



# **RSX-11M-PLUS Release Notes**

Order Number: AA-H427L-TC

**January 1999**

# RSX-11M-PLUS Release Notes

Order Number: AA-H427L-TC

## January 1999

This revised manual contains the technical changes made to the RSX-11M-PLUS operating system since Version 4.5, documentation updates since Version 4.1, a summary of supported layered products, additional information on RMS-11 and details of Year 2000 Certification

**Revision/Update Information:** This revised manual supersedes the *RSX-11M-PLUS Version 4.5 Release Notes*  
AA-H427K-TC

**Operating System:** RSX-11M-PLUS Version 4.6

---

**January 1999**

Printed in the U.S.A.

---

Copyright ©1995-1999, Mentec, Inc., Previously - 1979, 1982, 1983, 1985, 1987, 1988-1993, Digital Equipment Corp.

In this document, references to Digital Equipment Corporation shall refer to Compaq Computer Corporation, Digital Products

Mentec Incorporated and Digital Equipment Corporation make no representations that the use of their products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained herein imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

Possession, use, or copying of the software described herein is authorized only pursuant to a valid written license from Mentec Incorporated, Digital Equipment Corporation (an authorized licensor) or an authorized sublicensor.

Mentec Incorporated makes no representation that all of the hardware and software described herein will be warranted or supported by its authorized licensors, or sublicensors now or at any time in the future.

Digital Equipment licensees and support contract customers should consult the *Digital Software Product Description* to determine the warranted and supported components. Some of the hardware devices and software described in this document are not warranted or supported by Digital Equipment Corporation.

The following are trademarks of Digital Equipment Corporation: DDCMP, DEC, DECnet, DEC/CMS, DEC/MMS, DECUS, DECwriter, DIBOL, LA50, LA75, LN01, LN03, LSI, MicroPDP-11, Micro/RXS, MicroVMS, PDP, Professional, Q-bus, RA70, RA82, RA90, RD31, RD32, RD50, RD54, ReGIS, RK07, RL01, RL02, RSTS, RSX, RSX-11, RSX-11M/S, RQZX1, RZ23L, RZ24L, TK50, TU81, TZ30, UNIBUS, VAX, VAXcluster, VAX DOCUMENT, VMS, VT, and the DIGITAL logo.

**Mentec, Inc.**

Nashua, New Hampshire

Phone: +1(603)883-7711

FAX: +1(603)883-7799

**Mentec Ltd.**

Dublin, Ireland

Phone: +353 1 2059797

FAX: +353 1 2059798

---

# Contents

---

## Preface

xi

---

## Chapter 1 New Features, Corrections, and Restrictions

1.1	Summary of New Features	1-1
1.2	Hardware Support in Version 4.6	1-2
1.2.1	New and Updated Hardware	1-2
1.2.2	Additional Hardware Information	1-2
1.2.2.1	KDJ11-E TOY Clock Restrictions	1-2
1.3	Distribution Kits	1-3
1.3.1	TK50 Single Tape Kit	1-3
1.3.2	RL02 Pregen Kit	1-3
1.4	Software Enhancements for Version 4.6	1-3
1.4.1	Year 2000 Certification	1-3
1.4.2	RSX and Inferencing	1-3
1.4.3	Year 2000 Enhancements	1-4
1.4.3.1	Two digit year support	1-5
1.4.3.2	Century inferencing support for two digit years	1-5
1.4.3.3	Four digit year support	1-5
1.4.3.4	ISO 8601:1988 Compliant Dates	1-6
1.4.3.5	Component Modifications for Year 2000	1-6
1.4.4	BOOT Enhancements	1-8
1.4.4.1	Hardware Boot	1-9
1.4.4.2	Bootable Virtual Disks	1-9
1.4.5	Enhancement to F11ACP to support I/D space buffers	1-10
1.4.6	FLX - Multi-partitioned RT-11 disks	1-11
1.4.7	Additional ICB pool savings	1-11
1.4.8	Large disk support	1-12
1.4.9	Utility UIC for Customer Specific Use	1-12
1.4.10	Utility Enhancements	1-12
1.4.10.1	BAD Enhancements	1-12
1.4.10.2	DCL Enhancements	1-13
1.4.10.3	DEV Enhancements	1-13
1.4.10.4	ICM Enhancements	1-14
1.4.10.5	RMD Enhancements	1-14
1.4.10.6	TKB Enhancements	1-14
1.4.10.7	PIP Enhancements	1-15
1.4.11	SYSLIB/FCSRES Enhancements	1-15
1.4.11.1	\$DAT Routines	1-15
1.4.11.2	\$CDDMG Enhancements	1-15
1.4.11.3	\$EDMSG Enhancements	1-15
1.4.12	Enhancement to DUDRV to allow for large disks	1-16
1.4.13	Online reconfiguration enhancements	1-16

1.4.14	New Vectored Symbols .....	1-17
1.4.15	Hardware Feature Mask Change .....	1-17
1.4.16	FCSRES rebuilt to use EIS Instructions .....	1-17
1.5	Software Problems Corrected in Version 4.6 .....	1-18
1.5.1	Corrections for Executive Routines .....	1-18
1.5.2	HRC would use incorrect drive type on some MSCP devices .....	1-18
1.5.3	KDJ11-E TOY clock correction .....	1-18
1.5.4	HELLO would incorrectly calculate the day of week .....	1-19
1.5.5	Corrections to BOOT Utility .....	1-19
1.5.6	INSTALL would allow non-privileged users to install ACPs .....	1-19
1.6	Software Problems and Restrictions in Version 4.6 .....	1-19
1.6.1	Restrictions in SYSGEN .....	1-19
1.6.1.1	Selecting FCPMDL in SYSGEN .....	1-19
1.6.1.2	Incorrect Displays for Multiple MM Devices in SYSGEN .....	1-19
1.6.1.3	Specifying a Nonexistent Device as the Crash Device During SYSGEN .....	1-20
1.6.2	VT Terminal Support .....	1-20
1.6.3	Error Logging Restrictions .....	1-20
1.6.3.1	Support for Error Logging History Summaries .....	1-20
1.6.3.2	Incorrect Displays of Register Fields for DU and MU Devices .....	1-20
1.6.4	BRU Restrictions .....	1-20
1.6.4.1	Use of Wildcards in BRU .....	1-20
1.6.4.2	Problem with the /NEW_VERSION Qualifier .....	1-20
1.6.5	DCL Command ABORT Taskname/TAS Does Not Translate Correctly ..	1-21
1.6.6	Tasks Hang When Attaching to A Terminal Already Attached to a Task .....	1-21
1.6.7	ELI/SH Reports No Errors if ELI/NOLIM Has Been Set .....	1-21
1.6.8	Problem with Unused Terminal Ports .....	1-21
1.6.9	Task Builder (TKB) Restriction .....	1-22
1.6.10	RMD Restriction .....	1-22
1.6.11	Indirect Processor Restrictions .....	1-22
1.6.11.1	IPP handles .END statements in an ENABLE data block incorrectly .....	1-22
1.6.11.2	IND handles .ONERR incorrectly with .BEGIN/.END blocks .....	1-22

---

## Chapter 2 Supplementary Software Information

2.1	Disk Save and Compress Utility .....	2-1
2.2	File Control Services (FCS) .....	2-1
2.3	Task Builder (TKB) .....	2-2
2.3.1	OTS Fast Map Restriction .....	2-2
2.3.2	Hybrid Fast Task Builder (HFT) .....	2-2
2.4	GIN Directives .....	2-4
2.4.1	GI.DVJ - Son of Get Device Information .....	2-4
2.4.2	GI.VEC - Translate Executive Entry Point Vector .....	2-5
2.5	Enhanced security features within HELLO, and ACNT .....	2-7
2.6	Shadow Load Balancing Support .....	2-7

2.6.1	Introduction	2-8
2.6.2	START/MERGE command	2-8
2.6.3	SET command	2-9
2.6.4	DISPLAY command	2-10
2.7	Corrections for Universal Receiver Task	2-11
2.8	SYSGEN Assumes no-FPP support	2-12

---

## Chapter 3 Corrections to Documentation

3.1	RSX-11M-PLUS Batch and Queue Manual	3-1
3.1.1	Enhancements to the PRINT and SUBMIT Commands	3-1
3.1.1.1	ASSIGN/QUEUE (QUE /AS: in MCR) Command	3-2
3.1.1.2	PRINT (PRI in MCR) Command	3-2
3.1.1.3	DEASSIGN/QUEUE (QUE /DEA in MCR) Command	3-3
3.2	RSX-11M-PLUS Command Language Manual	3-3
3.3	RSX-11M-PLUS Indirect Command Processor Manual	3-8
3.4	RSX-11M-PLUS MCR Operations Manual	3-9
3.5	RSX-11M-PLUS System Generation Guide and Installation Guide	3-18
3.6	RSX-11M-PLUS and Micro/RSX Crash Dump Analyzer Reference Manual	3-20
3.7	RSX-11M-PLUS and Micro/RSX Debugging Reference Manual	3-21
3.8	RSX-11M-PLUS and Micro/RSX Error Logging Manual	3-21
3.9	RSX-11M-PLUS and Micro/RSX Executive Reference Manual	3-21
3.9.1	The GINABO Call	3-22
3.9.2	The GINAPR Call	3-23
3.9.3	The GINDEF Call	3-23
3.9.4	The GINDEV Call	3-23
3.9.5	The GINDVJ Call	3-24
3.9.6	The GINFMK Call	3-24
3.9.7	The GINGAS Call	3-24
3.9.8	The GINQMC Call	3-25
3.9.9	The GINREN Call	3-25
3.9.10	The GINSPR Call	3-26
3.9.11	The GINTSK Call	3-26
3.9.12	The GINUAB Call	3-27
3.9.13	The GINUIC Call	3-27
3.9.14	The GINUPD Call	3-27
3.9.15	The GINVEC Call	3-28
3.9.16	Other Corrections to the RSX-11M-PLUS and Micro/RSX Executive Reference Manual	3-28
3.10	RSX-11M-PLUS and Micro/RSX Guide to Writing an I/O Driver	3-36
3.11	RSX-11M-PLUS and Micro/RSX I/O Drivers Reference Manual	3-37
3.12	RSX-11M-PLUS and Micro/RSX System Library Routines Reference Manual	3-46
3.13	RSX-11M-PLUS and Micro/RSX System Management Guide	3-46
3.14	RSX-11M-PLUS and Micro/RSX Task Builder Manual	3-49
3.15	RSX-11M-PLUS and Micro/RSX Utilities Manual	3-54
3.16	RSX-11M/M-PLUS and Micro/RSX I/O Operations Reference Manual	3-59
3.17	RSX-11M/M-PLUS RMS-11 Macro Programmer's Guide	3-61

3.18	RSX-11M/M-PLUS RMS-11 User's Guide .....	3-62
3.19	RSX-11M/M-PLUS RMS-11 Utilities .....	3-63

---

## Chapter 4 RMS-11 V2.0 Supplementary Information

4.1	RMS-11 Enhancements .....	4-1
4.2	RMS-11 Corrections .....	4-1
4.2.1	RMS-11 Local Access .....	4-1
4.2.2	RMS-11 Ease of Use .....	4-1
4.2.3	RMS-11 Utilities Enhancements .....	4-2
4.3	Software Restrictions .....	4-2
4.3.1	RMS-11 Access Methods .....	4-2
4.3.2	Restrictions to RMS-11 Remote Access Methods (RMSDAP) .....	4-2
4.3.2.1	DAP Date Restrictions .....	4-3
4.3.2.2	Incorrect Interpretation of Keysize Field Values .....	4-3
4.3.2.3	Using RMSDAP on Your System .....	4-3
4.3.3	RMS-11 Utilities .....	4-4
4.4	RMS-11 Supplementary Information .....	4-4
4.4.1	Enhancements to RMSDES Utility .....	4-4
4.4.2	RMS-11 Installation .....	4-5
4.4.2.1	Location of RMS-11 Files on the Distribution Kit .....	4-5
4.4.2.2	Startup Command Procedures .....	4-8
4.4.2.3	Utility Configurations .....	4-9
4.4.2.4	Utility Command and ODL Files .....	4-9
4.4.2.5	Rebuilding the Utilities .....	4-10

---

## Chapter 5 Layered Product Support

5.1	Correction Files .....	5-1
5.2	Installing Layered Products Restrictions .....	5-1
5.3	Invalid BASIC-PLUS-2 Installation Warning .....	5-2
5.3.1	BASIC-PLUS-2 DATE\$ Function .....	5-2
5.4	FORTTRAN-77 DATE/IDATE Functions .....	5-2
5.5	Datatrieve-11 Report Writer Heading .....	5-3

---

## Chapter 6 Virtual Device Subsystem Reference

6.1	Introduction .....	6-1
6.2	VCP Command Line .....	6-1
6.3	Using Virtual Devices .....	6-2
6.3.1	Setup the Virtual Device Units .....	6-3
6.3.2	Setting up Virtual Device Container Files .....	6-3
6.3.3	Using the devices .....	6-4
6.4	Commands .....	6-5
6.4.1	The CONnect command .....	6-5
6.4.2	The DISconnect command .....	6-9
6.4.3	The SHOW command .....	6-10

6.4.4	The DUMp command	6-11
6.4.5	The COpy command	6-11
6.4.6	The CREate command (Privileged)	6-12
6.4.7	The SET command (Privileged)	6-13
6.4.8	The STOp command (Privileged)	6-15
6.4.9	The TRAcE command (Privileged)	6-15
6.4.9.1	Tasks	6-15
6.4.9.2	Logical Block Numbers/Ranges	6-15
6.4.9.3	I/O function monitoring	6-16
6.4.9.4	Status attribute monitoring	6-16
6.4.9.5	Examples	6-16
6.5	Technical Overview	6-16
6.5.1	General	6-17
6.5.1.1	Access rights, and privileges	6-17
6.5.1.2	Create Function	6-17
6.5.2	Connect Function	6-18
6.5.2.1	File access	6-18
6.5.2.2	Device access	6-18
6.5.2.3	Virtual Unit selection	6-18
6.5.2.4	File linkage to virtual device	6-18
6.5.3	Disk Data Caching, and Volume Shadowing	6-19
6.5.3.1	Multiple Device Transfers	6-19
6.5.3.2	Shadowing	6-19
6.5.3.3	Disk Data Caching	6-19
6.5.4	Device Data Structures	6-19
6.5.5	U.DCBP,U.DCBS	6-22
6.5.6	U.IOPQ	6-22

