span Blocks has been selected with an invalid total record size.
- In Section A.1, add the following text to the description of the error ER$NOD:

  Or, the remote node rejected the operation. (STV contains the Network Services Protocol (NSP) code and can be found in Appendix C of the *DECnet-RSX Programmer's Reference Manual.*)
- In Section A.1, add the following text to the description of the error ER$FUL:

ER$FUL, Device or File Allocation Failure

```
Octal: 176360
Decimal: -784
```

Explanation: The specified device or directory does not have enough room for file creation or extension. In the case of a contiguous request, it is also possible that there is not enough contiguous space on the device.

2.16 RSX-11M/M-PLUS RMS-11 User's Guide

Please add the following corrections to the *RSX-11M-PLUS RMS-11 User's Guide*:

- In Section 2.2.3.3, please add the following information to the discussion of deadlock:

  An application should use multistream rather than multichannel access to write to the same indexed file. When RMS–11 updates an RRV in a bucket that is currently locked, it must wait for that lock to be released. Control will not be returned to the program until this release occurs. Deadlock will occur when the lock is held on another channel within the same program; however, RMS–11 can update an RRV in a bucket that is locked on another stream within the same program. See your programming language documentation for details on the implementation of multistreaming.
- In Section 6.2.4, please add the following note to the discussion of writing a record:

  In the event that the record includes a partial alternate key but is not large enough to include space for the full alternate key field, RMS–11 will act as follows:

  RMS–11 will treat the alternate key as if it were not present in the record, making no entry in the alternate key index structure.
- According to the *RSX-11M/M-PLUS Macro Programmer's Guide*, RMS–11 cannot perform an UPDATE operation on an alternate key with the key characteristics CHANGES and NODUPLICATES. This description is misleading. To clarify the description, please add the following information to Chapter 6, Section 6.2.5.2:

Although RMS–11 does not support the CHANGES and NODUPLICATES combination, it does not prevent you from performing an UPDATE operation on an alternate key with these characteristics. When an update causes a duplicate of an alternate key, RMS–11 returns the completion code ER$DUP. However, it does not terminate the UPDATE operation. Instead, RMS–11 updates the primary data level for the record without updating the alternate index. As a result, the file contains duplicates of the alternate key.

To prevent RMS–11 from creating duplicates when you make changes on alternate keys, modify your application as follows:

1. Create the file with the key characteristics DUPLICATES and CHANGES.
2. To disallow duplicates, perform a FIND operation on each alternate key. Then, perform an UPDATE operation on the modified record.

- In Section 6.3, please add the following information to the discussion of contiguous areas:

  You will gain a small benefit by setting areas to contiguous on a noncontiguous multiarea file; however, RMS–11 cannot determine if those areas remain contiguous. Consequently, RMSDSP and RMSDES will display them as noncontiguous. As long as the areas are preallocated, they will behave like contiguous areas; as soon as they need to be extended, they will behave like noncontiguous areas.

- In Section 8.1.2.1, in the discussion of task building against the RMS–11 resident library, incorrect syntax is documented for the cluster option in the Task Builder command file. The correct syntax is as follows:

  ```
  CLSTR = RMSRES,DAPRES:RO
  ```

- In Section 8.1.2.1, change paragraph 2, sentence 3 to read as follows:

  To add remote access (DAP) support to a task that is built against the RMSRES in supervisor mode you must include the modules:

  ```
  LB:[1,1]RMSLIB/LB:R0EXSY:R0IMPA-LB:[1,1]RMSDAP/LB:R0AULS
  ```

  and include DAPRES as a LIBR or CLSTR option in the task builder command sequence.

  Note

  Unlike RMSRES, the DAPRES cannot be used in supervisor mode.

- In Appendix B, in the discussion of remote file and record access using the DECnet package, the documentation states that the RSTS/E file access listener (FAL) does not support remote record access to indexed files. This is no longer true.

2.17 RSX-11M/M-PLUS RMS-11 Utilities

Please make the following corrections to the *RSX-11M/M-PLUS RMS-11 Utilities* manual:

- In Table 2–1, add the following RMSDES commands:

Command	Format and Function
EXIT_SUPERSEDE	EXIT_S[UPERSEDE] filename[.typ] Names the description file in which the file design is stored.
SAVE_SUPERSEDE	SAVE_S[UPERSEDE] filename[.typ] Names the description file in which the file design is saved.

- In Section 2.2.5, add the new command EXIT_SUPERSEDE, as follows:

  The EXIT_SUPERSEDE command stores the file design in the description file specified in the command string, superseding any existing file by the same name. EXIT_SUPERSEDE then terminates RMSDES and returns the system prompt.

  The format for the EXIT_SUPERSEDE command is as follows:

  ```
  EXIT_S[UPERSEDE] filename[.typ]
  ```

  EXIT_SUPERSEDE names the description file in which the file design is stored. The default file type is DES. If you do not want to supersede an existing description file, use the EXIT command.

- In Section 2.2.5, add the following statement to the description of the EXIT command:

  To supersede an existing description file, use the EXIT_SUPERSEDE command.

- In Section 2.2.9, add the new command SAVE_SUPERSEDE, as follows:

  The SAVE_SUPERSEDE command stores the file design in the description file specified in the command string, superseding any existing file by the same name. If you do not define areas when you issue the SAVE command, RMSDES prompts you for the areas.

  The format for the SAVE_SUPERSEDE command is as follows:

  ```
  SAVE_S[UPERSEDE] filename[.typ]
  ```

  SAVE_SUPERSEDE names the description file in which the file design is saved. The default file type is DES. If you do not want to supersede an existing description file, use the SAVE command.

  If you want to design another file, issue a CLEAR ALL command to restore the attribute values in the design buffer to their defaults.

- In Section 2.2.9, add the following statement to the discussion of the SAVE command:

  To supersede an already existing description file, use the SAVE_SUPERSEDE command.

- In Section 2.6.2, add the following corrections:

  - Modify paragraph 3 of the ALLOCATION field discussion to read as follows:

    If you intend to create a single-area indexed file and do not require RSX positioning, RMS–11 uses the allocation from the file section if no area section exists in your design buffer. If you are allowing RMSDES to define areas for an indexed file by default, RMSDES will automatically calculate an allocation value for each area it defines.

  - Modify paragraph 4 of the EXTENSION field discussion to read as follows:

    If you intend to create a single-area indexed file and do not require RSX positioning, RMS–11 uses the extension from the file section if no area section exists in your design buffer. If you are allowing RMSDES to define areas for an indexed file by default, RMSDES will automatically calculate an extension value for each area it defines.

  - Modify paragraph 5 of the BUCKETSIZE field discussion to read as follows:

    If you intend to create a single-area indexed file and do not require RSX positioning, RMS–11 will use the bucket size from the file section if no area section exists in your design buffer. If you are allowing RMSDES to define areas for an indexed file by default, RMSDES will assign a bucket size value for each area it defines. However, if you choose to define areas explicitly and specify a bucket size value for each area, you should accept the default for the file section and set the bucket size value in each area section.

- In Section 2.8, add the following correction to the explanation of the error message "?DES-F-VOR":

  You entered a value in response to an attribute prompt that was not in the legal range of values for that attribute, or the values you entered resulted in a calculation that caused an overflow for RMSDES. If the value was not within the legal range, the error message is followed by a display of the incorrect value.

- In Table 4–1, add the following switch and description to the table of RMSCNV switches:

/ER[:filespec]	Continue processing after encountering an exception record. If a file specification is provided, then write the primary keys of exception records into the specified file. If no file specification is provided, then output the exception records to the terminal. Default: Stop processing and report RMS error code.

- In Section 4.3, add the following information to the description of RMSCNV switches:

  /ER(:filespec)

  Directs RMSCNV to continue processing when it encounters an exception record in the input file that cannot be written to the output file (see Section 3.4). If you specify a file specification, the exception records will be written to that file. If you do not specify a file specification, the primary key of each exception record will be issued to the terminal. RMSCNV also issues exception record codes (see RMSIFL exception codes, Section 3.3.2).

  If you specify an exception file specification, RMSCNV will create the file as an RMS–11 Variable Fixed Control (VFC) sequential file upon encountering the first exception record. RMSCNV will then write the exception record with a 4-byte exception code to the fixed-control area of the record.

  By default, if you do not specify the /ER switch, RMSCNV will stop processing upon encountering the first exception record and will issue an error message indicating the type of exception record.

- In Table 5–1, add the following information to the description of RMSDSP switches:

/BR	Briefly displays attributes.
/SU	Supersedes existing output file.

- In Section 5.2, include the asterisk (*) and percent sign (%) in the description of wildcard characters permitted in the input file specification.
- In Section 5.3, add the following information to the discussion of RMSDSP commands:

  /BR

  Directs RMSDSP to issue basic displays for indexed files (see Section 5.4, Example 5–3) and container files (see Section 5.4, Example 5–6).

  /SU

  Directs RMSDSP to supersede any existing output file with the same name and version number as the output file specification. If this switch is not supplied and the version numbers are the same, RMSDSP will issue the following error message:

  ```
  ?DSP-F_OPNINP, Error opening DDnn:file.dat as output
  -RMS-E-ER$FEX, File already exists
  ```

- In Table 6–1, add the following information to the table of RMSBCK switches:

/NV	Creates a new version of the output file.

- In Section 6.2, in the discussion of RMSBCK command line format, include the asterisk (*) and percent sign (%) in the description of wildcard characters permitted in the input file specification.

- In Section 6.3.2, add the following information to the description of RMSBCK output switches:

  /NV

  Directs RMSBCK to create a new version of the disk output file if a file currently exists with the same version number as the input file. The current file is not deleted. If you do not specify this switch and a file currently exists with the same file name and version number as the input file specification, RMSBCK will issue the following fatal error message:

  ```
  ?BCK-F-CREOUT, Error opening ddnn:file.dat;n as output
  -RMS-E-ER$FEX, File already exists
  ```

- In Table 7–1, add the following information to the table of RMSRST switches:

/NV	Creates a new version of output file.

- In Section 7.2, in the discussion of the RMSRST command line format, include the asterisk (*) and percent sign (%) in the description of wildcard characters permitted in the input file specification.

- In Section 7.3.2, add the following information to the discussion of RMSRST commands:

  /NV

  Directs RMSRST to create the next higher version number if the expanded input file has the same version number as an existing output file. If this switch is not used and the file name and version number are the same, RMSRST will issue the following error message:

  ```
  ?RST-F-CREOUT, Error opening ddnn:file.dat;n as output
  -RMS-E-ER$FEX, File already exists
  ```

Chapter 3

RMS–11 Version 2.0 Software Modifications, Restrictions, and Supplementary Information

This chapter describes RMS–11 software modifications, lists restrictions, and provides supplementary information for RSX–11M–PLUS Version 4.3.

RMS–11 Version 2.0 has not changed version numbers for this release of RSX–11M–PLUS. Information in this chapter has not been incorporated into the RMS–11 manuals.

3.1 RMS–11 Corrections

The following sections describe corrections for RMS–11. These problems have all been corrected.

3.1.1 RMS–11 Local Access

RMS–11 applications, that performed multi-streaming asynchronous record operations resulting in a high rate of competition for the same buckets, encountered the following set of problems:

- Execution of breakpoint trap in RMS modules R0RLSB and R0RSET.
- Looping indefinitely in RMS modules R0RSET and R0RLCH.
- Stalled indefinitely while waiting on the RMS event flag.
- Receiving "Dynamic Memory Exhausted" error (ER$DME) on a $FIND, $GET, $PUT, $UPDATE or $DELETE operation (operations which do not require new use of dynamic memory).
- Receiving "Bucket Header Checkbyte" error (ER$CHK) when in fact the file does not have this problem.

3.1.2 RMS–11 Ease of Use

Two new ODL files have been added to simplify the building of RMS programs that utilize RMS in supervisor mode.

- LB:[1,1]RMSSLX.ODL - for programs performing local access only.
- LB:[1,1]DAPSLX.ODL - for programs performing local and remote access.

Note

The remote access library (DAPRES) cannot be used in supervisor mode.

3.1.3 RMS–11 Utilities Enhancements

A correction was made in RMSIFL to preserve worst case error on input file read errors. While the error was reported, if RMSIFL was able to continue, the exit status was incorrectly reported as success.

3.2 Software Restrictions

This section describes software restrictions that apply to RMS–11 Version 2.0.

3.2.1 RMS–11 Access Methods

RMS–11 tasks built prior to Version 3.0 of RSX–11M–PLUS will return the error message "Directory not found" (ER$DNF) on certain file operations that are executed from an account set to nonamed directory mode. This will occur if one or both of the following conditions are true:

- The tasks were not built against the RMS–11 resident library.
- A directory is not provided in either the file specification or the default file specification.

You can resolve this problem by first providing a directory in the file specification or by setting your terminal to named directory mode. If that is not possible, you should rebuild those tasks by using the new version of RMS–11.

3.2.2 RMS–11 Remote Access Methods

RMSDAP does not interpret the keysize field value correctly if it is zero. This causes keyed access to a remote indexed file to fail. The application passes the value zero in the keysize field for access to a non-string key and the following error message is issued:

```
ER$RNF (Record not found)
```

If your program is written in F77 or other high-level languages, you may have to use a "USER OPEN" in order to be able to set this value.

3.2.3 RMS–11 Utilities

This section describes restrictions to the RMS–11 utilities for Version 2.0.

RMSCNV Restriction

RMSCNV ignores user-provided area extension quantities when loading a file. The values that it uses are large enough to reduce the number of file extensions in most cases.

RMSDEF Restriction

The DCL command DEFINE defines logical names; therefore, when you invoke RMSDEF, you need to either invoke MCR before typing DEF (to use the MCR mode of DEF instead of the default DCL mode) or insert the following in the LB:[1,2]STARTUP.CMD file:

```
INSTALL LB:[3,54]RMSDEF/TASK=...DFN (or a task name of your choice)
```

3.3 RMS–11 Supplementary Information

The following sections describe information that has been documented in the previous release notes, but is still applicable.

3.3.1 Enhancements to RMSDES Utility

The following enhancements have been made to the RMSDES utility:

- There are two new RMSDES commands: SAVE_S[UPERSEDE] and EXIT_S[UPERSEDE].
- RMSDES issues the warning message "%DES-W-CBK, Continuation buckets will be allocated for this key" when continuation buckets are required. (Continuation buckets may result in decreased performance; see the *RSX–11M/M–PLUS RMS–11 User's Guide*.)
- The informational message "?DES-F-VOR, Value out of legal range" is issued if overflow occurs during calculations of indexed file area allocations. Also, the "number of duplicates:" prompt has been expanded to "number of duplicates per record on this key:" for clarity.
- The error message "%DES-F-NHF, Help file is not available. Check release notes for the location of RMSDES.IDX on your installation media" is issued if the help file LB:[1,2]RMSDES.IDX is not found.

  This is a feature for small systems that have limited disk space.

3.3.2 RMS–11 Installation

The following sections describe information for installing RMS–11 on an RSX–11M–PLUS Version 4.2 or higher operating system.

On RSX–11M–PLUS operating systems, all RMS–11 files are automatically on your system.

3.3.2.1 Location of RMS–11 Files on the Distribution Kit

Table 3–1 describes the location of RMS–11 Version 2.0 files on the RSX–11M–PLUS Version 4.3 distribution kit.

Note

Some file names are marked with an asterisk (*). This indicates that the files are not included on RL02 distribution kits. They are excluded because of space limitations on the kit, and because they are used only to rebuild components of RMS–11 Version 2.0.

Table 3–1: RMS–11 Files on RSX–11M–PLUS Distribution Kit

File Name	Destination	Comments
RMSMAC.MLB	LB:[1,1]	Can be deleted if you are not using MACRO–11 RMS–11 programs
RMSLIB.OLB	LB:[1,1]	Object library for RMS–11 local access
RMSDAP.OLB	LB:[1,1]	Object library for RMS–11 remote access (can be deleted if you do not need remote access)
RMSFUN.OBJ	LB:[1,1]	RMS function routines for the system library
RMSRESAB.MAP	LB:[1,34]	Map for library root
RMSRESAB.TSK	LB:[1,1]	Installed as the root of the resident library
RMSFAKAB.CMD RMSROTAB.STB	LB:[1,24]	For rebuilding RMSLBL and RMSLBM
RMSRESAB.CMD MSRESAB.ODL	LB:[1,24]	For rebuilding RMSRESAB
RMSBCK.TSK	LB:[3,54]	RMSBCK utility; uses RMSRES
RMSRST.TSK	LB:[3,54]	RMSRST utility; uses RMSRES
RMSDEF.TSK	LB:[3,54]	RMSDEF utility; uses RMSRES
RMSDSP.TSK	LB:[3,54]	RMSDSP utility; uses RMSRES
RMSCNV.TSK	LB:[3,54]	RMSCNV utility; uses RMSRES
RMSDES.TSK	LB:[3,54]	RMSDES utility; uses RMSRES
RMSIFL.TSK	LB:[3,54]	RMSIFL utility; uses RMSRES
RMSDES.IDX	LB:[1,2]	Indexed help file used by RMSDES
RMS11.ODL	LB:[1,1]	Prototype ODL file
R0RMS1.MAC	LB:[1,1]	For use with the prototype ODL
RMS11S.ODL	LB:[1,1]	ODL file for sequential access
RMS12S.ODL	LB:[1,1]	ODL file for sequential access
RMS11X.ODL	LB:[1,1]	Standard indexed file ODL
RMS12X.ODL	LB:[1,1]	Indexed file ODL
RMSRLX.ODL	LB:[1,1]	ODL for use with RMSRES
RMSSLX.ODL	LB:[1,1]	ODL for use with RMSRES in supervisor mode
DAP11X.ODL	LB:[1,1]	ODL for use with overlaid RMSDAP
DAPRLX.ODL	LB:[1,1]	ODL for use with DAPRES