---

## Chapter 7 Extended Security and Related Features

7.1	Account File Maintenance Utility (ACNT) changes	7-1
7.2	Password Expiration	7-2
7.3	Password Lifetime	7-2
7.4	Password Minimum Length	7-3
7.5	Disabling Account Access	7-4
7.5.1	Account Disabled	7-4
7.5.2	Remote Login Disabled	7-4
7.5.3	Network Login Disabled	7-4
7.5.4	Account expiration	7-5
7.6	System Inactivity Logout	7-6
7.6.1	Parameters	7-6
7.6.1.1	Inactivity logout timer	7-7
7.6.1.2	Warning message initial time remaining	7-7
7.6.1.3	Warning message time interval	7-7
7.6.2	Messages	7-7
7.6.3	Other conditions	7-7
7.7	Login Failure Disconnect	7-8

7.7.1	Login Date Display Options .....	7-8
7.7.2	BYE Enhancements .....	7-9

---

## Chapter 8 Year 2000 Certification for RSX-11M-PLUS Version 4.6

8.1	Date Presentation and Input .....	8-1
8.2	RSX and Inferencing .....	8-1
8.3	Internal Date Storage .....	8-2
8.4	Disk File System .....	8-2
	8.4.1 Files-11 File Headers .....	8-2
	8.4.1.1 ODS-I File Header Identification Area Description .....	8-3
	8.4.1.1.1 Date format to support year 2000 and beyond .....	8-4
	8.4.2 File system attributes .....	8-4
8.5	Magnetic Tape Standards .....	8-5
8.6	Foreign System Interchange .....	8-6
8.7	Backup Utilities .....	8-6
8.8	Networking .....	8-6
	8.8.1 Data Access Protocol V7.1 .....	8-6
	8.8.2 Network Management Protocol V4.2.0 .....	8-7
8.9	File System Utilities .....	8-7
	8.9.1 Utility Date Format .....	8-7
8.10	Assemblers, Compilers, and Linkers .....	8-7
8.11	Batch and Queue .....	8-8
	8.11.1 Queue Manager Processing Options .....	8-8
8.12	System Management .....	8-9
8.13	Certification .....	8-9

---

## Appendix A Reporting Problems

---

## Appendix B Applying Corrections to Source Files

B.1	Updating an Executive Source Module .....	B-1
B.2	Updating an MCR Source Module .....	B-2
B.3	Updating a DCL Source Module .....	B-4
B.4	Applying Corrections to the File Systems (F11ACP) .....	B-5
B.5	Applying Corrections to the Reconfiguration Tasks CON and HRC .....	B-6
B.6	Applying FCS Corrections .....	B-8
B.7	Updating the FCS Sources .....	B-8
B.8	Assembling FCS .....	B-9
B.9	Replacing the FCS Object Modules .....	B-11

---

## Index

---

## Tables

1-1	BAD command—Additional Switch Description . . . . .	1-13
1-2	DEVICE command—Additional Switch Description . . . . .	1-13
1-3	Indirect Date symbols . . . . .	1-14
1-4	ICP\$DT Definition Values . . . . .	1-14
1-5	Additional Date Output Routines . . . . .	1-15
2-1	GI.DVJ—Son of Get Device Information Subfunction Buffer Format . . . . .	2-4
2-2	GI.DVJ – Son of Get Device Information Subfunction DSW Return Codes and Their Meaning . . . . .	2-5
2-3	GI.VEC – Translate Executive Entry Point Vector Subfunction Buffer Format . . . . .	2-6
2-4	GI.VEC – Translate Executive Entry Point Vector Subfunction DSW Return Codes and Their Meaning . . . . .	2-7
3-1	Number of Index File Headers . . . . .	3-4
3-2	Default Number of Files – Approximation . . . . .	3-4
3-3	Number of Index File Headers . . . . .	3-13
3-4	Default Number of Files—Approximation . . . . .	3-13
3-5	SET Command Keyword Summary . . . . .	3-16
3-6	Summary of BRU Command Qualifiers . . . . .	3-54
4-1	RMS-11 Files on RSX-11M-PLUS Distribution Kit . . . . .	4-5
5-1	Layered Product Correction File References . . . . .	5-1
8-1	Identification Area . . . . .	8-3
8-2	Storage of Date Fields after 1990 . . . . .	8-4
8-3	F11ACP Date attributes . . . . .	8-5
8-4	RSX-11M-PLUS ANSI tape dates . . . . .	8-5
8-5	RSX-11M-PLUS Utilities Tested . . . . .	8-10



---

# Preface

---

## Manual Objectives

The *RSX-11M-PLUS Release Notes* contains important information for using the RSX-11M-PLUS Version 4.6 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
- Year 2000 Certification details

---

## Intended Audience

This manual is intended for all users of the RSX-11M-PLUS Version 4.6 operating system.

---

## Document Structure

Chapter 1 and Chapter 2 describe software features, software and hardware restrictions, and supplementary information for the RSX-11M-PLUS Version 4.6.

Chapter 3 corrects errors and omissions in the RSX-11M-PLUS documentation set.

Chapter 4 describes software modifications, restrictions, and supplementary information for RMS-11 Version 2.0.

Chapter 5 contains correction files and restrictions for layered products.

Chapter 6 contains a description of the Virtual Device utility.

Chapter 7 describes the enhancements to RSX system login security.

Chapter 8 describes the Year 2000 certification of RSX-11M-PLUS Version 4.6.

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

Appendix B 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
[option]	Brackets indicate optional entries in a command line or a file specification. If the brackets include syntactical elements, such as periods (.) or slashes (/), those elements are required for the field. If the field appears in lowercase, you are to substitute a valid command element if you include the field. Note that when an option is entered, the brackets are not included in the command line.
[, ... ]	Brackets around a comma and horizontal ellipsis points indicate that you can use a series of optional elements separated by commas. For example, (argument[, ... ]) means that you can specify a series of optional arguments by enclosing the arguments in parentheses and by separating them with commas.
{ }	Braces indicate a choice among required options. You must choose one of the options listed.
:argument	Some parameters and qualifiers can be altered by the inclusion of arguments preceded by a colon. An argument can be either numerical (COPIES:3) or alphabetical (NAME:QIX). In DCL, the equal sign (=) can be substituted for the colon to introduce arguments. COPIES=3 and COPIES:3 are the same.
( )	<p>Parentheses are used to enclose more than one argument in a command line. For example:</p> <pre data-bbox="886 1041 1211 1062">SET PROT = (S:RWED,O:RWED)</pre>
,	<p>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 within the command line. Although you might omit elements that come before the desired element, the commas that separate them must still be included.</p>
.	<p>Periods immediately following a number indicate a decimal number. For example, 600. would mean 600<sub>10</sub>.</p>

## Preface

Convention	Meaning
[g,m] [directory]	<p>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.</p> <p>This may also signify a User File Directory (UFD), commonly called a directory. A directory is the location of files.</p> <p>Other notations for directories are: [ggg,mmm], [gggmmm], [ufd], [name], and [directory].</p> <p>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.</p> <p>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]]).</p> <p>Vertical ellipsis points show where elements of command input or statements in an example or figure have been omitted because they are irrelevant to the point being discussed.</p>
filespec	<p>A full file specification includes device, directory, file name, file type, and version number, as shown in the following example:</p> <pre>DL2:[46,63]INDIRECT.TXT;3</pre> <p>Full file specifications are rarely needed. If you do not provide a version number, the operating system uses the highest numbered version. If you do not provide a directory, the system uses the default directory. Some system functions default to particular file types. Many commands accept a wildcard character (*) in place of the file name, file type, or version number. Some commands accept a file specification with a DECnet node name.</p> <p>A period in a file specification separates the file name and file type. When the file type is not specified, the period may be omitted from the file specification.</p> <p>A semicolon in a file specification separates the file type from the file version. If the version is not specified, the semicolon may be omitted from the file specification.</p>
@	<p>The at sign invokes an indirect command file. The at sign immediately precedes the file specification for the indirect command file, as follows:</p> <pre>@filename[.filetype;version]</pre>

Convention	Meaning
...	Horizontal ellipsis points indicate the following: <ul style="list-style-type: none"><li>• Additional, optional arguments in a statement have been omitted.</li><li>• The preceding item or items can be repeated one or more times.</li><li>• Additional parameters, values, or other information can be entered.</li></ul>
KEYNAME	This typeface denotes one of the keys on the terminal keyboard; for example, the Return key.
print and enter	As these words are used in the text, the system prints and the user enters.
<code>Ctrl/x</code>	The symbol <code>Ctrl/x</code> means that you are to press the key marked Ctrl while pressing another key. Thus, <code>Ctrl/Z</code> indicates that you are to press the Ctrl key and the Z key together in this fashion. Ctrl/Z is echoed on some terminals as ^Z. However, not all control characters echo.



# 1

---

## New Features, Corrections, and Restrictions

This chapter describes new software features, including new hardware support, lists software problems corrected in this release, and lists software and hardware restrictions for the RSX-11M-PLUS Version 4.6 operating system.

---

### 1.1 Summary of New Features

The RSX-11M-PLUS Version 4.6 operating system supports the following new features:

- Year 2000 Enhancements and Compliance Statement
  - ISO 8601:1988 compliant date format
  - Selectable Year 2000 Compliant date formats
  - Century inferencing support
  - Utility Corrections
- Enhancement to F11ACP to support I/D space buffers
- Additional ICB pool available for primary pool use.
- Increased large disk support including RZ29B
- Full 24-bit logical block support for disks
- New Mentec processor support
- Enhancement to support booting virtual disks.
- Customer specific application UIC
- Enhancements to Utilities

BAD	BOO	DCL
DEV	FLX	ICM
PIP	RMD	TKB
- Enhancements to SYSLIB/FCSRES Routines

\$DAT	\$DATS	\$DATI
\$DAT4	\$CDDMG	\$EDMSG
Fortran DATES	DATEI	DATE4
- Online reconfiguration enhancements
- New Vectored Executive Symbols
- Hardware Feature Mask Updated
- Single Tape Distribution for TK50 Media

---

### 1.2 Hardware Support in Version 4.6

Support has been added or updated for the following devices:

---

#### 1.2.1 New and Updated Hardware

- RZ29B (4.3 Gb, using RQZX1 controller)
- MENTEC M-Series processors:
  - M70, M80, M90, M100 series (DCJ11 based processors)
  - M11 (DCJ11 emulation)
  - M1 (DCJ11 emulation) (*New*)

---

#### 1.2.2 Additional Hardware Information

For a complete list of hardware supported by Version 4.6, refer to the *Software Product Description*.

---

##### 1.2.2.1 KDJ11-E TOY Clock Restrictions

On systems using Digital KDJ11-E processors (i.e. 11/93 and 11/94 systems) there are concerns related to the use of the TOY clock when operating with Console Monitor Version 1.06. The Console Monitor is part of the boot firmware for these processors. *These concerns do not apply to processors using the Console Monitor firmware Version 2.01*

These concerns are summarized below:

- The year 2000 is not considered to be a leap year.
- The only dates which are considered valid when entered by keyboard to the console monitor range from 1-JAN-1990 through 31-DEC-2009.
- The day of the week may not be calculated correctly beyond the year 1999.
- Dates considered invalid by the firmware will result in a stored date of 1-Jan-90.

By using the MCR TIME or DCL SET TIME commands these concerns can be bypassed, but only until the system is rebooted. When rebooted on a date which the firmware considers invalid, the date will be retrieved by RSX from the TOY clock as 1-JAN-90.

Again, the user must specify the correct date and time, and include the "/SETTOY" option to reset the date and time to the current values.

An upgrade to the Console Monitor Firmware is available from Digital and customers are advised to upgrade their firmware from to Version 2.01 as soon as possible, since this will have an impact on the "Year 2000 Readiness" of your system. While systems with attended operation can work around this problem, systems designed for unattended operation can not do so directly.

---

### 1.3 Distribution Kits

#### 1.3.1 TK50 Single Tape Kit

The RSX-11M-Plus Version 4.6 Kit distributed on TK50 tape now comprises a single cartridge, with BRUSYS, Baseline System, RSX-11M-Plus Installation Backup sets, Layered Product Corrections and the RSX-11M-Plus RL02 Pregen on the tape in that order.

Recent versions of BRU feature the ability to skip over the bootable BRUSYS portion of the tape, making a single tape distribution feasible.

---

#### 1.3.2 RL02 Pregen Kit

The RL02 was previously retired as a distribution media, but an RL02 Pregenerated system is available on the distribution kit and can be restored onto a single RL02 drive. e.g. -

```
>BRU /REW/INI/BAC:RSXMPRL02 MU0: DL0:
```

For further information, see the Chapter on *Corrections to Documentation* section on *RL02 Pregenerated System*.

---

### 1.4 Software Enhancements for Version 4.6

This section describes software enhancements included in Version 4.6.

---

#### 1.4.1 Year 2000 Certification

RSX-11M-PLUS Version 4.6 includes numerous upgrades to meet international standards for Year 2000 compliance. It has been exhaustively tested and has been certified against those standards. A description of the standards and certification performed for RSX-11M-PLUS Version 4.6 is contained in Chapter 8. Details of conformance for RSX utilities and the current known date limitations are listed.

---

#### 1.4.2 RSX and Inferencing

In the discussions of Year 2000, the term inferencing occurs widely in discussing 2 digit year formats. When a date is input to RSX-11M-PLUS Version 4.6 with a 2 digit year, it is necessary to determine a full 4 digit year that this 2 digit year is meant to represent, to maintain compatibility. This process is referred to as inferencing. Defined inferencing does meet the international standards.

On input, RSX-11M-PLUS Version 4.6, as shipped, will infer 2 digit years 65 through 99 as 1965 through 1999, and 2 digit years 00 through 64 as 2000 through 2064. It is possible to change the base year of 1965 (described later.)

On output, RSX-11M-PLUS Version 4.6 will obey the same rules when outputting 2 digit dates, and the correct century can be inferred on sight.

## New Features, Corrections, and Restrictions

The following functions have been enhanced to allow for century inferencing when using two-digit year fields in commands. This will permit the input of a two digit year to be inferred to the years 1965—2064 (or as modified by the currently specified inferencing base year).

- BRU /BEFORE/AFTER
- PIP /DD
- QUE /AF
- RPT /DA
- TIM
- VMR TIME

Below is an example of how dates with 2 digit years will be inferred to 4 digit years with the default base year:

RSX Date	ISO Date	Actual Date
01-JAN-98	1998-01-01	01-JAN-1998
01-JAN-00	2000-01-01	01-JAN-2000
01-JAN-15	2015-01-01	01-JAN-2015
01-JAN-66	1966-01-01	01-JAN-1966
01-JAN-64	2064-01-01	01-JAN-2064

### 1.4.3 Year 2000 Enhancements

RSX-11M-PLUS Version 4.6 has been modified to use any one of the following three date formats as the system default:

DD-MMM-YY	Traditional date format
DD-MMM-YYYY	Four digit year format
YYYY-MM-DD	ISO 8601:1988 compliant date format

This includes the traditional two-digit year date which by default is inferred to be in the range 1965 through 2064, a full four-digit year date, and an ISO 8601:1988 complaint date format.