(continued on next page)

Table 3–1 (Cont.): RMS–11 Files on RSX–11M–PLUS Distribution Kit

File Name	Destination	Comments
DAPSLX.ODL	LB:[1,1]	ODL for use with RMSRES in supervisor mode and DAPRES (not in supervisor mode)
RMSRES.TSK	LB:[1,1]	Task image for linking in nonsupervisor mode; *not* to be installed as the root of the library
RMSLBL.TSK	LB:[1,1]	Task image for RMSRES segment
RMSLBM.TSK	LB:[1,1]	Task image for RMSRES segment
RMSRES.STB	LB:[1,1]	STB file for RMSRES
DAPRES.TSK	LB:[1,1]	Task image for RMSDAP resident library
DAPRES.STB	LB:[1,1]	STB file for DAPRES
RMSLBL.MAP* RMSLBM.MAP* DAPRES.MAP*	LB:[1,34]	Maps for segmented library
RMSRES.TSK	LB:[3,54]	Task image for linking in supervisor mode; *not* to be installed as the root of the library
RMSRES.STB	LB:[3,54]	STB file for linking supervisor-mode tasks
RMSLBL.CMD* RMSLBL.ODL*	LB:[1,24]	For rebuilding RMSLBL
RMSLBM.CMD* RMSLBM.ODL*	LB:[1,24]	For rebuilding RMSLBM
DAPRES.CMD* DAPRES.ODL*	LB:[1,24]	For rebuilding DAPRES
BCKNON.CMD* BCKNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSBCK
BCKNRN.CMD* BCKNRN.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSBCK
BCKNSN.CMD* BCKNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSBCK
CNVNON.CMD* CNVNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSCNV
CNVNRN.CMD* CNVNRN.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSCNV
CNVNOO.CMD* CNVNOO.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSCNV with RMSDAP

(continued on next page)

Table 3-1 (Cont.): RMS-11 Files on RSX-11M-PLUS Distribution Kit

File Name	Destination	Comments
CNVNRR.CMD* CNVNRR.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSCNV with RMSDAP
CNVNSN.CMD* CNVNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSCNV
DEFNON.CMD* DEFNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSDEF
DEFNRN.CMD* DEFNRN.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSDEF
DEFNSN.CMD* DEFNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSDEF
DESNON.CMD* DESNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSDES
DESNRN.CMD* DESNRN.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSDES
DESNSN.CMD* DESNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSDES
DSPNON.CMD* DSPNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSDSP
DSPNON.CMD* DSPNON.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSDSP
DSPNSN.CMD* DSPNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSDSP
IFLNON.CMD* IFLNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSIFL
IFLNRN.CMD* IFLNRN.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSIFL
IFLNSN.CMD* IFLNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSIFL
RSTNON.CMD* RSTNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSRST
RSTNRN.CMD* RSTNRN.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSRST
RSTNSN.CMD* RSTNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSRST

(continued on next page)

Table 3–1 (Cont.): RMS–11 Files on RSX–11M–PLUS Distribution Kit

File Name	Destination	Comments
RMSUTL.OLB* RMSODL.ODL*	LB:[1,24]	For rebuilding the utilities
GSA.MAC	LB:[USER]	Demonstration program included as an illustration of how to extend an RMS–11 task in the event of pool exhaustion
PARSE.MAC SEARCH.MAC RENAME.MAC ERASE.MAC PARSE.TSK SEARCH.TSK RENAME.TSK ERASE.TSK	LB:[USER]	Demonstration programs for the directory and wildcarding facilities

Note

All RMSDAP files can be deleted if you are not using RMS–11 to access files on remote nodes.

3.3.2.2 Startup Command Procedures

To install RMS–11, install the resident libraries and RMS–11 utilities at system startup. To aid you in the installation process, the file LB:[1,2]STARTUP.CMD contains commands and sample comments that can be edited to become system startup commands. Please note the following items:

- On RL02 pregenerated systems, the RMS–11 segmented library (RMSRES, RMSLBL, RMSLBM) and all the RMS–11 utilities are already installed in the system image. For these kits, the only installation needed is the optional installation of the DAPRES resident library if the system has DECnet support and if RMS–11 remote access facilities are used.
- On systems other than RL02, the startup file contains commands that install the segmented resident library in the system image. Note that these are commands, not comments; if you do not want to install the library, you should edit the file to change these commands to comments. Please note the following items:
  - All resident libraries should be installed using the option /RON=YES.
  - The root of the library, RMSRESAB, must be the task image contained in LB:[1,1]]. The [1,1]RMSRES.TSK should never be installed in the system; it is only used when tasks are linked to use the non-supervisor-mode version of RMSRES.
  - Failure to install the root segment RMSRES will cause the error "INS—Common block not Loaded RMSRES" when a referencing task or utility is invoked. Failure to install any of the remaining resident library segments will not give an error at invocation. However, it may cause the error codes ER$LIB or ER$ENV, or a BPT trap (with R0 containing ER$LIB), when the missing segment is needed by RMS–11.

  - There are several circumstances involving "inconsistent or incomplete resident libraries" that can cause a BPT trap to be generated with R0 containing the error code ER$LIB. This can occur if not all the segments of the library are installed or if the version numbers of one or more segments do not match the root segment, the RMSDAP code, or the task itself. In particular, this can happen to the RMS–11 utilities if they are built to use the segmented resident library and the segments are installed incorrectly.
  - The resident libraries have been built with the PAR=parname option in the TKB command file, where parname is the name of the resident library. This feature was included for compatibility with RSX–11M systems. You should install the resident libraries in the GEN partition. Do not generate individual partitions for the libraries on the RSX–11M–PLUS operating system; doing so would negate much of the benefit of having a "demand-paged" segmented library.
- If you are using RMSRES in supervisor mode, you do not need to install a different task. You simply need to link your task.
- If you want to use the RMS–11 utilities or the RMSDAP resident library (RMSDAP), you must edit the startup command file, which contains template commands (comments) that you can edit to install these facilities.

3.3.2.3 Utility Configurations

The RMS–11 utilities that are provided on the distribution kit are built to use the segmented resident library. Consequently, RMSRES, RMSLBL, and RMSLBM should be installed before you use any of the RMS–11 utilities.

3.3.2.4 Utility Command and ODL Files

For each utility, a command file and one or more ODL files are provided to build the utility using disk-overlaid RMS–11. The names of the files are as follows:

Utility	Files
RMSBCK	BCKNON.CMD, BCKNON.ODL, and RMSODL.ODL
RMSRST	RSTNON.CMD, RSTNON.ODL, and RMSODL.ODL
RMSCNV	CNVNON.CMD and CNVNON.ODL
RMSDSP	DSPNON.CMD, DSPNON.ODL, and RMSODL.ODL
RMSDESAB	DESNON.CMD, DESNON.ODL, and RMSDESAB.ODL
RMSDEF	DEFNON.CMD, DEFNON.ODL, and RMSODL.ODL
RMSIFL	IFLNON.CMD and IFLNON.ODL

In addition, a command file and ODL file are provided to build each utility with the resident library RMSRES. The names of the files are as follows:

Utility	Files
RMSBCK	BCKNRN.CMD and BCKNRN.ODL
RMSRST	RSTNRN.CMD and RSTNRN.ODL
RMSCNV	CNVNRN.CMD and CNVNRN.ODL
RMSDSP	DSPNON.CMD and DSPNON.ODL
RMSDESAB	DESNRN.CMD and DESNRN.ODL
RMSDEF	DEFNRN.CMD and DEFNRN.ODL
RMSIFL	IFLNRN.CMD and IFLNRN.ODL

Finally, a command file and ODL file are provided to build each utility using the supervisor-mode library RMSRES. The names of the files are as follows:

Utility	Files
RMSBCK	BCKNSN.CMD and BCKNSN.ODL
RMSRST	RSTNSN.CMD and RSTNSN.ODL
RMSCNV	CNVNSN.CMD and CNVNSN.ODL
RMSDSP	DSPNSN.CMD and DSPNSN.ODL
RMSDES	DESNSN.CMD and DESNSN.ODL
RMSDEF	DEFNSN.CMD and DEFNSN.ODL
RMSIFL	IFLNSN.CMD and IFLNSN.ODL

For RMSCNV, two additional configurations are available if you want to use RMSCNV to access files on remote nodes. To build RMSCNV using the clustered RMS–11 and RMSDAP resident libraries, use the files CNVNRR.CMD and CNVNRR.ODL. To build RMSCNV using disk-overlaid RMS–11 and RMSDAP, use the files CNVNOO.CMD and CNVNOO.ODL.

Note

These command files and ODL files are not supplied with the RL02 kits.

3.3.2.5 Rebuilding the Utilities

To rebuild the utilities, perform the following steps:

1. Log in to a privileged account.
2. Set your default account to directory [1,24] on the system disk.
3. Use the Task Builder (TKB) to build the utility or utilities.