These are system wide date formats which can be selected at either sysgen time, or via the host reconfiguration utility (CON).

Dates may be input in any of these three formats, whatever the default setting for date output.

Dates representations are now handled as follows by RSX-11M-PLUS Version 4.6

- Input—All system applications that allow a date to be input will now accept any of the three date formats.
- Output—All system applications will output dates in only one of the three date formats at any given time. The selected output format is initially determined in SYSGEN and may later be changed on line.

## New Features, Corrections, and Restrictions

- Internal representations—As documented for the GTIM\$ directive, the year field of time vectors is the number of years since 1900. This field will reach 100<sub>10</sub> in the year 2000.
- Applications—The format of the date input and output by applications is totally dependent upon the application. Applications using RSX system routines \$DAT and \$EDMSG for processing dates on output will continue to output dates correctly. Applications processing dates in other manners may or may not produce valid dates.

---

### 1.4.3.1 Two digit year support

All utilities by default include support for traditional two digit years subject to date inferencing (see below). Two digit year output may be selected online with the

```
>CON SET SYS Y2
```

host reconfiguration command.

---

### 1.4.3.2 Century inferencing support for two digit years

An additional module has been included within SYSLIB, which defines the base year for century inferencing functions. This module, Y2KDEF, contains a single global value, \$Y2KHI, which defines the lowest year to be considered to be in the twentieth century. This value is currently defined to be 65.

The base year for inferencing may be changed by replacing this module, and all the above utilities and any user applications using date inferencing must be rebuilt using the replaced module.

Note that by setting this value to 99, the 2 digit year inferencing can extend from 1999 through 2098. It is not possible to span the entire 21st century (2000 through 2099) by inferencing.

Below is an example of how the utilities use the definition for the purposes of century inferencing:

```
CALL GETDC1 ; OBTAIN THE YEAR AS A DECIMAL #
CMP R0,$Y2KHI ; LOOK AT YEAR VALUE
BHIS 53$ ; WE ARE IN THIS CENTURY

ADD #100.,R0 ; BIAS TO Y2K
BR 55$ ; AND CONTINUE

53$: CMP R0,#100. ; THE YEAR MUST ALSO BE < 2000
BLT 55$ ; IF LT OK
SUB #1900.,R0 ; ADJUST BY 1900
BLT DATERR ; IF OUT-OF-BOUNDS, SAY SORRY

55$:
```

---

### 1.4.3.3 Four digit year support

All utilities include support for accepting four digit years. Four digit years may be used in all places within the system where two digit years were previously accepted.

In order to use this functionality, this must be selected in system generation, at the prompt

## New Features, Corrections, and Restrictions

```
;
; CE216 Do you want the default date format to include
; a four digit year
;
```

or via the

```
>CON SET SYS Y4
```

host reconfiguration command.

---

### 1.4.3.4 ISO 8601:1988 Compliant Dates

RSX-11M-PLUS Version 4.6 includes full support for ISO 8601:1988 compliant dates. In order to use this functionality, this must be selected in system generation, at the prompt

```
;
; CE215 Do you want the default date format to be
; ISO 8601:1988 compliant
;
```

or via the

```
>CON SET SYS ISO
```

host reconfiguration command.

---

### 1.4.3.5 Component Modifications for Year 2000

Most components of RSX-11M-PLUS were modified for the previous two versions to be operable in the year 2000 and beyond, however, the following components have been modified for Version 4.6 to correct identified problems and to provide updated support for standards compliance for the year 2000:

TIM (MCR/DCL)	CLQ (MCR/DCL)	QUE (MCR/DCL)
BRU	EDT	FLX
PIP	RPT/ERRLOG	DMP
INI/HOM	VMR TIM	MACRO-11
Help files		
Layered products		
BASIC-PLUS-2	FORTTRAN-77	Datatreve-11

#### MCR/DCL TIME Command

The output year from the TIME command has been updated to display the year in system selected output format (Y2, Y4, or ISO). The TIME command has also been updated to perform year inferencing on input according to the rules for inferencing described earlier.

For KDJ11-E based systems (11/93 and 11/94) with a TOY clock, the TIME command has been modified to infer the century from the data retrieved from the Time of Year (TOY) clock (if available). The updated firmware for the KDJ11-E TOY clock will operate without interference up to 2050. The century information is not stored in the TOY clock itself, but is inferred.

For dates beyond 2050, RSX-11M-Plus will infer the century using the day of the week register, and can support dates through 2199. Whenever the KDJ11-E based system is powered up, after Jan 1, 2051, the TOY clock must be set using MCR/DCL TIME command and not from the console firmware,

## **New Features, Corrections, and Restrictions**

since the firmware will attempt to set the day of the week based on the date range of 1951 to 2050.

For Mentec M-Series processors, this restriction does not apply, since the user must enter the day of the week to the TOY clock in the firmware.

The valid range of dates which are supported for Version 4.6 is from 1965 through 2064.

### **MCR/DCL CLOCK QUEUE Command**

MCR CLQ command has been updated to correctly display the time and date a task in the queue is next scheduled to run.

```
>CLQ  
SEN... Scheduled at 1998-12-15 12:07:08:01 Reschedule interval 20 Mins
```

### **BRU Dates**

BRU has been updated to use 4 digit and ISO standard date formats. BRU will also perform 2 digit year inferencing on input according to the current inferencing rules.

### **EDT Date Command**

Using EDT, in Nokeypad mode, it is possible to insert a date string into the text at the current cursor position with the "DATE" command. Previously, EDT would insert a 2 digit year prefixed by the characters "19". This has been updated to insert the date in the current selected system format: Y2, Y4 or ISO.

### **ERRLOG Date handling**

ERRLOG has been updated to permit the handling of dates up to 2155

### **RPT Date Range Switch**

RPT has been updated to accept and display dates in the currently selected system format. Previously it would not permit dates to be input with years after 1999. RPT will inference 2 digit years according to the defined inferencing rules.

### **FLX DOS and RT Year 2000 Date Update**

FLX has been updated to output dates in the currently defined system format and correctly interpreted DOS and RT-11 file structure dates past the Year 2000. The maximum year for a DOS format structure is 2035. The maximum year for an RT-11 disk is 2099.

### **INI / HOM Disk Home Block Dates Updated**

The Initialize (INI) and Home Block (HOM) commands have been updated to use 4 digit creation and revision dates in a Files-11 Disk home block.

### **PIP Date Range Comparison**

PIP has been updated to output and accept dates in the selected system format, and to correct a problem where dates were compared using signed branches, which could result in incorrect comparisons. PIP will now inference 2 digit years according to the defined inferencing rules.

## New Features, Corrections, and Restrictions

### QUE Date references

QUE has been updated to accept dates in the defined formats and correctly inference 2 digit years according to the defined inferencing rules.

### DMP Header ID area display

DMP has been updated to output dates in the selected system format when displaying file header information.

e.g.

```
>dmp ti:=bytest.ftn/hd
```

```
Dump of DU0:[1,376]BYTEST.FTN;9 - File ID 20274,1,0
                                         File header
```

```
HEADER AREA
      H.IDOF      027
      H.MPOF      056
      .....
IDENTIFICATION AREA
      I.FNAM,
      I.FTYP,
      I.FVER      BYTEST.TMP;1
      I.RVNO      1
      I.RVDT      1998-11-30
      I.RVTI      14:35:46
      I.CRDT      1998-11-30
      I.CRTI      14:35:46
      I.EXDT      --
      ....
```

### VMR TIM Command

The VMR TIM command has been updated to allow 2 or 4 digit year or an ISO date field in the same form as the MCR TIM command and will inference a 2 digit year according to the defined inferencing rules.

### Macro-11 Listings

MACRO-11 assembler has been update to display the year in the banner lines in the list file in the currently defined system format. Previously it would display it relative to 1900, thus displaying 100 for the year 2000. e.g.

```
KCTAB  MACRO V05.05  Tuesday 1998-12-15 12:33  Page 1-1
```

### HELP Files

HELP files have been updated to include references where applicable to 4 digit years and ISO dates for the various commands that use dates.

### Layered Product Corrections

The description of the corrections for the layered products is given in Chapter 5.

---

## 1.4.4 BOOT Enhancements

The BOOT command and the bootstrap software has been updated for Version 4.6 to provide the ability to do a hardware boot (permitting booting of non-RSX systems) and to boot virtual disks.

---

### 1.4.4.1 Hardware Boot

The BOOT utility has been enhanced to include the /HW switch, to be used for performing a hardware bootstrap on a given disk. The device must be mounted foreign, and the command sequence is specified as:

```
>MOU ddnn:/FOR
>BOO ddnn:/HW
```

This command is used to load in the boot block from the specified volume, and transfer control to the boot block. This function permits the booting of non-RSX disks.

Note that BOO/HW cannot boot a physical or virtual tape.

---

### 1.4.4.2 Bootable Virtual Disks

RSX-11M-PLUS V4.6 now has the capability to permit a hardware boot of a saved system image on a virtual disk. Modifications to provide this functionality were made to VCP, BOOT, BRU, SAVE, and the Executive.

When using this capability, the following conditions must be observed:

- The running and new system must be generated to support virtual devices.
- Only contiguous disk files may be made bootable
- Only full physical disk copies will remain hardware bootable.
- The virtual disk to be made bootable must not be a virtual "sub-disk". In other words, it must not be a virtual contained on a higher level virtual disk.

To permit hardware booting of a virtual disk, the hardware boot block of the physical disk on which the virtual is stored must be updated to start the bootstrap on the virtual disk. Thus, it is not possible to have more than one hardware bootable system on any one physical volume. The act of performing the Save - Write Bootstrap function will supercede any previous bootstrap. Consequently, creating a virtual disk based system on your system disk and making it bootable will result in the current system no longer being hardware bootable.

In order to generate a bootable virtual disk system, an RSX-11M-PLUS V4.6 system must first be generated on a physical device, then copied to a virtual disk, and subsequently booted and saved. It is not possible to generate a hardware bootable virtual disk system directly from either the baseline system, or from RSX-11M-Plus Version 4.5 or earlier.

The following steps may be used to generate a bootable virtual disk:

- Generate an RSX-11M-PLUS V4.6 system.
- Boot and SAVE the new system.
- Create a virtual disk, and copy the system to that virtual disk
- Re-VMR the system on the virtual device.
- Software boot, and SAVE the new system on the virtual disk.

## New Features, Corrections, and Restrictions

When the system is booted from a virtual device, the system disk will be called VF0:, and will be connected automatically by SAVE during startup.

Below is an example of the output when booting a virtual disk:

```
>BOO VF:[1,54]RSX11M.SYS
RSX-11M-PLUS V4.6 BL86
>SAV /WB/MOU="/ACP=UNIQUE"
RSX-11M-PLUS V4.6 BL86 2044.KW System:"PHEANX"
>RED VF0:=-SY:
>RED VF0:=-LB:
>RED VF0:=-SP:
>MOU DU0:"RSX11MPLUS"/ACP=UNIQUE
>MOU VF0:"RSX11MPBL86"/ACP=UNIQUE
>@VF0:[1,2]STARTUP.CMD
```

When using bootable virtual devices, it is advisable to use the SAVE "/MOUNT" switch such that each device has a "unique" ACP, to increase system throughput, particularly if the primary boot device is also used as an application disk.

Refer to Chapter 6 for more information on bootable virtual disks. Note that it is not possible to boot virtual tapes.

### 1.4.5 Enhancement to F11ACP to support I/D space buffers

RSX-11M-PLUS Version 4.6 has included support for F11ACP to be built using separated I/D space, which increases the amount of buffer space available in FCPLRG.

If the system is built using FCPLRG, and supports separated I/D space, SYSGEN will include the line

```
INS [3,54]FCPLRGID
```

in the SYSVMR command file. In comparison with the non-I/D space flavor of FCPLRG, there is room for an additional 200 file control buffers (FCBs) within the ACP's address space.

If desired, the original FCPLRG install command can be inserted, for use on all systems.

Any applications which access the data areas within the ACP will have to be modified in order to be able to support an I/D space ACP.

Below is an example from the MCR module DEVOV to access the file control block chain within the ACP address space, when using an I/D space ACP:

```
;+
; Note: This example is not complete, and is only intended for the
; reference purposes. It is not intended to handle all cases
; which are possible.
;
; Input:
; R4 - UCB address of device
; R2 - TCB address of ACP (U.ACP)
; R1 - PCB address of ACP (T.PCB)
;-
CALL $SWSTK,260$ ; switch to the kernel stack
```

## New Features, Corrections, and Restrictions

```
CLR      -(SP)                ;; set up an accumulator
BIT      #T4.DSP,T.ST4(R2)    ;; is this an I/D space ACP
BEQ      14$                  ;; if EQ, nope, use header length
;+
; For an I/D space ACP, we need to access the offset to window number 1,
; which is the D-space root for the task. For a non-I/D space ACP, we only
; need to use the header length in order to access the ACP buffers.
;-
MOV      P.REL(R1),@KSAR5     ;; map to the ACP header
MOV      @#12000+H.WND,R2     ;; get the pointer to the windows
MOV      <<2-20000>+W.BLGH+W.BOFF>(R2),(SP) ;; get the offset
BR       15$                  ;; and continue

14$:     MOVB    P.HDLN(R1),(SP) ;; save the header length
15$:     ADD     P.REL(R1),(SP)  ;; and form an offset
        MOV     U.VCB(R4),R2    ;; and ensure the VCB is available
        MOV     #MAXFIL,R3     ;; maximum number of slots
        ADD     #V.FCB,R2      ;; point to the FCB chain

20$:     MOV     (R2),R2        ;; get next FCB entry
```

### 1.4.6 FLX - Multi-partitioned RT-11 disks

RT-11 disks are capable of supporting up to 256 64K block partitions on a single drive under the RT-11 Operating System. FLX has been updated to provide support for these multiple partitions on an RT-11 disk.

```
>FLX
FLX>DU12:/RT:2/ZE
FLX>DU12:/RT:2=BYTEST.FTN
FLX>DU12:/RT:2/LI

Directory      DU12: [Partition:2.]
1998-12-17

BYTEST.FTN    1. 1998-12-17
< Unused > 65466.

65466. Free blocks

Total of 1. blocks in 1. files

FLX>^Z
```

### 1.4.7 Additional ICB pool savings

RSX-11M-PLUS Version 4.6 includes support for expanding Interrupt Control Block (ICB) Pool on the system. The following changes have been made:

- Merge crash and powerfail stacks together
- Merge crash device data structures for loadable crash drivers.
- Expand the top of ICB pool to the maximum possible address.

In order to maximize the available ICB pool, be sure not to include resident XDT, and also use a resident crash driver, rather than a loadable crash driver.