The utilities and the corresponding map files will be built in the current account. You may then want to put the utility in the system account ([3,54]).

Each command file for a particular utility creates a utility of the correct name. For example, CNVNRN.CMD and CNVNON.CMD both produce task images called RMSCNV.TSK. One is built to use the resident library and one is built with disk-overlaid RMS–11.

Chapter 4

Layered Product Support

4.1 Correction Files

Table 4–1 lists the layered products and correction files for the RSX–11M–PLUS Version 4.3 operating system. The layered products that have updated correction files for this release are marked with an asterisk (*). The application of the updated correction files to each layered product will result in improved software performance. Table 4–1 also includes the corresponding directories and documentation files for the layered products. The documentation files describe the procedure for applying corrections to the specific layered product.

Table 4–1: Layered Product Correction File References

Layered Product	Directories Used	Backup Set	Documentation File Name
PDP–11 BASIC–PLUS–2 Version 2.5	[211,200]	BP2	[211,200]BP2.DOC
COBOL–81/RSX * Version 3.0	[215,200]	COBOL81	[215,200]C81.DOC
DATATRIEVE–11/RSX Version 3.2	[222,200]	DTR	[222,200]DTR.DOC
DCX KMV11–A Development Tools * Version 1.0	[224,200]	DCX	[224,200]DCX.DOC
DECnet–11M–PLUS * Version 4.0	[230,200] [225,20] [225,40] [230,20] [230,40]	DECNET	[230,200]DECNET.DOC

(continued on next page)

Table 4-1 (Cont.): Layered Product Correction File References

Layered Product	Directories Used	Backup Set	Documentation File Name
RSX-11M-PLUS DIBOL * Version 1.3	[371,200]	DIBOL	[371,200]DIBOL.DOC
DPM-PLUS Version 1.1	[236,200]	DPM	[236,200]DPMP.DOC
IP11 Industrial I/O Subsystem Software Driver Version 4.1	[340,200]	IP11	[340,200]IPUPINST.DOC
KMV11-A HDLC Framing Software * Version 1.0	[253,200]	KMV11	[253,200]KMV11.DOC
KMV11 X25 Link Level * Version 1.0	[322,200]	KXI	[322,200]KXI.DOC
MIRA * Version 1.0	[370,200]	MIRA	[253,200]MIRA.DOC
PDP-11 Pascal/RSX Version 1.2	[262,200]	PASCAL	[262,200]PASCAL.DOC
PDP-11 SORT/MERGE Version 3.0	[311,200]	SORT	[311,200]SORT.DOC
PDP-11 Symbolic Debugger/RSX Version 2.1	[247,200]	PDPDBG	[247,200]PDPDBG.DOC
RSX-11 2780/3780 Protocol Emulator Version 4.1	[301,200] [301,25]	2780	[301,200]2780.DOC
RSX-11 3271 Protocol Emulator* Version 3.1	[345,200] [345,25]	3271	[345,200]3271.doc

4.2 Installing Layered Products Restriction

Because of support for named directories, the MCR command SET /NONAMED or the DCL command SET DEFAULT/NONAMED should be used before installing layered products.

Appendix A

The Digital Equipment Computer Users Society (DECUS)

DECUS, the Digital Equipment Computer Users Society, is one of the largest and most active user groups in the computer industry. It is a nonprofit association, supported and administered by Digital, but actively controlled by its members. DECUS headquarters, located in Marlborough, Massachusetts, administers all international policies and activities.

Members include individuals who have purchased, leased, ordered, or used a Digital computer, or anyone who has a genuine interest in DECUS. Membership is free and voluntary.

A.1 DECUS Goals

The primary goals of DECUS are as follows:

- To advance the art of computation through mutual education and exchange of ideas and information
- To establish standards and to provide channels to facilitate the exchange of computer programs
- To provide feedback to Digital regarding hardware and software customer needs
- To advance the effective use of Digital computers, peripherals, and software by promoting the interchange of information

To further these goals, DECUS serves its members by holding symposia; by maintaining a program library; by publishing an association newsletter, technical newsletters, and books; and by supporting a number of Special Interest Groups (SIGs).

A.2 DECUS Activities

Local, regional, and national DECUS organizations give members the opportunity to meet other Digital customers and employees in an informal setting. At local monthly meetings and national symposia, members can discuss their ideas, learn what others are doing, and give Digital valuable feedback for future product development.

Often the national meetings in various countries also provide the stage for major new product announcements as well as a showplace for interesting developments in both hardware and software technology. At any meeting, members might describe ideas and programs they have implemented or the fine-tuning of a particular feature or application. Members present papers, participate in panel discussions, lead workshops and Pre-Symposium Seminars, and conduct demonstrations for the benefit of other members.

United States DECUS Symposia are held twice per year, in the Spring and Fall, in varying locations around the country. Events which are currently scheduled include the following:

Spring DECUS	May 7–11, 1990	New Orleans, LA
Fall DECUS	December 10–14, 1990	Las Vegas, NV

Many of the technical papers and presentations from each symposium are published as a book, the *DECUS Proceedings*. Copies of the *DECUS Proceedings* are supplied to symposia attendees and can be purchased by DECUS members. DECUS also publishes newsletters focusing on special interests and a society newsletter.

DECUS has a Program Library, which contains over 1700 software packages that are written and submitted by users. A wide range of software is offered, including languages, editors, numerical functions, utilities, display routines, games, and other types of application software. Also available are library catalogs, which contain program descriptions and ordering information. The programs are available for a nominal service charge that covers the cost of reproduction and media.

Many DECUS members derive additional benefits from joining a DECUS Special Interest Group (SIG). SIGs often hold smaller meetings while attending regional and national meetings. They may also meet informally to discuss their special fields, such as operating systems, languages, processors, and applications. There are at least 25 SIGs, including the RSX SIG, in the United States alone. Many of the SIGs print newsletters and disseminate valuable technical information to members.

A.3 DECUS Membership

To obtain a membership form for DECUS, contact a Digital sales representative or the nearest DECUS chapter office. Membership information may be obtained directly from U.S. DECUS Headquarters by calling telephone number 508–480–3446.

Appendix B

Reporting Problems

This appendix describes the procedure for submitting a Software Performance Report (SPR). An SPR allows you to report any problems with or questions about your system directly to Digital.

An SPR can be used for:

- Software errors
- Documentation errors (when the Reader's Comments form is not appropriate)
- Follow-up on a previous SPR
- Questions
- Suggestions

An SPR cannot be used for:

- Software license and price policies
- Obvious hardware problems
- Logistical or clerical problems with kits, such as blank media, or failure to receive the *Software Dispatch*
- Problems with user-written software

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

- Describe only one problem per form. This will facilitate a more rapid response because it allows the person answering the problem to concentrate more fully on that particular problem. One problem per form also helps simplify record-keeping.
- Define as accurately as possible the state of the system and circumstances when the problem occurred.
- Illustrate the problem with specific examples.

- If you report a documentation error, specify the title of the manual, and include the section and page number where the error occurred. Include a table or figure number if appropriate.

SPRs are assigned a priority of 1 to 5. An SPR that is assigned a priority of 1 receives the highest priority. Priorities are described as follows:

1. Most production work cannot be run.
   - Major system functions are unusable.
   - You cannot boot the system.
   - Necessary peripherals cannot be used.
2. Some production work cannot be run.
   - Certain functions are unusable.
   - System performance has declined.
   - Installation does not have excess capacity.
3. All production work can be run with some user impact.
   - Significant manual intervention is required.
   - System performance has declined.
   - Installation has excess capacity.
4. All production work can be run with no significant impact on user.
   - Problem can be patched or easily bypassed.
5. No system modifications are needed to return to normal production.
   - Suggestions are supplied.
   - Errors in documentation are noted.

You can submit the following categories of SPRs:

- Problem/Error SPR

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

  This type of SPR contains a suggestion. It is assigned a priority of 5. You do not receive an answer to this report.
- Other

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

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

- CRASH–Supply a copy of the Executive task-build map, output from the console terminal, the SYSGEN saved-answer file, the Executive STB file, and the crash dump. If the crash is reproducible, accurately describe the details and supply a hard copy or user source code when necessary.
- DRIVERS–Supply controller/device information, software options, error log output, a copy of device registers, and a sample program.
- UTILITIES–Supply a copy of your terminal output, showing setup commands, before and after effects, and relevant file information.
- TASK BUILDER–Supply a copy of your terminal output command files, the task map, and a dump of the first few blocks of the task image.
- FILE SYSTEM–For a corrupted volume, supply output from the verification utility and dump of the volume. For improper results, supply the error code, a file header dump, and a sample program.
- ERROR LOG REPORT GENERATOR (RPT)–Supply a copy of the report file generated by RPT, either on a hardcopy listing or on machine-readable media.

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

The SPR process is sometimes lengthy. Therefore, if you have a critical problem, contact your local Digital office. In the meantime, read the *Software Dispatch* for news on the RSX–11M–PLUS operating system, which includes changes and problems other people have found and solved.