In order to maximize the total amount of ICB pool available, the I-space portion of the executive, which resides above APR 0, has been moved from the blank PSECT into a named PSECT of EXEC1. If additional modules are included into the executive, the following conditional should be included at the beginning of the source code:

## New Features, Corrections, and Restrictions

```
.IIF DF,K$$DAS&I$$CBP, .PSECT EXEC1
```

---

### 1.4.8 Large disk support

RSX-11M-PLUS Version 4.6 includes support for MSCP disks to access all blocks on disk drives up to a maximum size of 16,777,215 blocks (approx 8.6GB). Disks larger than 16,777,215 block, will be truncated by DUDRV at the maximum supported size.

At this time the largest supported disk with full support is the Digital RZ29B SCSI drive, which at approx. 4.3Gb is roughly half the maximum supported drive size.

This support has also been included in stand-alone BRU systems (BRUSYS).

---

### 1.4.9 Utility UIC for Customer Specific Use

The INSTALL utility, and SET command have been modified to include support for a customer specific utility UIC to help with application management.

The use is identical to the other system UICs available, i.e. SYSUIC, LIBUIC, and NETUIC. Below is the command syntax for setting and displaying the utility UIC:

```
>SET /UTLUIC[=[ggg,mmm]]
$SHO SYSTEM/UTILITY_UIC
$SET SYSTEM/UTILITY_UIC:[ggg,mmm]
```

The INSTALL utility now includes the utility UIC (UTLUIC) when searching for tasks to be installed. The directory search list consists of the following directories:

- 1 Utility UIC (UTLUIC)
- 2 Library UIC (LIBUIC)
- 3 System UIC (SYSUIC)

The Utility UIC is reserved for customer specific use. If it is set to [0,0], it will not be used, and the search will proceed directly to the Library UIC.

---

### 1.4.10 Utility Enhancements

---

#### 1.4.10.1 BAD Enhancements

The BAD utility has been enhanced to allow for only checking if all blocks on a disk can be read, rather than the traditional method of writing, reading and comparing the data. Although the traditional method is more thorough, on large disks, the time required may be exhaustive.

**Table 1-1 BAD command—Additional Switch Description**

Switch	Meaning
/READ	Perform Readonly check of device

**1.4.10.2 DCL Enhancements**

DCL has been updated to include the DUMP command. Refer to the Documentation Changes section for information regarding the format of the command.

In addition, DCL accepts each of the following formats for date entry:

DD- <i>MMM</i> -YY	Traditional date format
DD- <i>MMM</i> -YYYY	Four digit year format
YYYY-MM-DD	ISO 8601:1988 compliant date format

**1.4.10.3 DEV Enhancements**

The MCR DEVICE, and DCL SHOW DEVICE commands have been enhanced with the following additional switches:

**Table 1-2 DEVICE command—Additional Switch Description**

Switch	Meaning
/FUL	Display ACP summary of FCBs
/FIL	Display open files on mounted devices

```
>DEV DU1:/FULL/FILE
$ SHOW DEV/FULL/FILE DU1:
DU1:   Public Mounted Loaded Label=RSX11MPLUS Type=WDE91
      Shadow_set=(DU1:,DU0:)  Cached
      Free_blocks=4230195. Trans_count=11.
      File: RA60SYSA.DSK;1,      Device: VF0:
      File: RSXMASTER.DSK;1,    Device: DU10:
      File: RM05X.DSK;1,        Device: DR0:
      File: RM05A.DSK;1,        Device: DR1:
      File: RM05B.DSK;1,        Device: DR2:
      File: RM05C.DSK;1,        Device: DR3:
      File: RM05D.DSK;1,        Device: DR4:
      File: RSX11M.DSK;1,       Device: DB0:
      File: NETMASTER.DSK;1,    Device: DB1:
      File: ACNTRN.SYS;351,     Task: SYSLOG
      File: MPBL86DAT.TAP;1,    Device: MU3:

      FCBs: 0. (pool), 163. (ACP), 3. (directory)
```

This example shows the open files on the device, and the task or virtual device that they are associated with. The last line shows a summary of how many FCBs are in pool, or the ACP's address space, and how many are associated with a directory LRU. The last line is the result of the /FUL switch.

## New Features, Corrections, and Restrictions

### 1.4.10.4 ICM Enhancements

Indirect has included an additional string symbol <DATE4> which is of the format DD-*MMM*-YYYY, and the original symbol <DATE> has been modified to support the system specific date format. Below describes the new functionality of the date symbols:

**Table 1-3 Indirect Date symbols**

Symbol	Description
<DATE>	Assigned the current date, in system specific format. One of: DD- <i>MMM</i> -YY, DD- <i>MMM</i> -YYYY, or YYYY-MM-DD
<DATE4>	Assigned the current date, in DD- <i>MMM</i> -YYYY format

If desired, the original functionality of the <DATE> symbol can be restored by rebuilding ICM, and modifying the Global Definition within the TKB command file, shown below:

```
GBLDEF=ICP$DT:1
```

The following definitions will define the behaviour of the <DATE> symbol:

**Table 1-4 ICP\$DT Definition Values**

0	Traditional date format
1	System specific format. One of: DD- <i>MMM</i> -YY, DD- <i>MMM</i> -YYYY, or YYYY-MM-DD

### 1.4.10.5 RMD Enhancements

RMD has been updated to correctly display the number of free blocks on disks up to the current maximum supported disk size.

### 1.4.10.6 TKB Enhancements

The Task Builder (TKB) has been enhanced to include a new option, RNDSCCT, which specifies the size that a resulting PSECT should be. The format of the option is:

```
RNDSCCT=psect:size
```

where "psect" is the Program Section (PSECT) to affect, and "size" is the size in octal bytes of the resulting program section.

---

**1.4.10.7 PIP Enhancements**

PIP has been updated to allow the use of the /PAGE option when outputting a file to the users terminal. When used, PIP will output the number of lines specified in the terminal characteristics, and prompt to continue. This option is only applicable when typing a file out on the users terminal.

Below is an example of the command:

```
>PIP TI:=FILE.TXT/PA
$ TYPE/PAGE FILE.TXT
```

---

**1.4.11 SYSLIB/FCSRES Enhancements**

---

**1.4.11.1 \$DAT Routines**

Three additional date routines have been included into SYSLIB, and FCSRES with RSX-11M-PLUS Version 4.6. These routines are called in the same manner as the \$DAT routine. Word 1 of the Date value input area is an integer representing the number of years from 1900 (i.e. 99 for 1999, 100 for 2000) Word 2 is an integer from 1 to 12 representing the month. Word 3 is an integer representing the day of the month. The length of the output string will depend on the particular date and the particular \$DAT variant called.

In addition, the FORTRAN-77 callable subroutines, identical to DATE are also included. These are described below:

**Table 1-5 Additional Date Output Routines**

Macro	Fortran	Description
\$DAT	DATE	Convert the current date in DD- <i>MMM</i> -YY format
\$DAT4	DATE4	Convert the current date in DD- <i>MMM</i> -YYYY format
\$DATI	DATEI	Convert the current date in YYYY-MM-DD format
\$DATS	DATES	Convert the current date in system specific format. One of: DD- <i>MMM</i> -YY, DD- <i>MMM</i> -YYYY, or YYYY-MM-DD

---

**1.4.11.2 \$CDDMG Enhancements**

The original restriction for \$CDDMG, in which the largest value which could be displayed was 655,350,000 has been removed. \$CDDMG can now display a full 32-bit value.

---

**1.4.11.3 \$EDMSG Enhancements**

The \$EDMSG edit descriptor %Y has been modified to display the date in system specific format with parameters equivalent to \$DATS. Note that the length of the field will vary depending on the currently selected system date format.

## New Features, Corrections, and Restrictions

### 1.4.12 Enhancement to DUDRV to allow for large disks

MSCPDRV has been enhanced to allow for drives which are larger than 16 million blocks to be supported by the system. If a drive is greater than 16,777,215 blocks, the driver will automatically truncate the drive to the largest supported drive size.

### 1.4.13 Online reconfiguration enhancements

For RSX-11M-PLUS Version 4.6, the Host Reconfiguration functions have been modified to include additional System Attributes, and the output is displayed below:

```
>CON DISPLAY FULL ATTRIBUTES FOR SYS
SYS      SYS      Online,Accpath
                   PDP-11/94, EIS,UNIBUS_Map,D-Space,SWR,Cache,FPP,TOY
                   Clock=KW11-L, $TKPS=60., $TTPRM=000002, Cache_control=000001
```

Below is a summary of the available options which can now be altered using host reconfiguration:

#### *Formats*

```
CONfigure SET SYS [ CACHE=on/off/nnn
                   $TTPRM=nnnnnnn
                   $TKPS=value
                   NAME=sysnam      \ bold)
                   Y2
                   Y4
                   ISO ]
```

#### **Options**

##### **CACHE=on/off/nnn**

(Privileged keyword.) This option controls the system hardware cache option, if present, and support was included during sysgen. If the word "ON" is entered cache is enabled and the Cache Control Register (CCR) is set to the value of 000001<sub>8</sub>. If the word "OFF" is entered, cache is disabled and the CCR Register is set to a value of 000015<sub>8</sub>. Optionally, an actual value may be entered, the result of which will be processor specific. Consult your processor manual to correctly determine values for this word and their behaviour.

##### **\$TTPRM=nnnnnnn**

(Privileged keyword.) The terminal driver parameter word may be altered using the above format. The default value specified is taken as octal, unless terminated with a period ".", in which case it will be taken as decimal. Since only four bits are defined, the valid range for this option is 0 through 17<sub>8</sub>.

##### **\$TKPS=nnn**

(Privileged keyword.) This value is used to set the expected line frequency clock rate for the system. \$TKPS is the symbolic name for the executive location which represents the number of ticks per second on the system. For a system generated to operate on a line frequency of 50Hz, the value will be 50, and on a line frequency of 60Hz, the value will be 60. With this command, the system manager can change the expected line frequency for the system clock if the system is transported between countries with different line frequencies.

**Note:** If this parameter is altered to a value which does not correspond to the actual frequency of the system clock, all aspects relating to time will not be accurate.

### **NAME=sysnam**

(Privileged keyword.) This option controls the system name, and can only be set when DECnet is not active. If it is desired to change the system name, the name can be changed, and the system subsequently SAVed, to make the change permanent.

### **Y2**

(Privileged keyword.) This option controls the system date display and forces all system utilities to display dates as DD-*MMM*-YY.

### **Y4**

(Privileged keyword.) This option controls the system date display and forces all system utilities to display dates as DD-*MMM*-YYYY.

### **ISO**

(Privileged keyword.) This option controls the system date display and forces all system utilities to display dates as YYYY-MM-DD.

---

## 1.4.14 New Vectored Symbols

For this release, there are new vectored symbols added to the Executive. They can be found in [11,10]SFVC2.MAC and are listed below for convenience:

```
;
; SYMBOLS ADDED FOR VERSION 4.6
;
VECTOR $UTUIC
VECTOR $PATCH
```

---

## 1.4.15 Hardware Feature Mask Change

To accommodate the changes to the TOY clock support, the hardware feature mask has changed. In particular, the value HF.KDJ is now 2000 and not 1000. If you access this value, beware. The new feature mask layout is shown below:

```
;+
; SIXTH FEATURE MASK BITS
;-
F6.Y2K='B'^0100           ; DISPLAY ALL YEARS AS FOUR DIGITS
F6.ISO='B'^0400           ; DISPLAY ALL DATES IN ISO 8601 FORMAT
```

---

## 1.4.16 FCSRES rebuilt to use EIS Instructions

FCSRES has been built to use the EIS versions of various modules included within FCSRES. These modules reside in the library LB:[1,1]EISLIB.OLB. This library can be used when building tasks designed to run in an EIS only environment.

---

### 1.5 Software Problems Corrected in Version 4.6

This section describes software problems corrected in Version 4.6, other than those previously described associated with the year 2000.

---

#### 1.5.1 Corrections for Executive Routines

The following corrections have been made to executive routines

##### **Correct RDDRIV powerfail problem**

In Version 4.5, RDDRIV was vectored, and translation of executive vectors was done within the powerfail routine for RDDRIV. Once the powerfail routine had completed, it was checked to determine if the executive vectors had been translated, and if so, the subroutine was exited.

The problem exists because RDDRIV would fail to remove a register from the stack before returning to the previous caller, and would cause a system fault. This problem has been corrected.

##### **Full leap year algorithm has been included**

Previously, the system assumed that all years evenly divisible by four were leap years. This assumption is true between the years of 1901 through 2099, but is not the full leap year date algorithm.

The correct leap year date calculation of those years evenly divisible by four, except those divisible by one hundred, unless it is also divisible by four hundred, has been implemented.

##### **System Could Crash Shadowing virtual devices above a cached MSCP-disk**

When shadowing was started on a pair of virtual disks, connected to a cached MSCP disk, the system would crash. This has been rectified.

---

#### 1.5.2 HRC would use incorrect drive type on some MSCP devices

In some instances, HRC would insert an incorrect drive type on some MSCP type devices. For example, the device name ZIP00 would be incorrectly reported.

---

#### 1.5.3 KDJ11-E TOY clock correction

Previously, the MCR/DCL time command would incorrectly set the year after 1999. The problem was caused by writing an invalid BCD digit to the high order digit of the year stored in the TOY clock. This is now fixed.

In addition, the day-of-week register was not updated in the TOY clock.

For support beyond 2064, the MCR/DCL TIME command has also been modified to use the day of week register to determine the correct century.

---

### 1.5.4 HELLO would incorrectly calculate the day of week

The Day of week subroutine used within HELLO would not correctly determine the day of the week in some leap year situations. This problem has been fixed.

---

### 1.5.5 Corrections to BOOT Utility

Boot has been corrected to initialize both an MSCP, and ethernet controller prior to loading the new system image.

Previously, the ethernet controller could potentially corrupt system memory which the system image was being loaded.

---

### 1.5.6 INSTALL would allow non-privileged users to install ACPs

In some cases, INSTALL would allow a non-privileged user to install an ACP. This problem has been corrected.

---

## 1.6 Software Problems and Restrictions in Version 4.6

This section describes software problems and restrictions that have not been corrected for Version 4.6.

---

### 1.6.1 Restrictions in SYSGEN

This section describes restrictions in SYSGEN.