Appendix C

Applying Corrections to Source Files

Interim changes to the Executive, MCR, and device drivers are made by creating correction files that are processed by the Source Language Input Program (SLP). SLP generates a new copy of the modules that contain the errors by applying the corrections to the source file on the distribution kit. (See the *RSX-11M-PLUS Utilities Manual* for complete information on SLP.)

After you have applied the corrections and have obtained a new version of the file, *do not* delete the original source file. Interim changes that may be distributed later are cumulative and depend on the availability of the original sources.

C.1 Updating an Executive Source Module

To update an Executive source file (ABCDEF.MAC, for example), mount the disk on which you performed your system generation and create a SLP correction file named ABCDEF.COR in the directory [11,40]. (All the following instructions assume that you are working on the disk on which you performed your system generation.) Then, while your system is running under User Identification Code (UIC) [11,10], submit the correction file to SLP. For example, you could follow this sequence to create REQSB.MAC:

```
>
>
[Creating new file]
Input




*
[Exit]

>
>
```

If the updated Executive module in your system is not a loadable driver, use the following procedure:

1. Assemble the new module, using the RSXMC.MAC file for the target system. For example, type the following commands:

   ```
   >
   >
   ```

2. Use the Librarian Utility Program (LBR) to replace the old version of the module in the RSX11M.OLB file on the target system.

   For example:

   ```
   >
   >
   ```

3. Perform the following sections of SYSGEN:
   - Building the Executive and Drivers
   - Building the Privileged Tasks
   - Creating the System Image File

If the modified file in your system is a loadable device driver (ZZDRV.MAC, for example), the updated module can be replaced without rebuilding the Executive. Assemble the updated module and replace the resulting object file in the RSX11M.OLB file of your target system. For example, assume that [11,10] contains the RSXMC.MAC file resulting from your system generation and do the following:

```
>
>
>
>
```

Use [200,200]ZZDRVBLD.CMD to rebuild the driver. If necessary, copy ZZDRV.TSK and ZZDRV.STB into the directory that corresponds to the system UIC, or the library UIC if the driver is vectored.

For example:

```
>
>
>
SYSUIC=[g,m]
>
>
```

Use the Virtual Monitor Console Routine (VMR) to unload the old device driver and to load the new one. If the new driver is larger than the old one, it may not fit into the same locations as the old one. It may be necessary to unload and reload all of the loadable drivers in that partition to create enough room.

For example:

```
>
Enter filename:
VMR>
VMR>
VMR>
>
```

If the driver is vectored, you must use the /VEC switch with the UNL and LOA commands. In the preceding example, you would substitute the following VMR commands:

```
VMR>
VMR>
VMR>
```

Hardware boot the modified system.

C.2 Updating an MCR Source Module

To update the MCR source file SETOV.MAC, use the following procedure:

1. Create the SLP correction file [12,40]SETOV.COR and use it to update [12,10]SETOV.MAC. Assemble SETOV for the target system, as follows:

   ```
   >
   >
   [Creating new file]






   *
   [Exit]
   ```

   ```
   >
   >
   ```

2. Assume that directory [11,10] contains the RSXMC.MAC file resulting from your target system generation and do the following:

   ```
   >
   >
   ```

   All of the Task Builder command files output a map to the logical device MP, which must be assigned to NL or another device to avoid a diagnostic error message from the Task Builder.

3. If it was necessary to modify the MCR Task Builder command file (MCRBLD.CMD) during the last system generation, it may now be necessary to repeat those changes. To rebuild the secondary portion of MCR (...MCR) and replace the module SETOV, use the following procedure:

```
>
>
>
>
>
>
>
Enter filename:
VMR>
VMR>
VMR>
```

To rebuild the MCR dispatcher (MCR...) and to replace the module MCRDIS, use the following procedure:

```
>
>
>
>
>
>
>
Enter filename:
VMR>
VMR>
VMR>
>
```

There is only one procedure for replacing an external MCR task. It involves the following steps:

1. Create the SLP file, apply it, and create the object file.
2. Incorporate the updated module into the task's object library.
3. Rebuild the task and install it in the system using MCR or VMR. Before using VMR, you must assign SY and LB to the disk containing the target system.

The following example replaces the module INSLB of the external MCR task INSTALL:

```
>
>
```

4. If it was necessary to modify the external task's Task Builder command file during the last system generation, it may be necessary at this time to repeat those changes:

   ```
   >
   >
   >
   >
   Enter filename:
   VMR>
   VMR>
   VMR>
   >
   ```

   Hardware boot the system.

C.3 Updating a DCL Source Module

The procedure is different for the DCL task. There are two DCL object libraries: DCLR.OLB for modules in the root segment of the DCL task, and DCLO.OLB for modules in DCL's overlay segments. (One module, COMMAND, has versions in both the root and overlay libraries.) These libraries are located in directory [1,24] on your distribution kit. If you are not sure whether a module belongs in the root or the overlay library, use the Librarian Utility Program (LBR) to scan the module names in the libraries. For instance, to view the module names in DCLO.OLB, enter the MCR or DCL command shown next.

```
>

$
```

The modules you are most likely to need to modify are the DCL syntax tables, which are in DCLO.OLB. (All DCL syntax tables are in overlay segments.) See the *RSX–11M–PLUS and Micro/RSX System Management Guide* for more information on the structure of the DCL task and on the DCL task-building process.

File DCL.CMD, located in directory [23,24] on your distribution kit, can be used either to assemble DCL overlay modules and to rebuild DCL, or as a template for your own commands. The file contains its own instructions. Note that to use this command file directly, you must copy various files into directory [23,24] and appropriately modify any directories that they reference. File DCL.CMD also shows how to assemble the special module COMMAND.

The following process shows how to reassemble a DCL overlay module and rebuild DCL. The process is similar for root modules; simply specify DCLR rather than DCLO. See the DCL.CMD command procedure for details on rebuilding COMMAND.

If your command line interpreter (CLI) is MCR, use the following commands:

```
>
>
>
>
>
```

If your CLI is DCL, use the following commands:

```
$
$
$
$
$
```

When you have built a new version of DCL.TSK, you must replace the old copy of DCL as a system CLI. First, any terminals whose CLI is DCL must be logged out or set to another CLI. Then, the following commands must be executed (note that these commands are for MCR only):

```
>
>
>
>
```

The REMOVE and INSTALL commands are also valid VMR commands and can be used to modify your system's image on disk. However, the CLI /INIT command is for MCR only, and it must be put in your system's startup file (if it is not there already).

C.4 Applying Corrections to the File Systems (F11ACP)

All F11ACP updates begin by creating a SLP correction file in directory [13,40]. The following example patches a module called WTRN1.

1. Boot your system and log in to a privileged account, as follows:

```
>
Password:
```

2. If necessary, restore the required files from the distribution tape (the following example assumes that the files in [13,10] have been deleted, but the files in [1,24] have not):

```
>
>
>
BRU>
From:
To:
BRU--Completed
BRU>
```

3. Create the correction file, as follows:

```
>
>
```

4. Apply the correction, as follows:

```
>
>
```

5. Assemble the corrected module with the Executive macro library, the Executive prefix file RSXMC.MAC, and the prefix file F11PRE.MAC, as follows:

```
>
>
```

6. Replace the defective module in the file control processor (FCP) library, as follows:

```
>
>
Modules replaced
WTRN1
```

All of the task-build command files require that the logical device MP be assigned to the appropriate device.

In the following example, xxx must be replaced by the 3-character designation for your desired FCP, that is, MDL or LRG.

1. Task build the new FCP by using the updated library as follows:

```
>

>
```

2. Install the updated FCP in the system image as follows:

```
>
Enter filename:
VMR>
VMR>
VMR>
>
```

3. Reboot the system to place the new FCP in use.

C.5 Applying Corrections to the Reconfiguration Tasks CON and HRC

The following example illustrates how to patch module CNCMR for the CON task. Unless indicated otherwise, use a similar command sequence for correcting module HRONL for the HRC task.

This example assumes that:

- You have deleted the source files in [27,10].
- You have not deleted the object libraries and command files in [1,24] and [1,20].
- The object library for both CON and HRC is [1,24]OLR.OLB.
- The disk to which you will apply the patches is mounted Files-11 on drive DB0.
- The distribution tape is mounted foreign on MM0.

Perform the following steps to patch CNCMR:

1. Boot your system, and log in to a privileged account.
2. If necessary, restore the required files from the distribution tape as follows:

```
>
>
>

>
BRU>
From:
To:
BRU -- Completed
BRU>
```

   If you were correcting module HRONL, you would substitute the following for the From: line in the previous command sequence:

```
From: MM0:
```

3. Create the SLP correction file in UIC [27,40], as follows:

```
>
>
```

4. Apply the SLP correction file to CNCMR as follows:

```
>
>
```

5. Assemble the corrected CNCMR module by using the Executive macro library and the Executive prefix file RSXMC.MAC; you also use these when assembling a corrected HRC module. In addition, use either the prefix file CNPRE.MAC, if assembling a CON module, or HRPRE.MAC, if assembling an HRC module.