---

#### 1.6.1.1 Selecting FCPMDL in SYSGEN

During SYSGEN if both the nonstandard EXECUTIVE and FCPMDL are selected, the generated system will fail because FCPMDL is not present on the distribution and is not built during SYSGEN. This occurs because SYSGEN was modified to eliminate rebuilding of all vectored privileged tasks (except SAV) but FCPMDL is the only vectored privileged task not shipped on the baseline system. You can build FCPMDL by answering Y to the *BN010* question in the section on Building Nonprivileged Tasks and entering "FCPMDL" in response to the following question:

```
* BN020 Enter task name(s) [S]: FCPMDL
```

---

#### 1.6.1.2 Incorrect Displays for Multiple MM Devices in SYSGEN

The RSX-11M-PLUS SYSGEN RH configuration display lists only one MM device when there are actually two MM devices. ACF finds two TU77s (and a TM03 formatter) but SYSGEN prompts for the number of tapes drives (question CP1608) and formatters (CP1612). SYSGEN does configure correctly for two MM devices.

## New Features, Corrections, and Restrictions

---

### 1.6.1.3 Specifying a Nonexistent Device as the Crash Device During SYSGEN

If a nonexistent device is specified as the crash device during a SYSGEN operation, then you cannot change the crash device after the initial boot of the system. If the system reports an error, you must SYSGEN the system again and specify an existing device or the device mnemonic XX: as the crash device. XX: allows for changing the crash device.

---

## 1.6.2 VT Terminal Support

While the system will recognize the VT300, 400 and 500 series terminals with a SET TERM/INQ command, these terminals will be treated as if they are VT2xx series terminals. None of the additional features of these terminals is supported by RSX and its utilities, although this will not preclude individual applications determining the exact terminal type and using the additional features directly

---

## 1.6.3 Error Logging Restrictions

This section describes restrictions to Error Logging.

---

### 1.6.3.1 Support for Error Logging History Summaries

Error logging history summaries are no longer supported. See Section 3.8.

---

### 1.6.3.2 Incorrect Displays of Register Fields for DU and MU Devices

The errorlog driver internal timeout packet SA register fields for both DU and MU devices are not being displayed properly.

---

## 1.6.4 BRU Restrictions

This section describes restrictions to the Backup and Restore Utility (BRU).

---

### 1.6.4.1 Use of Wildcards in BRU

BRU supports the asterisk (\*) character only as a replacement of an entire field in a file specification. BRU will issue the "Invalid value or name" fatal syntax error when an asterisk is used to replace part of a file specification field.

---

### 1.6.4.2 Problem with the /NEW\_VERSION Qualifier

A problem with the /NEW\_VERSION qualifier occurred when used in a BRU command line during a disk-to-disk backup. From the input device, BRU backed up multiple versions of a file in the order that they appeared in a directory, instead of in ascending order by version number. As a consequence, when the files were restored to the output device, new version numbers were assigned when each file was restored, and the original order of the files was not preserved. Thus, the last version of a file on the output device may not have been the last version created on the input device.

The following workarounds are suggested to avoid this problem:

- Sorting files on the input device.

- Purging files prior to backing up your files.

---

### 1.6.5 DCL Command ABORT Taskname/TAS Does Not Translate Correctly

When an ABORT taskname/TAS command is entered, it translates to ABORT taskname/TERM=TAS, which is incorrect. It should be ABORT/TAS taskname.

The correct syntax for the DCL ABORT command requires that the /TASK qualifier follow immediately after the ABORT command. Furthermore, the MCR ABO command does not allow any of the following (you get an “invalid keyword” error message):

```
ABO/TAS task
ABO /TAS task
ABO task/TAS
ABO task /TAS
```

ABO task does work in MCR, which is identical to the action of the DCL ABORT command.

---

### 1.6.6 Tasks Hang When Attaching to A Terminal Already Attached to a Task

Tasks would occasionally hang when they attempted to attach a terminal that was already attached to another task. The attach QIO request never returned and the task had to be manually aborted.

As a workaround, the Marktime directive (MRKT\$) can be used to implement a timeout function within the application code. If the attach function is issued as a QIOW, you can precede it with a MRKT\$ directive, specifying the same event flag number using the I/O status block to distinguish between timeouts (IS.PND) and I/O completion (other values). Alternatively, the MRKT\$ directive can specify an AST routine to be entered at the expiration of the timer with a CMKT immediately following the QIOW. There are other workarounds such as using WTLO or WTSE with QIO and MRKT\$ directives. In these workarounds, the expiration of the time interval without I/O completion would signal the need for an I/O kill to be issued.

---

### 1.6.7 ELI/SH Reports No Errors if ELI/NOLIM Has Been Set

If error log limiting has been disabled with the ELI /NOLIM switch, then subsequent hard errors are not displayed by the ELI /SH switch even though a full report generated with RPT shows that device hard errors have been logged.

This restriction remains.

---

### 1.6.8 Problem with Unused Terminal Ports

The system can crash or suffer performance problems if there are unused terminal ports with cables connected to them. This can be eliminated by preferably removing the unused cables or setting the unused ports to a slaved state.

## New Features, Corrections, and Restrictions

---

### 1.6.9 Task Builder (TKB) Restriction

To obtain the long version of the memory allocation file (map) when building an I&D task, you should specify only the /-SH switch (the DCL /LONG qualifier) after the map file in the command line. The /MA switch (the DCL /MAP qualifier) should not be specified with the /-SH switch for I&D tasks.

---

### 1.6.10 RMD Restriction

If the RMD C (cache) page is being displayed interactively and the caching configuration is changed, RMD must be exited and a new RMD C command issued to display the new caching configuration correctly. RMD cannot refresh the display correctly for a caching configuration changed during an existing RMD display.

---

### 1.6.11 Indirect Processor Restrictions

---

#### 1.6.11.1 IPP handles .END statements in an ENABLE data block incorrectly

The Indirect Preprocessor (IPP) does not handle .END statements in an ENABLE data block correctly.

---

#### 1.6.11.2 IND handles .ONERR incorrectly with .BEGIN/.END blocks

Indirect incorrectly handles .ONERR statement processing when in a .BEGIN/.END block.

# 2

---

## Supplementary Software Information

This chapter provides supplementary information for the RSX-11M-PLUS Version 4.6 operating system and important information from earlier revisions of the release notes including additional information on the following topics:

- Removal of Disk Save and Compress
- Changes to File Control Services
- Restrictions and additional information on the Taskbuilder utility
- Changes to the GIN Directives
- Changes in HELLO and ACNT
- Changes in Shadow Recording

---

### 2.1 Disk Save and Compress Utility

RSX-11M-PLUS Version 4.3 was the last release to include the Disk Save and Compress utility (DSC). Any data currently saved using the DSC utility will have to be restored using the DSC from an earlier version and then resaved using BRU. Beginning with Version 4.4, Backup and Restore Utility (BRU) is to be used to provide the save and compress functionality.

---

### 2.2 File Control Services (FCS)

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.

## Supplementary Software Information

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> [1,1]SYSLIB/RP=[1,1]FCSNOVACP.OBS 
```

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.

---

### 2.3 Task Builder (TKB)

---

#### 2.3.1 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.**

---

#### 2.3.2 Hybrid Fast Task Builder (HFT)

This version of RSX-11M-PLUS provides an additional version of TKB, known as Hybrid Fast Task Builder (HFT). HFT uses a fast-mapped VSECT region for virtual memory table storage. This version of TKB may only be used on I&D space systems.

Currently, TKB uses a disk workfile for virtual memory table storage of symbols. For tasks with a large number of symbols, link time is greatly increased because of the amount of disk I/O required for swapping virtual memory pages in and out of dynamic memory (the free space between the end of TKB's code and its highest possible virtual address). Using a fast-mapped region will improve performance by eliminating this disk I/O. Performance improvements will vary depending on the task's configuration. As an example

## Supplementary Software Information

of possible improvements, test builds of PDP-11 C and PDP-11 BP2 done on a standalone Micro/R SX system using HFT build 35% to 43% faster than with TKB on the same system. Please note however, that for smaller tasks that do minimal amounts of paging, there will be little or no benefit.

HFT has been optimized to provide faster linking by using a fast-mapped VSECT region in place of the disk workfile used by TKB, TKBRES, and TKBFSL. Task images produced by HFT are identical to ones produced by TKB, TKBRES, and TKBFSL.

HFT can be installed with a default task name of ( . . . HFT). The user may also find it advantageous to install HFT as . . . TKB in place of the standard TKB task image; however, the task image is considerably larger. You can install one, the other, or both. It is recommended that you not install both tasks on the system, as the HFT requires a great deal of memory.

A task image for HFT built with the standard TKB defaults can be found in LB:[3,54]. If nonstandard defaults are desired, the build command file (HFTRESBLD.COMD) and .ODL file (HFTRESBLD.ODL) can be found in LB:[1,24], and HFT can be rebuilt using the section of SYSGEN which permits rebuilding of supplied system tasks (Building Nonprivileged Tasks). However, please note that during this section of sysgen, HFT will not automatically be rebuilt if you specify ALL, ALLRES, or ALLFSL due to the way HFT is configured. You must specify HFTRES when sysgen prompts for the task to be rebuilt in sysgen question BN020 in order to rebuild HFT. Rebuilding HFT is only possible on RSX-11M-PLUS, as the required files are not included with Micro/R SX.

All references to and use of the disk workfile have been removed in HFT. Initialization has been changed to immediately extend dynamic memory to D-APR 7 (instead of being extended 3100 bytes at a time on demand as TKB executes). The new HFT has been built with I&D, with D-APRs 0-2 mapped to D-space used by the task code and D-APRs 3-6 mapped to dynamic memory. D-APR 7 is used to fast map to the VSECT region. (Note that TKB, TKBRES, and TKBFSL also provide only 4 APRs worth of dynamic memory, so there is no reduction in dynamic memory with this implementation.)

The fast-mapped VSECT region will be the same size as the maximum disk workfile (64KW) and pages will be located in the VSECT region in a fashion similar to that used with the disk workfile. Virtual memory pages will be allocated sequentially, with each virtual memory page 256 words long. A page will be located by means of its page block number (the high byte of its address). When a page is read in or out of virtual memory, its page block number will be used to determine the following:

- Which APR in the VSECT region needs to be fast mapped
- The correct offset into the APR to locate the page

After fast mapping the appropriate APR, a subroutine copies the 256 words of the page into (or out of) the VSECT region to (or from) the designated location in virtual memory.

As a further optimization, FCSFSL is used as a resident library to avoid the overhead from context switching.

## Supplementary Software Information

Modified versions of the appropriate TKB modules to support the new HFT have been added to [31,10] and carry an *X* suffix (for example, there will now be an INIDMX used solely by the new HFT, in addition to the original INIDM used by TKB). The TKB assembly command file(s) have been updated.

---

## 2.4 GIN Directives

### 2.4.1 GI.DVJ –Son of Get Device Information

The Son of Get Device Information function returns information about the mount status and accessibility of a particular mountable mass storage device. The device on which information is returned is determined by first performing a logical assignment (if required) and then following any redirects. Device assignments are checked if the high bit in the flags byte is clear; otherwise, no check of device assignments is made.

#### Macro Call

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

#### Parameters

##### GI.DVJ

GIN\$ function code (18.)

##### buf

Address of the buffer to receive the unit information.

##### siz

Buffer size in words.

##### dev

Device name. (If zero, use task's TI.)

##### unt

Device unit number. (If high bit clear, follow assignments.)

**Table 2-1 GI.DVJ—Son of Get Device Information Subfunction Buffer Format**

Word 0	Device Mount Status
	If word 0 equals - 1, then the device is either not mountable or not a mass storage device.
	If word 0 equals - 2, then the device is not accessible to the issuing task.

**Table 2-1 (Cont.) GI.DVJ—Son of Get Device Information Subfunction Buffer Format**

Word 0	Device Mount Status
	<p>If word 0 is greater than zero, then the individual bits have the following meanings:</p> <ul style="list-style-type: none"> <li>Bit 0 – The issuer is not allowed read access to the device.</li> <li>Bit 1 – The issuer is not allowed write access to the device.</li> <li>Bit 2 – The issuer is not allowed extend access to the device.</li> <li>Bit 3 – The issuer is not allowed delete access to the device.</li> <li>Bit 4 – The device is public.</li> <li>Bit 5 – The device is private (allocated).</li> <li>Bit 6 – The device is mounted foreign.</li> <li>Bit 7 – The foreign mounted device has an ACP.</li> </ul>

**Macro Expansion**

```

GIN$   GI.DVJ, DVBUF, DVSIZ, "DU, 0
.BYTE  169.,18.
.WORD  GI.DVJ
.WORD  DVBUF
.WORD  DVSIZ
.WORD  "DU
.WORD  0
    
```

**Table 2-2 GI.DVJ – Son of Get Device Information Subfunction DSW Return Codes and Their Meaning**

Return Code	Meaning
IS.SUC	Successful completion.
IE.ADP	Part of 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 subfunction code or the DPB size is invalid.

**Note: 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 (the device unit number parameter).**

**2.4.2 GI.VEC – Translate Executive Entry Point Vector**

The translate executive entry point vector subfunction may be used by privileged tasks to provide independence from a particular system's symbol table file.

**Macro Call**

```
GIN$ GI.VEC,buf,siz
```

## Supplementary Software Information

### Parameters

#### GI.VEC

GIN\$ function code (15.)

#### buf

Address of the buffer that contains the offset codes and will contain the returned entry point addresses.

#### siz

Buffer size in words.

**Table 2-3 GI.VEC – Translate Executive Entry Point Vector Subfunction Buffer Format**

---

Word 0	Flag word  If word 0 equals 0, the translation is performed.  If word 0 is greater than zero, then the directive completes with status IS.SUC (success), but no translation is performed. This prevents successive translates on the same buffer from resulting in system faults. Word 0 is set to 1 when the translation is performed.
Word 1 – n Offsets	Each word beyond word 0 in the buffer is an offset into a table. Each of these offsets represents a symbol in the executive. The file LB:[3,54]RSXVEC.STB contains these symbols. The names of these symbols are identical to the represented entry point. For example, RSXVEC.STB might define \$TSKHD to a value of 1236. The real value of \$TSKHD for that system is at offset 1236 in the executive's vectored entry point table. For example, a value of 20646. The GI.VEC subfunction will replace the 1236 in the buffer with value of 20646. The GI.VEC subfunction will replace the 1236 in the buffer with the corresponding real symbol value, 20646.  If the word in the buffer does not represent a valid offset, or if it represents a symbol that is not included due to SYSGEN options on the target system, the value 160001 is returned for that value. No error is returned in this case. Applications that employ vectoring should test for the existence of optional features before using any entry points associated with those features.  For more information on vectoring, see the description of using vectored executive entry points in the <i>RSX-11M-PLUS and Micro/RSX Guide to Writing an I/O Driver</i> .

---

### Macro Expansion

```
GIN$    GI.VEC, EXEBUF, EXESIZ
.BYTE   169.,18.
.WORD   GI.VEC
.WORD   EXEBUF
.WORD   EXESIZ
```

**Table 2-4 GI.VEC – Translate Executive Entry Point Vector Subfunction DSW Return Codes and Their Meaning**

Return Code	Meaning
IS.SUC	Successful completion.
IE.ADP	Part of DPB is out of task's address space.
IE.SDP	Invalid subfunction code or the DPB size is invalid.

## 2.5 Enhanced security features within HELLO, and ACNT

The multi-user protection feature of RSX has been enhanced to provide increased security and user activity tracking. This enhanced security is selectable as a SYSGEN option, and is included with a standard function sysgen. Chapter 7 of this manual describes the enhancements and is intended as a supplement to the System Management Guide.

The following parameters have been added to the Account File Maintenance Utility (ACNT), and are enforced by the login task (HELLO) and set password facility (PSW):

- Password lifetime
- Password minimum length
- Inactivity logout timer
- Inactivity warning message initial time remaining
- Inactivity warning message time interval
- Password expiration
- Disable account
- Remote login disabled (from dialup or LAT terminals)
- Network login disabled (using DECnet-11M-PLUS V4.6)
- Account expiration

## 2.6 Shadow Load Balancing Support

Disk shadowing was previously enhanced to include disk load balancing, and maintenance enhancements, which include:

- Ability to balance I/O across both the primary and secondary devices
- Ability to limit the number of console messages produced when errors are detected on a shadow pair
- Ability to disable a device after multiple consecutive errors in order to be able to maintain overall system throughput

## Supplementary Software Information

- Ability to determine how far into the catchup operation a shadow set has progressed
- For MSCP devices, the ability to "MERGE" a pair of devices, and update the primary volume with the data from the secondary volume if a forced error mark is encountered on the primary volume
- Ability to dynamically alter the attributes of a shadow set, without dissolving, and recreating the shadow set.

The following sections document the changes to Shadow Recording:

### 2.6.1 Introduction

---

When a shadow pair is formed, and shadow load balancing is enabled, I/O requests queued to the primary device are evaluated to determine if better system throughput can be attained by queuing the request to the secondary device instead.

By default, load sharing is enabled for all shadow pairs created using the "START" or "MERGE" command, and with the exception of ACP I/O requests when the shadow pair is formed, using the "CONTINUE" option. This exception is due to the fact that any I/O performed to one volume prior to the formation of the shadow pair will have left the index file bitmap, and storage control bitmap in a slightly different state from the second volume, possibly corrupting the primary volume structure if the ACP received incorrect information from the secondary volume.

If there is any question about the integrity of the secondary volume, and load sharing is desired on the shadow pair, the shadow set should be formed using the "START" or "MERGE" commands only.

An additional integrity check has been included in the formation of a shadow pair when using the "CONTINUE" option. If the home block of the secondary volume does not match the home block of the primary volume, the shadow pair will not be formed, and an error will be reported indicating that the two volumes are not identical. Since the secondary volume is no longer used only in the event of an error, there is a higher probability that an inconsistent secondary volume would result in incorrect data being returned to a task issuing I/O requests to the shadow pair.

### 2.6.2 START/MERGE command

---

The START command has been enhanced to allow switches on the START command line. In addition, a MERGE function has been included which will allow forced error marks encountered on the primary volume to have the data re-written to remove that forced error mark. If the "MERGE" command is used, all blocks on the input volume will be copied to the secondary volume, and if a forced error mark is encountered on the primary, but not on the secondary volume, the data from the secondary volume will be written back to the primary volume. The format of the START/MERGE command is shown below:

#### Format

**SHA START/MERGE *ddnn: TO ddnn:[/switches]***

### Parameter

**ddnn: TO ddnn:**

**Privileged keyword.** Specifies the primary and secondary device to be used to form the shadow pair. The primary device must be mounted Files-11, and the secondary device must be mounted foreign.

### Switches

`/[NO]SHARE`  
`/[NO]ACP`  
`/[NO]LOG`  
`/[NO]ERROR`

#### **/SHARE**

The “/SHARE” switch enables load sharing between devices. If load sharing is selected during SYSGEN, this will be the default for all shadowing operations. The “/NOSHARE” switch should be used to disable load sharing if desired.

#### **/ACP**

The “/ACP” switch is used to force all ACP I/O requests to the primary device. This is the default for the CONTINUE command, but is disabled by default for the START/MERGE commands. If ACP requests should be shared between devices when using the CONTINUE command, the “/NOACP” switch should be used.

#### **/LOG**

The “/LOG” switch enables SHE . . . to report shadowing errors as they occur. The DISPLAY command has been updated to indicate how many errors have occurred on a given shadow pair, up to a maximum of 255. Once 255 errors have been displayed for a given pair, SHE . . . will no longer report errors which occur on that pair. If error reporting is not desired for a given shadow pair, the “/NOLOG” switch should be used.

#### **/ERROR**

The “/ERROR” switch is used to enable the system to disable a device after eight consecutive errors have occurred on the device. When a device is disabled, all read requests will only be serviced on the other device, however all write requests will still be forwarded to both devices. If a subsequent write request succeeds, the device will be enabled for use in load sharing. The default for this switch is always “/NOERROR” which indicates that the device cannot be disabled if multiple errors occur.

---

### 2.6.3 SET command

A new function, SET, has been included into the SHA task, which allows dynamic changing of attributes for a shadow pair, without the need to “STOP” and “START” the shadow pair again.

#### **Format**

**>SHA SET *ddnn:/switches***

## Supplementary Software Information

### Parameter

#### ddnn:

**Privileged keyword.** Specifies the primary device of a currently existing shadow pair to modify shadowing attributes.

### Switches

/[NO]SHARE  
/[NO]ACP  
/[NO]LOG  
/[NO]ERROR

Refer to the documentation for the START/MERGE command for a description of the functions of each switch.

---

## 2.6.4 DISPLAY command

The "DISPLAY" function has been updated to allow for more information to be displayed for each shadow pair. Below is shown the format of the DISPLAY command:

### Format

>SHA DISPLAY [/FUL]

### Parameter

#### /FUL

Specifies that all attributes for the shadow set should be shown. This only includes the following attributes:

ACP uses Primary  
Disable on Error

```
>SHA DIS Return
  UMB      Devices      I/O   Errors   Status
027130    DU0:, *DU7:      1.    0.      Load_Share, Merge, LBN=102640.
035550    *DU1:, (DU11:)    0.    10.     Load_Share, Catchup complete
041714    DR2:, DR3:       0.    *0.     Catchup complete
035570    DR0:, *DR1:      0.   >255.   Load_Share, Catchup complete
>
```

This example displays all of the shadowed devices in the system. The first column is the address of the UMB for the shadowing data structure.

The "\*" is used to indicate that the device will be selected next, if all other criteria are equal in the load balancing selection. If the shadow pair is not using load sharing, the "\*" will not be displayed.

The "(" for DU11: indicate that the device has been disabled, and is not included in the load sharing algorithm. However, all writes are still issued to both devices.

The "\*" or ">" characters in the errors column indicate that errors for those devices are no longer being reported via SHE . . . . The "\*" indicates that the reporting was disabled via the "SHA SET" command, while the ">" indicates that the error limit has been reached.

Under the Status column, both Catchup and Merge operations will indicate the current LBN which is being copied by the catchup task.

```
>SHA DIS /FUL Return
  UMB      Devices      I/O   Errors   Status
027130    DU0:, *DU7:      1.    0.    Load_Share, Merge, LBN=101984.
035550    *DU1:, DU11:      0.    0.    Load_Share, Catchup complete
                                ACP uses Primary
041714    DR2:, *DR3:      0.    0.    Load_Share, Catchup complete
                                ACP uses Primary, Disable on Error
035570    DR0:, *DR1:      0.    0.    Load_Share, Catchup complete
>
```

This example displays all information shown in the previous example, and also includes a second line for the additional information concerning the shadow pair.

---

## 2.7 Corrections for Universal Receiver Task

When the Universal Receiver task was announced in RSX-11M-PLUS Version 4.4, there were a number of problems identified. These include the following:

- Incomplete documentation
- INSTALL/REMOVE error
- Task activation sequence

Below is a revised description of the Universal Receiver task;

A Universal Receiver Task has been added to RSX. When installed, the Universal Receiver Task receives all SDAT type packets that would have received IE.INS. This task then routes the packets to the appropriate tasks and/or systems.

If a task is installed with the INS /URT switch, and with the name URT . . . , it is considered to be a Universal Receiver Task for all variants of the Send Data (SDAT) executive directive. SDAT\$, VSDA\$, SDRD\$, SDRP\$, VSRC\$, and VSUN\$ will not return the status IE.INS (specified task not installed), but instead the message will be routed to URT. It should be noted that a sequence of SDAT\$ followed by an RQST\$ will result in the data being delivered to URT . . . but the RQST\$ directive returning a status of IE.INS, indicating that the intended destination task was not installed.

To install and implement the Universal Receiver Task, the following command should be executed:

```
$ INSTALL/UNIVERSAL/TASK:URT... file_spec
>INS file_spec/URT=YES/TASK=URT...
```

If either the URT switch/qualifier, or the task name is not URT . . . , the task will not be considered the Universal Receiver Task.

In RSX-11M-PLUS V4.4, the vectored symbol \$URTCB was not included in the list of symbols to be translated by the GIN\$ directive. This caused the INSTALL and REMOVE utilities to reference the low memory address 003112 to be used as the address for \$URTCB. Usually, this value was zero, and a part of the crash dump buffer area. Since only INSTALL and REMOVE used this location, the Universal Receiver was never actually known to the system.

## Supplementary Software Information

If the issuing task issued an SDAT\$ directive, followed by a RQST\$ directive, the system would never activate the Universal Receiver Task, and data would be waiting in the tasks Receive Queue, but it would not be activated. This problem has been corrected in that the now the Universal Receiver task will be activated during the SDAT\$ process, to ensure that it will process the request in a timely manner.

When the Universal Receiver Task sends data to another task, it must use a Variable Send Data (VSDA\$), and the first two words of the buffer being sent should be the name of the task which any replies to the send data should be addressed. If a normal Send Data (SDAT\$) directive is used, the sending task name given for the receiver will be URT . . .

If the Universal Receiver Task attempts to send data to a task which is not installed, the packet will not be requeued to the Universal Receiver, but instead will result in the error task not installed (IE.INS).

---

### 2.8 SYSGEN Assumes no-FPP support

Previously, SYSGEN would always assume that no-FPP support was required in the system. If the floating point option was actually present, and Fortran-77 attempted to initialize on a system which did not include FPP support, the SFPA\$ directive would fail, and the task would get a task initialization failure, as shown below:

```
>RUN FPUTST
TTO -- Exiting due to ERROR 2
Task initialization failure
```

The only recovery for this problem was to re-SYSGEN the system.

SYSGEN has been modified to modify the default for FPP support on a system which supports I/D-space, and a standard function system is selected.

**Note: All DCJ11 systems support floating point instructions, regardless of whether the FPJ11 floating-point accelerator is installed. For DCF11 systems, floating-point support is optional, and requires either the KEF11-AA or FPF11 floating-point options.**

# 3

## 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.

### 3.1 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.

#### 3.1.1 Enhancements to the PRINT and SUBMIT Commands

---

The following was included in the RSX-11M-PLUS Version 4.4 Release Notes, and is included here as a supplement for the PRINT and SUBMIT commands documented in the RSX-11M-PLUS Batch and Queue Manual.

**Note: The description in the RSX-11M-PLUS Version 4.4 Release Notes contained an error regarding the use of DCL style qualifiers when defining a logical name to handle different forms types. The following section replaces that information.**

The PRINT and SUBMIT commands have been enhanced to have a mnemonic queue name translated to a processor name without requiring definition and assignment for the mnemonic queue name.

Previously, the Queue Manager restricted the system to a total of 16 print queues. By providing the capability to translate the PRINT command mnemonic queue name to a processor name, you can avoid the need to define and assign the second queue for each printer, thereby effectively doubling the capacity of the Queue Manager. The device-specific queue must still be created. As a side benefit, you also can define a logical name (for instance, in your Session Table) and avoid the need to remember which /FORM=*n* to specify when printing to a terminal-like device. Previously, you might have issued a command similar to the following to print 80 columns, letter quality on an LN03:

```
PRINT/FORM=4 resume.txt
```

Now, you can define a queue similar to the following and then issue the next command:

```
DEF QMG$QUE_NICE "TT20:/FO:4"  
PRINT/QUE=NICE resume.txt
```

## Corrections to Documentation

Rather than requiring the system manager define a mnemonic queue and assign it to the processor-related queue, the ASSIGN/QUE command will now define a logical name in the System Table to describe the relationship, if the queue being assigned has not previously been created. Subsequent use of the PRINT/QUE=name command attempts to translate the queue name to a processor name, that is then substituted for the original queue name. The translation repeats until a failure occurs, at which time the last translated name is submitted to the Queue Manager.

**Note:** The format for the destination queue in a logical name must be in MCR format.

---

### 3.1.1.1 ASSIGN/QUEUE (QUE /AS: in MCR) Command

DCL

```
$ ASSIGN/QUEUE que_name proc_name[/qualifiers]
```

MCR

```
>QUE proc_name:[/qualifiers]/AS:que_name
```

#### Description

All command qualifiers and MCR Job Switches valid for the PRINT command may be used where appropriate.

**Note:** If your system does not have logical name support installed, QMG cannot use logical names.

If que\_name has previously been defined to the Queue Manager, the command exhibits prior behavior; for example, a path is established from the queue to the processor. In this case, any qualifiers are ignored and a warning message is issued to the user. Note that queue names PRINT and BATCH are always defined in the default system.

If que\_name has not been defined to the Queue Manager, a special logical name is defined in the System Table to represent the translation from que\_name to proc\_name, including qualifiers. This is equivalent to issuing the following DCL command:

```
DEF QMG$QUE_que_name "proc_name[/qualifiers]"
```

---

### 3.1.1.2 PRINT (PRI in MCR) Command

DCL

```
$ PRINT/QUEUE:que_name [/qualifiers] filespec[/filequalifiers]
```

MCR

```
> PRI que_name:[jobname][/jobswitch]=filespec[/filequalifiers]
```

All command qualifiers and MCR Job Switches valid for the PRINT command may be used where appropriate.

**Note:** Note, however, that any qualifiers that were included in the ASSIGN/QUEUE (or QUE /AS:) command must not be repeated here. The resulting multiple qualifiers after que\_name translation will cause an error return from the Queue Manager.

Execution of the PRINT command first iteratively translates que\_name. At each step, the original que\_name is replaced by the translation, including qualifiers (if any), and the new que\_name is extracted and submitted for further translation. When there are no more translations, the last constructed string is submitted to the Queue Manager for execution.

---

### 3.1.1.3 DEASSIGN/QUEUE (QUE /DEA in MCR) Command

**DCL**

**\$ DEASSIGN/QUEUE que\_name proc\_name**

**MCR**

**> QUE proc\_name:/DEA:que\_name**

If que\_name has previously been defined to the Queue Manager, the command exhibits prior behavior; for example, the path from the queue to the processor is severed.

If que\_name has not been defined to the Queue Manager, the special logical name representing the que\_name to proc\_name translation is deleted from the System Table. This is equivalent to issuing the following DCL command:

**DEASS/SYSTEM QMG\$QUE\_que\_name**

The following commands are unchanged from their prior behavior:

INITIALIZE/QUEUE	(QUE /CR)
DELETE/QUEUE	(QUE /DEL)
STOP/QUEUE	(QUE /STO)
START/QUEUE	(QUE /STA)

**Note:** Note, however, that they will affect only queues that *exist* in the Queue Manager because they were created by the INITIALIZE/QUEUE command (or by default in the case of PRINT and BATCH queues); they will not affect mnemonic queues that are mapped to processors by means of logical names.

---

## 3.2 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 and later 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 and later series terminals in VT200 mode only. You cannot specify VT3xx\_SERIES as a terminal type, nor VT4xx or VT5xx.
- 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:

## Corrections to Documentation

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 multiheader index files). You can use Table 3-1 to determine an appropriate value to use for the `/HEADERS:n` parameter.

**Table 3-1 Number of Index File Headers**

Maximum Number of Files	Disk Size	Number of File Headers
Maximum files fewer than 25588	Disks fewer than 209 Mb	Headers=maximum/2
Maximum files greater than 25588 and fewer than 51176	Disks between 210Mb and 419Mb	Headers=25593
Maximum files greater than 51176 and fewer than 51693	Disks between 420Mb and 423Mb	Headers=25846
Maximum files greater than 51693 and fewer 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 3-2 as the default approximations:

**Table 3-2 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:

$$\text{MAXIMUM FILES} = ((N - ((N + 4095.) / 4096.) + 9.) * 127.) / 258.$$

- Please add the following information to Section 5.11.1, page 5-80 at the end of the first paragraph:  
You must specify the `/WRITE:n` qualifier when formatting volumes on DL-type devices.
- 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` command qualifiers `/CREATED` and `/MODIFIED` is incorrect. The correct formats are:

## Formats

**BACKUP/CREATED/arg**

**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.