   For example:

   ```
   >
   >
   ```

6. Replace the CNCMR object module in the OLR library as follows:

   ```
   >
   >
   ```

7. Task build CON by using the updated library as follows:

   ```
   >
   >
   >

   >
   ```

   Note

   If you do not have the task-build command files [1,24]CONBLD.CMD and [1,24]CONBLD.ODL, or [1,24]HRCBLD.CMD and [1,24]HRCBLD.ODL, you must use the section of SYSGEN that rebuilds system-supplied tasks.

8. Install the updated reconfiguration utility in the system image as follows:

   For CON, type: For HRC, type:

   ```
   >                       >
   Enter filename:          Enter filename:
   VMR>                    VMR>
   VMR>                    VMR>
   VMR>                     VMR>
   ```

9. In order to use the new reconfiguration tasks, follow these steps:

   a. Shut the system down using the SHUTUP utility as follows:

      ```
      >
      ```

   b. Reboot the system.

C.6 Applying FCS Corrections

Correcting the File Control Services (FCS) modules on an RSX–11M–PLUS system can be done by updating the source files, by assembling them, and by replacing modules in the system library, usually LB:[1,1]SYSLIB.OLB.

This process is complicated by the fact that there are three kinds of FCS, as follows:

- ANSI–Supports ANSI-format magnetic tape and big buffers.
- Non-ANSI–Does not support ANSI tape or big buffers.
- Multibuffered–Supports ANSI tape, big buffers, and multiple. buffers

An FCS source file like CLOSE.MAC contains conditional assembly directives that can produce three different CLOSE objects, depending on the global symbols defined when CLOSE.MAC is assembled. These three different CLOSE objects correspond to the three kinds of FCS. Other FCS source files, like DELETE.MAC, have no such conditional assembly directives. They are only assembled one way; that is, only one DELETE object exists.

The SYSLIB.OLB file provided on the kits contains the ANSI FCS. Thus, this system library contains modules like CLOSE, assembled with the ANSI tape conditionals and big buffer conditionals enabled, and modules like DELETE, which have no such conditionals and are the same in any FCS.

An alternate system library called NOANSLIB.OLB is also provided. It contains an FCS that does not support ANSI tape, big buffers, or multiple buffers. Be sure that you know whether the system library on your system contains the ANSI FCS, or if it has been replaced with the non-ANSI or multibuffered FCS.

Details of the correcting procedure follow. MCR syntax is used throughout.

C.7 Updating the FCS Sources

The FCS source files are found in [50,10] on the kits. Updating a source is done by entering the correction file into [50,10], and by entering “SLP @filename” to apply the correction.

C.8 Assembling FCS

Assembling the updated source or sources can be done in either of two ways. One way is simple and time-consuming; the other is quick, but it must be done with great care. The simple way is to set your UIC to [50,24] and enter MAC @FCSASM. This will assemble every FCS variant properly, producing over 100 object files.

The other choice is to assemble only the sources that have been updated. This is not as simple, but it saves machine time. To assemble only a particular file, look at the five *.CMD files in [50,24] on the kit. Search the command files for all references to the file you want to assemble. A description follows of each command file and what you do with its contents.

1. FCSBOTH.CMD assembles files like DELETE.MAC, which contain no code specifically written to support or deny support to ANSI tape, big buffers, or multiple buffers. If the source file you have updated is mentioned in FCSBOTH.CMD, then it is assembled the same way regardless of which kind of FCS is in your system library. For example, the following lines assemble DELETE:

   ```
   [50,24]DELETE,[50,34]DELETE/-SP=-
   [50,10]FCSPRE,DELETE
   ```

   (FCSPRE.MAC defines necessary macros and global symbols.)

   To assemble DELETE.MAC, you should set your UIC to [50,24] and enter the following command line:

   ```
   >
   ```

As an alternative, you could put the command line in a file and enter "MAC @filename". A file like DELETE.MAC, which has no conditional assembly directives, is mentioned only in FCSBOTH.CMD.

2. FCSANSI.CMD assembles source files that contain ANSI tape, big buffer, or multibuffering conditionals. It produces objects for the ANSI kind of FCS. If your system library contains the ANSI FCS, and the source file you have updated is mentioned in FCSANSI.CMD, then you should use the command line you found in FCSANSI.CMD to assemble the source file. For example, the following lines from FCSANSI.CMD assemble CLOSE.MAC:

   ```
   [50,24]CLOSE.MTA,[50,34]CLOSE.MTA/-SP=-
   [50,10]FCSANSI/PA:1,FCSBIGBUF/PA:1,FCSPRE,CLOSE
   ```

   FCSANSI.MAC enables the ANSI tape conditional assembly directives, and FCSBIGBUF.MAC enables the big buffer conditionals. Note the sequence of input file names. It is absolutely imperative that the source file is the last file name specified, and that FCSPRE is next to it, in every FCS assembly. Otherwise, the proper conditionals will not be enabled, with potentially confusing and dangerous results. Any file that is assembled by FCSANSI.CMD is also assembled by FCSNOANSI.CMD and FCSMULBUF.CMD. Note the MTA file types above, which distinguish between the three kinds of CLOSE objects.

3. FCSNOANSI.CMD also assembles source files that contain ANSI tape, big buffer, or multibuffering conditionals. It produces objects for the non-ANSI kind of FCS. If your system library contains the non-ANSI FCS, and the source file you have updated is mentioned in FCSNOANSI.CMD, then you should use the command line you found in FCSNOANSI.CMD to assemble the source file. For example, this command line from FCSNOANSI.CMD assembles CLOSE.MAC:

   ```
   [50,24]CLOSE.NMT,[50,34]CLOSE.NMT/-SP=-
   [50,10]FCSPRE,CLOSE
   ```

4. FCSMULBUF.CMD assembles source files that contain ANSI tape, big buffer, or multibuffering conditionals. It produces objects that support all of these features. For example, this command line from FCSMULBUF.CMD assembles CLOSE.MAC:

   ```
   [50,24]CLOSE.MBF,[50,34]CLOSE.MBF/-SP=-
   [50,10]FCSANSI/PA:1,FCSMULBUF/PA:1,FCSBIGBUF/PA:1,FCSPRE,CLOSE
   ```

   FCSMULBUF.MAC enables the multibuffering conditionals.

5. FCSASM.CMD simply causes the other command files to be executed, to assemble FCS in all ways. If you are assembling single sources, ignore FCSASM.CMD.

More information is necessary to correctly assemble a few FCS sources that require additional prefix files. These sources are GET.MAC, PUT.MAC, OPEN.MAC, FINIT.MAC, and RDWRIT.MAC. GET.MAC can be assembled in the normal way (ANSI, non-ANSI, or multibuffered) to produce GET.MTA, GET.NMT, or GET.MBF. An additional file, GPSEQ.MAC, will enable conditional assembly directives in GET.MAC to produce GETSQ, the sequential GET module, for each kind of FCS. For example, this is the command line from FCSANSI.CMD, which produces GETSQ.MTA:

```
[50,24]GETSQ.MTA,[50,34]GETSQ.MTA/-SP=-
[50,10]GPSEQ/PA:1,FCSANSI/PA:1,FCSBIGBUF/PA:1,FCSPRE,GET
```

If you correct GET.MAC, be sure that you assemble it to produce objects GET and GETSQ for whatever kind of FCS your system library contains.

Similarly, GPSEQ.MAC will produce PUTSQ, the sequential PUT module, for each kind of FCS. This is the command line from FCSNOANSI.CMD, which produces PUTSQ.NMT:

```
[50,24]PUTSQ.NMT,[50,34]PUTSQ.NMT/-SP=-
[50,10]GPSEQ/PA:1,FCSPRE,PUT
```

If you correct PUT.MAC, be sure that you assemble it to produce objects named PUT and PUTSQ.

OPEN.MAC can be assembled in the normal way (ANSI, non-ANSI, or multibuffered) to produce OPEN.MTA, OPEN.NMT, or OPEN.MBF. Three additional prefix files exist: OPFID.MTA, OPFNB.NMT, AND OPENR.MAC.

OPFID.MAC produces the open-by-file-id module, OPFID. This is the command line from FCSANSI.CMD, which produces OPFID.MTA:

```
[50,24]OPFID.MTA,[50,34]OPFID.MTA/-SP=-
[50,10]OPFID/PA:1,FCSANSI/PA:1,FCSBIGBUF/PA:1,FCSPRE,OPEN
```

OPFNB.MAC produces the open-by-file-name block module, OPFNB. This is the command line from FCSANSI.CMD, which produces OPFNB.NMT:

```
[50,24]OPFNB.NMT,[50,34]OPFNB.NMT/-SP=-
[50,10]OPFNB/PA:1,FCSPRE,OPEN
```

OPENR.MAC produces the open module for resident libraries, OPENR. This is the command line from FCSMULBUF.CMD, which produces OPENR.MBF:

```
[50,24]OPENR.MBF,[50,34]OPENR.MBF/-SP=-
[50,10]OPENR/PA:1,FCSANSI/PA:1,FCSMULBUF/PA:1,FCSBIGBUF/PA:1,FCSPRE,OPEN
```

If you correct OPEN.MAC, be sure that you assemble it to produce objects OPEN, OPFID, OPFNB, and OPENR for whatever kind of FCS your system library contains.

FINIT.MAC and RDWRIT.MAC can be assembled to produce FINIT.OBJ and RDWRIT.OBJ. The prefix file FCSSUP.MAC enables conditionals to produce objects FINTSL.SUP and RDWRSL.SUP to be used in supervisor-mode libraries. These objects are necessary for correct execution of the user asynchronous system trap (AST) completion routines, which may be specified for FCS READ$ and WRITE$ functions. This is the line from FCSANSI.CMD that produces FINTSL.SUP:

```
[50,24]FINTSL.SUP,[50,34]FINTSL.SUP/-SP=[50,10]FCSSUP/PA:1,FCSPRE,FINIT
```

Object files FINTSL.SUP and RDWRSL.SUP contain the same entry point names as FINIT.OBJ and RDWRIT.OBJ. Therefore, they must be replaced in the system library with their entry points deleted so that they are only used when they are explicitly called when a supervisor-mode library is task built.

C.9 Replacing the FCS Object Modules

Set your UIC to [1,1], make a backup copy of the libraries, and use LBR to replace the corrected FCS objects that you have assembled. For example, suppose you have updated and assembled DELETE, PUT, and FINIT on a system with an ANSI FCS in the system library. You would then use the following procedure:

```
>
>
>
>
```

If you use NOANSLIB.OLB, also use the following procedure:

```
>
>
>
```

If you want to incorporate the corrected modules into an FCSRES resident library or FCSFSL supervisor-mode library, first rebuild the library, and then rebuild every task that links to it with SYSGEN. If you rebuild and install a resident library and run a task that linked to the old resident library, then the task will call routines in the resident library at the wrong addresses. The results will be uncertain and potentially damaging, especially for privileged tasks.

Index

A

Account Maintenance Program
  See ACNT
ACNT
  blank account fields, 2–35
ALLOCATION field
  RMS–11, 2–47
ALT command
  keywords
    /RPRI, 2–7
    /TERM, 2–7
Alternate record key field
  RMS–11, 2–44
ASSIGN command, 2–7
AST
  user completion routines, C–11
Asynchronous system trap
  See AST

B

Backup and Restore Utility
  See BRU
BACKUP command
  qualifiers
    /CREATED, 2–2
    /ERROR, 2–3
    /MODIFIED, 2–2
  restrictions
    /LENGTH qualifier, 2–3
    /NEW_VERSION qualifier, 2–3
    /REWIND/APPEND qualifiers, 2–3
    /TAPE_LABEL qualifier, 2–3
    /VERIFY qualifier, 2–3
Bad Block Replacement Control Task
  See RCT
BRU
BRU (Cont.)
  compatibility with VAX–11 RSX and VAX CoProcessor/RSX, 1–3
  error messages, 1–6
  file size allocation, 1–10
  /LENGTH qualifier restriction, 2–3
  qualifiers
    /COMPARE, 1–4
    /ERRORS, 2–3
    /VERIFY, 1–4
  restoring data, 1–10
BUCKETSIZE field
  RMS–11, 2–47

C

C compiler support, 1–3
CLI/INIT command, C–6
COBOL–81 correction file, 4–1
Communication driver
  example, 2–32
  programming hints, 2–30
Communication mode
  setting full duplex (communication driver), 2–28
Communication set operation
  IO.SYN function (communication driver), 2–28
/COMPARE qualifier
  BRU, 1–5
Contiguous file areas
  RMS–11, 2–45
Continuation buckets
  RMSDES utility, 3–3
Controller identification
  DSA driver, 2–33
Crash driver
  memory dump, 2–15

Crash dump
  loadable support, 2–15
Create Regions Directive, 1–3

D

Data cache
  readahead support, 1–10
  write loading temporary files, 2–35
DATATRIEVE–11/RSX correction file, 4–1
DCL
  updating source file, C–5
DCX KMV11–A Development Tools, 4–1
Deadlock
  RMS–11, 2–44
DECnet–11M–PLUS correction file, 4–1
DECUS, A–1
DEFINE LOGICALS command, 2–7
Device
  allocation failure, 2–44
  disconnecting (communication driver), 2–29
  initializing (communication driver), 2–29
  large disk restriction, 1–2
  LP:, 1–11
  RA70 disk drive, 1–2
  RA90 disk drive, 1–2
  TK25 cartridge tape, 2–3, 2–16
  TK50 cartridge tape, 2–3
Device driver
  changes, 1–10, C–1
  updating loadable, C–2
DIGITAL Command Language
  See DCL
Digital Equipment Computer Users Society
  See DECUS
Disk
  maximum files on, 1–2
Disk Save and Compress utility, 1–13
DMC11 (communication driver)
  message send, 2–32
  powerfail, 2–31
  synchronous line interface, 2–26
DPM–PLUS correction file, 4–2
DSA driver
  controller identification, 2–33
  error log file, 2–33
  error log support, 2–33
  error packets, 2–33
  I/O sequence number, 2–33
  offset P.SEQ, 2–33
  powerfail recovery
    type, 2–33, 2–34
DSA driver (Cont.)
  Read Serial Number function, 2–33
  time delay, 2–33
DU driver
  Read Serial Number function, 2–32
  setting to volume valid, 2–32
  Suppress Error Logging subfunction, 2–32
DUP11 (communication driver)
  sync character, 2–26
  synchronous line interface, 2–26

E

$EDMSG directive, 2–35
  file name string conversion, 2–35
ER$FUL error, 2–44
ER$KEY error, 2–44
ER$MRS error, 2–44
ER$NOD error, 2–44
Error log file
  DSA driver, 2–33
  report, 1–12
Error log support
  DSA driver, 2–33
  RA70 disk drive, 1–2
  RA90 disk drive, 1–2
Error packets
  DSA driver, 2–33
/ERROR qualifier
  BACKUP command, 2–3
  BRU Utility, 2–3
/ER switch
  RMS–11, 2–47
Executive
  modifying source file, C–1
  support for volume valid, 1–9
  virtual terminals, 1–10
EXIT_SUPERSEDE command
  RMSDES utility, 2–46, 3–3
Extend Task directive
  APR protection mask, 2–20
EXTENSION field
  RMS–11, 2–47
EXTK$ directive, 2–20

F

FCS
  ANSI, C–8
  assembly files, C–9
  code size, 1–13
  modifying modules, C–8
  multibuffered, C–8

FCS (Cont.)
- non-ANSI, C–8
- READ$ function, C–11
- source file
  - conditional assembly, C–9
  - updating, C–9
- VMS ACP function support, 1–13
- WRITE$ function, C–11

File Control Services
- See FCS

File loading
- RMS–11 restriction, 3–2

File specification
- RMS–11, 2–48
  - wildcard, 2–48

FMS
- command file example, 2–37

.FORM directive
- terminal support, 2–6

FORTRAN–77 support, 2–37

FORTRAN IV–PLUS, 2–37

Full-duplex mode
- considerations (communication driver), 2–31
- setting mode (communication driver), 2–28

G

GEN region name, 2–23

Get Device Information Junior function (GI.DVJ), 2–22

GIN$ directive
- GI.DVJ function, 2–22

GLUN$ macro
- get information
  - communication driver, 2–26

H

Half-duplex mode
- considerations (communication driver), 2–31
- set mode (communication driver), 2–29

HELLO, 1–12

HELP, 1–12

I

I/O sequence number
- DSA driver, 2–33

Indexed file
- RMS–11, 2–47

INI command
- /INF keyword, 2–10
- /MXF keyword, 2–10

INITIALIZE command

INITIALIZE command (Cont.)
- /HEADERS qualifier, 2–1
- /MAXIMUM_FILES qualifier, 2–1

Interface
- message-oriented (communication driver), 2–25

IO.FDX function (communication driver), 2–28

IO.HDX function
- set mode (communication driver), 2–29

IO.INL function (communication driver), 2–29
- after sync, 2–32

IO.RLB function (communication driver), 2–29

IO.RNS function (communication driver), 2–29

IO.SYN function (communication driver), 2–30
- set operation, 2–28
- with IO.INL, 2–32

IO.TRM function (communication driver), 2–29

IO.WNS function (communication driver), 2–30

IOSUB.MAC module
- modification, 2–20

IP11 correction file, 4–2

K

KMV11–A HDLC Framing Software, 4–2

KMV11 X25 Link Level, 4–2

L

Layered product
- correction files, 4–1

LOAD command, 2–11

Logical unit number
- See LUN

Low-traffic sync (communication driver), 2–31

LUN
- get information
  - communication driver, 2–26

M

MCR
- updating source file, C–4

Message-oriented interface (communication driver), 2–25

MIRA, 4–2

Mode
- maintenance (communication driver), 2–29
- set (communication driver), 2–28, 2–29