```
$ BACKUP/MODIFIED/AFTER: (14-MAY-87 17:00)/VERIFY DU0: MS0: Return
```

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.

- Please add the following text to Section 5.13.1 of the manual:

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

## Formats

**\$ SHOW ASSIGNMENTS** [*logicalname*]/[*qualifier*]

**\$ 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[:n]  
/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.

## Corrections to Documentation

### Examples

```
$ SHOW ASSIGNMENTS 
```

```
(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.

```
$ SHOW ASSIGNMENTS S* 
```

```
(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.

```
$ SHOW LOGICALS "SYS$CLI" 
```

```
(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.

```
$ SHOW ASSIGNMENTS "SYS$CLI" 
```

```
$
```

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

```
$ SHOW ASSIGNMENTS /ALL 
```

```
(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.

```
$ SHOW LOGICALS S*/ALL Return
```

(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.
- Section 8.1.3, page 8-6, states that the LOW and HIGH parameters are the low and high pool limits in bytes respectively. Please add the following text:

On systems with ICB pool, the low and high limit values may be specified as "ICB". This will cause the low or high limit threshold to be traversed when ICB pool is accessed due to a shortage of primary pool.

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

The behavior of the BRU qualifier /ERROR with restore operations is unchanged. However, the qualifier can 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 it 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.

## Corrections to Documentation

- 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

---

### 3.3 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:

```
>@ti: 
AT.>.enable substitution 
AT.>.sets a "1,2," 
AT.>.parse a "," b c d 
AT.>.'b' 
>;1
AT.>.'c' 
>;2
AT.>.'d' 
>;
AT.>.parse a "," b c  (null substring)
AT.>.'b' 
>;1
AT.>.'c' 
>;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.

---

### 3.4 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.

## Corrections to Documentation

```
>ASN 
(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"

>
```

This example displays all the login and local assignments for the issuing terminal.

```
>ASN S* 
(Session Login Logical Table for VT2:)
"SYS$LOGIN" = "DU:[USER]" [Final]
"SYS$PROMPT" = "Jim>"

(Session Local Logical Table for VT2:)
"SYS$CLI"  = "DCL"

>
```

This example displays the login and local assignments that begin with letter S for the issuing terminal.

```
>ASN "SYS$CLI" 
>
```

This example displays no matching login and local assignments that match the string "SYS\$CLI" for the issuing terminal.

```
>ASN /ALL 
(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:)
```

```
"SYSSCLI" = "DCL"  
>
```

This example displays all the system, group, login, and local assignments for the issuing terminal.

- In Section 3.13 of the manual, immediately after the second paragraph of step 3, please add the following text:

Much of the information the system displays at logout can be suppressed with the switch /S[ILENT].

Please change the format of the command to the following:

```
BYE [ / [-]H ] [ /S ]
```

Please change the error message text as follows:

A keyword other than a form of /H[OLD],/-H[OLD],/NOH[OLD] or /S[ILENT] was specified in the command line.

- 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.

```
>DFL "SYSSCLI"   
(Session Local Logical Table for VT2:)  
"SYSSCLI" = "DCL"  
>
```

## Corrections to Documentation

This example displays the login and local assignments that match the string SYS\$CLI for the issuing terminal.

```
>DFL S*/ALL Return
(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"
>
```

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

- Please add the following information to Section 3.26 on page 3-67:

The HELLO task 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:
```

The overlay structure of the HELLO 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.

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.

- Please add the following information to Section 3.27:

The overlay structure of the HELLO 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.

- 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 3-3 to determine an appropriate value to use for the /INF parameter.

**Table 3-3 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
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 3-4 as the default approximations:

**Table 3-4 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)]**

## Corrections to Documentation

### 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**

**/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.

```
>OPE 0/REG=TSTREG   
00000000/50712
```

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

```
>OPE 0/REG=TSTREG   
00000000/50712 %  MCR
```

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

```
>OPE 0/REG=TSTREG   
00000000/50712 %  MCR %DCL   
00000002/00000   
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.

## Corrections to Documentation

- Please add the following information to Table 3-4 in Section 3-42:

**Table 3-5 SET Command Keyword Summary**

<b>Keyword</b>	<b>Description</b>
<b>Ensuring System Protection</b>	
<code>/INTRUSION:yes</code>	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.

Resource Accounting has been expanded to include a new qualifier to the SET command. 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`

- 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.

**Formats**

`SET /CRASHDEV=ddnn:[CSRaddr]`

`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, if crash dump support is included, and the crash device selected during sysgen was "XX."

- In Section 3.42, page 3-161, include the following new SET command.

**Formats**

`SET /CKP[=option]`

### option

(Privileged keyword.) The optional item allows for either the RSX-style of checkpointing, where the system always starts at the bottom of the partition when selecting a region to be checkpointed, or the P/OS-style of checkpointing which will start from the previously selected region to select a region to be selected.

### RSX

The RSX checkpointing algorithm allows the system to start at the base of the partition which is attempting make space available, and work towards the top. This can occasionally cause problems if an application selected for checkpointing has multiple I/O requests pending, such that its I/O count never goes to zero. With this algorithm, the same application will be repeatedly selected for checkpointing, and so the target task will not be loaded unless other applications terminate and eliminate the need for checkpointing to permit the load.

### POS

The P/OS checkpointing algorithm will start, not at the base of the partition, but rather, after the last sub-partition attempted for checkpointing on the last pass. This allows an application described under the RSX style to be skipped, and permit another task to be selected for checkpointing. When the top of the partition is reached, the algorithm will again start at the bottom.

- Please add the following information to Section 3.42, page 3-162:

#### Format

##### **SET /CTRLC=ttnn:**

This qualifier establishes whether a CTRL/C causes an abort (/CTRLC=ttnn:) or causes an explicit MCR> prompt (/NOCTRLC=ttnn:).

- 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, on the section describing the command SET /PLCTL for the parameters "low" and "high", please add the following:

The low and high pool limits may also be set to the value "ICB" for systems in which ICB pool is enabled. This value will set the low or high limit threshold to violated when a pool allocation occurs from ICB pool.

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

## Corrections to Documentation

The MCR command SET/INQUIRE supports the VT300 and later 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 and later series terminals in VT200 mode only. You cannot specify VT3xx, VT4xx, or VT5xx 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.

```
>SET /SECPOL Return  
SECPOL=285.:640.:44%
```

Displays the amount of available secondary pool.

- In Section 3.47, page 3–206, the format for the time command has been changed to include two new switches. Below is the correct format for the time command:

### Formats

**TIM[E] [/SYNC]**

**TIM[E] [hrs:mins[:secs]] [m1/day/year] [/SETTOY]**

**TIM[E] [hrs:mins[:secs]] [day-m2-year] [/SETTOY]**

### Switches

#### **/SYNC**

(Privileged keyword.) The option allows for the system time to be set based on the current time from the systems TOY clock. This option is only applicable for KDJ11–E systems, or MENTEC M–series processors which have the TOY clock option installed.

#### **/SETTOY**

(Privileged keyword.) The option allows for the system time to be set, and update the time kept on the TOY clock for KDJ11–E systems, or MENTEC M–series processors which have the TOY clock installed.

- 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.

---

## 3.5 RSX–11M–PLUS System Generation Guide and Installation Guide

Please add the following information to the text in Section 3.2.2, Choosing Executive Options (CE), on page 3–26. This text should be added after the question \* *CE200 Which FCP do you want? [S R:1-6 D:“FCPLRG”]*: and just before the \* *CE210: Do you want support for file windows in secondary pool? [Y/N D:N]*: question:

If you entered FCPMDL, then you must answer Y to question BN010 in the section on Creating the System Image File and enter FCPMDL to question *BN020 Enter task name(s) [S]: FCPMDL*. Both of these questions appear on page 3–64 of the *RSX–11M–PLUS System Generation Guide and Installation Guide*.

## Corrections to Documentation

Section 5.2 on page 5-2 of the *RSX-11M-PLUS System Generation Guide and Installation Guide* is incomplete. Since the RL02 distribution is no longer provided on RL02 media, the tape backup set must be restored to an RL02 prior to the description given in Section 5.2. The following example assumes that the tape drive being used is MU0:. If another type of tape drive is being used, replace MU0: with the appropriate device name.

If you are restoring the RL02 distribution from an online RSX-11M-PLUS environment, use the following commands:

```
>MOU MU0:/FOR 
>MOU DL0:/FOR 
>BRU /REWIND/INIT/VERIFY/BAC:RSXMPRL02 MU0: DL0: 
BRU - Starting Tape 1 on MU0:
BRU - End of Tape 1 on MU0:
BRU - Starting verify pass Tape 1 on MU0:
BRU - End of Tape 1 on MU0:
BRU - Completed
>
```

If you are restoring the tape backup set using from a standalone environment, refer to the description in Section 2.4.1.2 starting on page 2-8, and replace the example shown in step 10 with the following BRU example:

```
>RUN BRU 
>
BRU>/INIT/VERIFY/BAC:RSXMPRL02 
From: MU0: 
To: DL0: 
BRU - Starting Tape 1 on MU0:
BRU - End of Tape 1 on MU0:
BRU - Starting verify pass Tape 1 on MU0:
BRU - End of Tape 1 on MU0:
BRU - Completed
BRU> 
```

After completing step 10, return to Section 5.2 to complete the installation of the Pregenerated system.

The paragraph in Section 5.4.7 on page 5-43 is incorrect. Please replace it with the following paragraph:

RMS-11 Version 2.0 is included on the pregenerated kit disk. The RMS-11 segmented library (RMSRES, and RMSLBL through RMSLBM) and all the RMS-11 utilities are already installed in the system image. No further installation is needed, unless you install the DECnet package on your system and you want to use the RMS-11 remote access facilities. See Section 5.4.1 in the *RSX-11M-PLUS System Generation Guide and Installation Guide* for information on installing the RMS-11 remote access package (DAPRES).

### 3.6 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; 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 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.

#### Formats

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

**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.

---

### 3.7 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 Section 1.17.1.

---

### 3.8 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

---

### 3.9 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.

## Corrections to Documentation

- 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.
- Please add the following descriptions of the new high-level language calls to the GIN directives to Chapter 5 of the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual*. The following GIN directives may be called:
  - CALL GINABO
  - CALL GINAPR
  - CALL GINDEF
  - CALL GINDEV
  - CALL GINDVJ
  - CALL GINFMK
  - CALL GINGAS
  - CALL GINQMC
  - CALL GINREN
  - CALL GINSPR
  - CALL GINTSK
  - CALL GINUAB
  - CALL GINUIC
  - CALL GINUPD
  - CALL GINVEC

---

### 3.9.1 The GINABO Call

**CALL GINABO** ( *buf* , *siz* [*ids*] )

**buf**

A buffer area containing the name of the device for the aborts.

**siz**

An integer containing the size of the buffer in words. If the buffer is declared as an INT\*2 array, then the size is equal to the number of elements in the array.

**ids**

An integer to receive the directive status.

See the description of the GI.ABO subfunction of the GIN directive in the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

---

### 3.9.2 The GINAPR Call

**CALL GINAPR** ( *buf* , *siz* [*ids*] )

**buf**

A buffer area for the return of the information.

**siz**

An integer containing the size of the buffer in words. If the buffer is declared as an INT\*2 array, then the size is equal to the number of elements in the array.

**ids**

An integer to receive the directive status.

See the description of the GI.APR subfunction of the GIN directive in the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

---

### 3.9.3 The GINDEF Call

**CALL GINDEF** ( *uic* [*ids*] )

**uic**

An integer representing an user identification code (UIC).

**ids**

An integer to receive the directive status.

See the description of the GI.DEF subfunction of the GIN directive in the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

---

### 3.9.4 The GINDEV Call

**CALL GINDEV** ( *buf* , *siz* , *devnam* , *unit* [*ids*] )

**buf**

A buffer area for the return of the information.

**siz**

An integer containing the size of the buffer in words. If the buffer is declared as an INT\*2 array, then the size is equal to the number of elements in the array.

**devnam**

A two-character ASCII device mnemonic.

**unit**

An integer containing the device unit number.

**ids**

An integer to receive the directive status.

## Corrections to Documentation

See the description of the GI.DEV subfunction of the GIN directive in the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

---

### 3.9.5 The GINDVJ Call

**CALL GINDVJ** ( *buf* , *siz* , *devnam* , *unit* [,*ids*] )

**buf**

A buffer area for the return of the information.

**siz**

An integer containing the size of the buffer in words. If the buffer is declared as an INT\*2 array, then the size is equal to the number of elements in the array.

**devnam**

A two-character ASCII device mnemonic.