Monitor Console Routine
- See MCR

MSCP support
- RCT, 2–34

O

- ODL file
  - RMS–11, 3–8
- Offset P.SEQ
  - DSA driver, 2–33
- OPENREGISTER command
  - /TASKD keyword, 2–11
  - /TASKI keyword, 2–11
- OTS Fast Map routine
  - IOT restriction, 1–14
- Overlaid libraries
  - Manually loading, 1–11

P

- PDP–11 BASIC–PLUS–2 correction file, 4–1
- PDP–11 Pascal/RSX correction file, 4–2
- PDP–11 SORT/MERGE correction file, 4–2
- PDP–11 Symbolic Debugger/RSX correction file, 4–2
- PIP
  - wildcard support, 1–8
- Powerfail
  - DMC11 (communication driver), 2–31
- Powerfail recovery
  - DSA driver, 2–33, 2–34
- Programming hints
  - communication driver, 2–30

Q

- QIO$ macro
  - communication driver, 2–27
  - device-specific function
    - communication driver, 2–28
  - standard function
    - communication driver, 2–27
- Queue Manager, 1–12

R

- RCT
  - memory allocation, 1–12
  - MSCP support, 2–34
  - write-protected disks, 2–34
- Read operation
  - logical block (communication driver), 2–29
  - sync character (communication driver), 2–29
- Record Management Services
  - See RMS–11
- Redundancy checking (communication driver), 2–30
- Report Generator Task
- Report Generator Task (Cont.)
  - /HISTORY qualifier, 1–13, 2–16
- Resource Monitoring Display
  - See RMD
- RL02 installation kit, 3–3
- RLOG$ macro expansion, 2–18
- RLON$ macro expansion, 2–18
- RMD
  - BOTTOM command, 2–34
  - TOP command, 2–34
  - Writes Load Rates, 1–7
- RMS–11
  - accessing files on remote nodes, 3–9
  - ALLOCATION field, 2–47
  - alternate record key field, 2–44
  - BUCKETSIZE field, 2–47
  - building utilities, 3–8
  - contiguous file areas, 2–45
  - creating indexed file, 2–47
  - deadlock, 2–44
  - DEF command restriction, 3–3
  - DEFINE command restriction, 3–3
  - errors
    - ER$DNF, 3–2
    - ER$FUL, 2–44
    - ER$KEY, 2–44
    - ER$MRS, 2–44
    - ER$NOD, 2–44
  - /ER switch, 2–47
  - EXTENSION field, 2–47
  - file loading restriction, 3–2
  - files on the distribution kit, 3–3
  - file specification, 2–48, 3–2
  - local access, 3–1
  - nonamed directory mode restriction, 3–2
  - ODL file, 3–8
  - rebuilding utilities, 3–10
  - small systems, 3–3
  - startup command procedure, 3–7
  - task building against resident library, 2–45
  - Utilities, 3–2
- RMSBCK utility
  - command line format, 2–48
  - /NV switch, 2–48
- RMSCNV utility
  - /ER switch, 2–48
  - file loading restriction, 3–2
- RMSDAP
  - keysize field, 3–2
- RMSDEF utility
  - DEF command restriction, 3–3
  - DEFINE command restriction, 3–3

RMSDES utility
- continuation buckets, 3–3
- EXIT_SUPERSEDE command, 2–46, 3–3
- SAVE_SUPERSEDE command, 2–46, 3–3

RMSDSP utility
- /BR switch, 2–48
- /SU switch, 2–48

RMSRST utility
- command line format, 2–49
- /NV switch, 2–49

RSX–11 2780/3780 correction file, 4–2

RSX–11M–PLUS DIBOL correction file, 4–2

RUN command
- /PRI keyword, 2–13
- /UIC keyword, 2–12, 2–13

S

SAVE_SUPERSEDE command
- RMSDES utility, 2–46, 3–3

SET /CRASHDEV command, 2–14, 2–15

SET /CRASH_DEVICE command, 2–15

SET /DPRO command, 2–14

SET /INQUIRE command, 1–11, 2–14

SET/TERM command, 2–15

SET Command
- /INTRUSION qualifier, 1–9

Set mode
- half duplex (communication driver), 2–29
- IO.HDX function (communication driver), 2–29

SET SYSTEM /CRASH_DEVICE command, 2–5

SET TERMINAL/INQUIRE command, 2–1

SET TERMINAL/model command, 2–1

Shared regions
- I- and D-space restriction, 2–20

SHOW ASSIGNMENTS command, 2–3

SHOW LOGICALS command, 2–3

SHOW QUEUE command
- /ALL qualifier, 2–1

SHUTUP utility, C–8

SLP
- correcting source files, C–1

SORT command
- /INDEXED_SEQUENTIAL qualifier, 1–11

Source file
- correcting, C–1
- updating, C–1

Source Language Input Program
- See SLP

START Command
- /INTRUSION qualifier, 1–9

Sync character (communication driver), 2–30
- specifying, 2–30

System library
- using with TKB, 1–14

T

Task
- extending, 2–20
- installing, 1–10

Task Builder
- See TKB

Terminal driver, 1–10

Time delay
- DSA driver, 2–33

TKB
- allocating library APRs, 2–39
- Bitmask option, 1–11
- building new FCP, C–6
- CLSTR option, 2–39
- command file, C–3
- compatible library requirement, 1–14, C–12
- EXTK$ task directives, 1–2
- EXTM$ task directives, 1–2
- FMS command file example, 2–37
- Illegal APR reservation message, 2–42
- LIBR option, 2–39
- /LI subswitch, 2–39
- RESLIB option, 2–39
- RESSUP option, 2–38
- RMS–11 resident library, 2–45
- SUPLIB option, 2–38
- task-build command file, C–6
- using system library, 1–14
- using undefined symbols, 1–11
- VSECT option, 2–39

TLOG$ macro expansion, 2–19

TLON$ macro expansion, 2–19

Transmission
- validation (communication driver), 2–30

U

UPDATE operation
- RMS–11 correction, 2–44

V

/VERIFY qualifier
- BRU, 1–5

VFY
- warning message, 1–7

Virtual Monitor Console Routine

Virtual Monitor Console Routine (Cont.)
See VMR
VMR
INSTALL command, C–6
installing with deferred binding, 1–11
LOAD command, 1–12
REMOVE command, C–6
UNLOAD command, 1–12
updating a loadable driver, C–2
VMS ACP function support
FCS, 1–13

W

Write operation
no sync (communication driver), 2–30

X

XDT
loadable support, 2–16

HOW TO ORDER ADDITIONAL DOCUMENTATION

From	Call	Write
Alaska, Hawaii, or New Hampshire	603–884–6660	Digital Equipment Corporation P.O. Box CS2008 Nashua NH 03061
Rest of U.S.A. and Puerto Rico[1]	800–DIGITAL	
Canada	800–267–6219 (for software documentation) 613–592–5111 (for hardware documentation)	Digital Equipment of Canada Ltd. 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6 Attn: Direct Order Desk
Internal orders (for software documentation)	—	Software Supply Business (SSB) Digital Equipment Corporation Westminster MA 01473
Internal orders (for hardware documentation)	DTN: 234–4323 508–351–4323	Publishing & Circulation Services (P&CS) NRO3–1/W3 Digital Equipment Corporation Northboro MA 01532

[1] Prepaid orders from Puerto Rico, call Digital's local subsidiary (809–754–7575)

Reader's Comments

RSX-11M-PLUS
Release Notes
AA-H427H-TC

Your comments and suggestions will help us improve the quality of our future documentation. Please note that this form is for comments on documentation only.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (product works as described)	□	□	□	□
Completeness (enough information)	□	□	□	□
Clarity (easy to understand)	□	□	□	□
Organization (structure of subject matter)	□	□	□	□
Figures (useful)	□	□	□	□
Examples (useful)	□	□	□	□
Index (ability to find topic)	□	□	□	□
Page layout (easy to find information)	□	□	□	□

What I like best about this manual: ______________________________

____

What I like least about this manual: ______________________________

____

I found the following errors in this manual:

Page ______ Description ______________________________

______ ______________________________

______ ______________________________

______ ______________________________

My additional comments or suggestions for improving this manual:

____

____

____

Please indicate the type of user/reader that you most nearly represent:

□ Administrative Support
□ Computer Operator
□ Educator/Trainer
□ Programmer/Analyst
□ Sales

□ Scientist/Engineer
□ Software Support
□ System Manager
□ Other (please specify) ______________

Name/Title ______________________________ Dept. ______________

Company ______________________________ Date ______________

Mailing Address ______________________________

______________________________ Phone ______________

— — Do Not Tear — Fold Here and Tape —

digital™

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
CORPORATE USER PUBLICATIONS
PKO3-1/30D
129 PARKER STREET
MAYNARD, MA 01754-2198

— — Do Not Tear — Fold Here —

Cut Along Dotted Line

digital

Printed in U.S.A.