**unit**

An integer containing the device unit number.

**ids**

An integer to receive the directive status.

See the description of the GI.DVJ subfunction of the GIN directive in Section 2.4.1 and also, later in this section in this document, for details on the buffer format and use of this directive.

---

### 3.9.6 The GINFMK Call

**CALL GINFMK** ( *buf* , *siz* [,*ids*] )

**buf**

A buffer area for the return of the information.

**siz**

An integer containing the size of the buffer in words. If the buffer is declared as an INT\*2 array, then the size is equal to the number of elements in the array.

**ids**

An integer to receive the directive status.

See the description of the GI.FMK subfunction of the GIN directive in the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

---

### 3.9.7 The GINGAS Call

**CALL GINGAS** ( *buf* , *siz* , *devnam* , *unit* , *udev* , *unum* [,*ids*] )

**buf**

A buffer area for the return of the information.

**siz**

An integer containing the size of the buffer in words. If the buffer is declared as an INT\*2 array, then the size is equal to the number of elements in the array.

**devnam**

A two-character ASCII device mnemonic.

**unit**

An integer containing the device unit number.

**udev**

**unum**

A device name and unit number pair, as described for devnam and unit. This device name is used to identify a terminal for the interpretation of logical names.

**ids**

An integer to receive the directive status.

See the description of the GI.GAS subfunction of the GIN directive in the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

---

**3.9.8 The GINQMC Call**

**CALL GINQMC** ( buf , siz [,ids] )

**buf**

A buffer area containing the command line.

**siz**

An integer containing the size of the buffer in words. If the buffer is declared as an INT\*2 array, then the size is equal to the number of elements in the array.

**ids**

An integer to receive the directive status.

See the description of the GI.QMC subfunction of the GIN directive in the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

---

**3.9.9 The GINREN Call**

**CALL GINREN** ( nam1 , nam2 [,ids] )

## Corrections to Documentation

**nam1**

**nam2**

Two integers that represent, respectively, the first and second half of a task name expressed in RAD50.

**ids**

An integer to receive the directive status.

See the description of the GI.REN subfunction of the GIN directive in the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

---

### 3.9.10 The GINSPR Call

**CALL GINSPR** ( *flag* [, *ids*] )

**flag**

An integer that is interpreted as true if not zero; false if zero.

**ids**

An integer to receive the directive status.

See the description of the GI.SPR subfunction of the GIN directive in the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

---

### 3.9.11 The GINTSK Call

**CALL GINTSK** ( *buf* , *siz* , *nam1* , *nam2* [, *ids*] )

**buf**

A buffer area for the return of the information.

**siz**

An integer containing the size of the buffer in words. If the buffer is declared as an INT\*2 array, then the size is equal to the number of elements in the array.

**nam1**

**nam2**

Two integers that represent, respectively, the first and second half of a task name expressed in RAD50.

**ids**

An integer to receive the directive status.

See the description of the GI.TSK subfunction of the GIN directive in the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

---

**3.9.12 The GINUAB Call**

**CALL GINUAB** ( *buf* , *siz* , *devnam* , *unit* [*ids*] )

**buf**

A buffer area for the return of the information.

**siz**

An integer containing the size of the buffer in words. If the buffer is declared as an INT\*2 array, then the size is equal to the number of elements in the array.

**devnam**

A two-character ASCII device mnemonic.

**unit**

An integer containing the device unit number.

**ids**

An integer to receive the directive status.

See the description of the GI.UAB subfunction of the GIN directive in the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

---

**3.9.13 The GINUIC Call**

**CALL GINUIC** ( *buf* , *siz* [*ids*] )

**buf**

A buffer area for the return of the information.

**siz**

An integer containing the size of the buffer in words. If the buffer is declared as an INT\*2 array, then the size is equal to the number of elements in the array.

**ids**

An integer to receive the directive status.

See the description of the GI.UIC subfunction of the GIN directive in the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

---

**3.9.14 The GINUPD Call**

**CALL GINUPD** ( *buf* , *siz* [*ids*] )

**buf**

A buffer area for the return of the information.

**siz**

An integer containing the size of the buffer in words. If the buffer is declared as an INT\*2 array, then the size is equal to the number of elements in the array.

## Corrections to Documentation

### ids

An integer to receive the directive status.

See the description of the GI.UPD subfunction of the GIN directive in the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

---

### 3.9.15 The GINVEC Call

**CALL GINVEC** ( *buf* , *siz* [*ids*] )

#### buf

A buffer area for the return of the information.

#### siz

An integer containing the size of the buffer in words. If the buffer is declared as an INT\*2 array, then the size is equal to the number of elements in the array.

#### ids

An integer to receive the directive status.

See the description of the GI.VEC subfunction of the GIN directive in Section 2.4.2 in this document for details on the buffer format and use of this directive.

---

### 3.9.16 Other Corrections to the RSX-11M-PLUS and Micro/RSX Executive Reference Manual

This section contains additional corrections to the *RSX-11M-PLUS and Micro/RSX Executive Reference Manual*.

- 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      ;CMKT$ 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 canceled and the DPB size equals 1.

- In Section 5.18, page 5-53, please add the following text to the description of the CRRG\$ directive:

A 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 current 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.

Additionally, if RS.TOP is included as a logical or with RS.RES, the region will be created in memory in a top-down manner, allowing a region which is expected to be permanent, or semi-permanent to be created in a top-down allocation in the partition specified.

- In Section 5.35, page 5-91, please delete the second bullet in the list. Add the following text to the description of the EXTK\$ directive:

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.

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 occur only 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 that 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.

- 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:

## Corrections to Documentation

**Table 5-1: System Feature Symbols**

Symbol	Value	Meaning
FE\$SEC	48.	System supports extended security
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
FE\$SLS	81.	System supports shadow load sharing
FE\$UBM	82.	System supports UNIBUS memory
FE\$ICP	83.	System supports using ICB pool as backup for primary pool
FE\$TOY	-9.	System has TOY clock (bit 0)
FE\$TY1	-10.	System has TOY clock (bit 1)
FE\$KDJ	-11.	System is KDJ-11 processor
FE\$UME	-12.	System has UNIBUS memory
FE\$BMV	-13.	System supports block-move instruction

- 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)

```

## Corrections to Documentation

```
.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:

$$n = n + 120000 + (\text{base} \& 177700)$$

- 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
T4\$LBW	42.	Task waiting for local buffer
T4\$LRW	43.	Task waiting for local ring
T4\$RON	44.	Priv task mapped read-only to executive

- 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.

## Corrections to Documentation

### 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 see the additional text for Section 5.35 explaining enhancements to the EXTK\$ directive. This text appears earlier in this section.) 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.
- 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.

## Corrections to Documentation

### 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 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.

```
>TKB task=object
```

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.

### New SDUN\$ and VSUN\$ Directives

Two new directives have been added to the system. These are Send Data and UNstop (SDUN\$), and Variable Send and UNstop (VSUN\$). Additionally, two new high-level language interfaces (SDUN and VSUN) have been added. These directives, and their functional descriptions, are identical to SDAT/SEND and VSRC, with two exceptions:

- The DIC for this directive is 179.
- The directives may return a status of IS.CLR(0) or IS.SET (+2). A status of IS.SET indicates the send was successful but the target task was not active. A status of IS.CLR indicates the send was successful and the target task was active but not stopped.

### 3.10 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 callable routine 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 2.4 of the manual:

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

- Please add the following information to Section 4.5.12 of the manual:

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 that 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 multivolume 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.

- 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, pages 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

```

### 3.11 RSX-11M-PLUS and Micro/R SX I/O Drivers Reference Manual

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

The format shown in Section 2.4.3 on page 2–23 is incorrect. The correct format is:

```

QIO$C IO.ATA [ !TF.ESQ
              !TF.NOT ],lun,[efn],[pri],[isb],.<[ast1],[parameter2],[ast2]>
              !TF.XCC ]

```

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/R SX I/O Drivers Reference*

## Corrections to Documentation

*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

Bit	Setting	Meaning
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
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 <sup>1</sup>
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)

<sup>1</sup>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).

## Corrections to Documentation

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

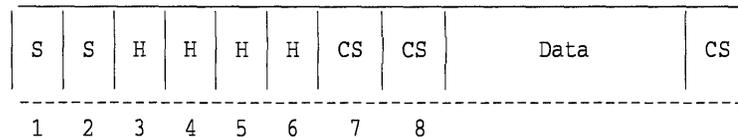
IO.INL initializes a physical device unit for use as a communications link. It turns the device on line, sets device characteristics, and ensures that the appropriate data terminal is ready.

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

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.

## Corrections to Documentation

### 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.

## Corrections to Documentation

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$$,QIO$$
.
.
.
ALUN$$  1,"XP,0                                ;USE LUN1 FOR DP11
QIO$$   IO.HDX!IO.SYN,<1,,,,,226>             ;SET DEVICE PARAMETERS
QIO$$   IO.INL,1                               ;PUT DEVICE ON LINE
QIO$$   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.

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.

## Corrections to Documentation

- 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 UCBD.F.MAC module to determine if error packets should be logged during bad block replacement by RCT.

### 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.

### 3.12 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.
	%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.

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.

### 3.13 RSX-11M-PLUS and Micro/RSX System Management Guide

- Please add the following information to Section 1.3.3:

The print job attributes "Print adjacent to prior job" and "Print job should be held" are included in [1,20]QMGBLD.BLD as valid options for \$JATDF. This global symbol determines the selection of default qualifiers for print jobs that 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.

## Corrections to Documentation

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.

- In Section 5.5.16, page 5-74, the third example is incorrect. The correct example is as follows:

```
VMR>SET /SECPOL Return
SECPOL=285.:640.:44%
```

Displays the amount of available secondary pool.

- Please add the following to Section 7.9 and replace Example 7-6 with the following example:

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

```
>rmd c
RSX-11M-PLUS V4.4 BL78      Cache Statistics (General)      17-JAN-93
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%
```

- Please add the following to Section 7.10 and replace Example 7-7 with the following example:

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

```
>rmd d
RSX-11M-PLUS V4.4 BL78      Cache Statistics (Detailed)      17-JAN-93
12:57:04

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

Reads      Virtual  Readahead  Directory  Logical  Overlay  Total
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):
Primary Pool Allocation 0%      Deferred Write Rate 0%
Cache Pool Allocation 0%       Write Load Rate 0%
Read Load 0%
```

- Please add the following information to Section 10.2.1:

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

## Corrections to Documentation

/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

- 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:

```
*** ddn: -- Replace command failure
```

If you receive this message, it is recommended 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) ADD and MODIFY commands behave differently. They 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, it is recommended 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.

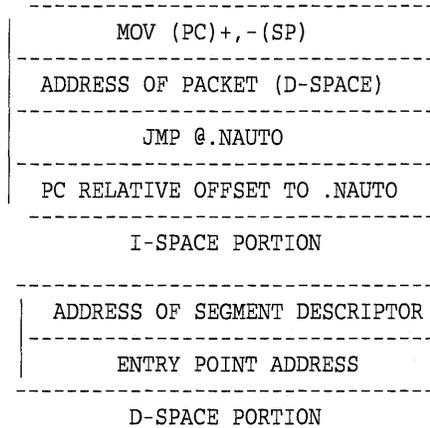
---

### 3.14 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:

## Corrections to Documentation



- Change all references to FORTRAN IV-PLUS in Chapter 5 to FORTRAN-77. FORTRAN IV-PLUS is no longer supported.
- The example on page 4-16 in Section 4.5 is incorrect. Please replace it with the following correct example. (Note the corrected positioning of the period (.) in first line of the example.)

```
MOV @#N.OVPT,R0  
BISB #200,N.FAST(R0)
```

- 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>VSECT,VSECT/-SP=VSECT, LB:[1,1]F77FCS/LB   
TKB>/   
Enter Options:  
TKB>WNDWS=1   
TKB>VSECT=MARRAY:160000:20000:200   
TKB>//   
>
```

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

```
$ LINK/TAS/MAP:VSECT/NOPRINT/OPT VSECT, LB:[1,1]F77FCS/LIB   
Option? WNDWS=1   
Option? VSECT=MARRAY:160000:20000:200   
Option?   
$
```

- Replace FOROTS.OLB with F77FCS.OLB in the second paragraph under the command sequence in Section 5.4.2.
- Replace the Note in Section 7.8.1 with the following text:

Prior to Version 4.3, 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:

## Corrections to Documentation

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

Support has been added for manually-loaded overlaid libraries.

- In Section 8.6.1, page 8–8, the examples are incorrect. Please replace them with the following correct examples:

```
TKB>CSM/-HD/LI/PI,CMS/MA, CMS= 
TKB>LB:[1,1]SYSLIB/INCLUDE:COMPAL,SY:[301,55]CSM
TKB>/ 
Enter Options:
TKB>STACK=0 
TKB>PAR=GEN:0:2000 
TKB>CMPRT=$CMPCS 
TKB>BLGXCL=$SAVAL 
TKB>// 
>
```

Or, you use the following LINK command sequence to build the same library:

```
TKB>LINK/TAS:CSM/NOH/SHARE:LIB/CODE:PIC/MAP:CSM/SYS/SYM:CSM/OPT -

->LB:[1,1]SYSLIB/INCLUDE:COMPAL,SY:[301,55]CMS 
Option?STACK=0 
Option?PAR=GEN:0:2000 
Option?CMPRT=$CMPCS 
Option?GBLXCL=$SAVAL 
Option? 
$
```

Also, add the following sentence to the end of page 8–8:

Note that the SYSLIB module COMPAL contains both the \$CMPCS and \$COMPAL routines.

- 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:

```
$ LINK/TAS/MAP/SYM INLIB1/INCLUDE:(MOD1,MOD2),- 
->inputfile2,INLIB1/LIBRARY 
```

- In Section 11.39, page 11–55, the description of the /SAVE qualifier is incorrect. All references to the file ATLINK.COMD 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.

## Corrections to Documentation

### 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 12.25, page 12–34 of the manual and to Section 12.29, 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*]

**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 12.34, 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 12.5 on page 12–10, Section 12.7 on page 12–13, and Section 12.21 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 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 4.2 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]]**

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

**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 an 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.

## Corrections to Documentation

- In Appendix B, Figure B-5: Table Block 0-Task and Resident Library Data is incorrect. The correct figure is shown next.
- In Appendix B, Figure B-8: Task Header, Fixed Part is incorrect. The correct figure is shown next.
- 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.

---

### 3.15 RSX-11M-PLUS and Micro/RSX Utilities Manual

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

- Please add the following supplemental information to Chapter 3 in the description of the Backup and Restore Utility (BRU):

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.

- Please add the following information to Table 3-3:

**Table 3-6 Summary of BRU Command Qualifiers**

Command Qualifiers	Options	Defaults
/COMPARE	/DOUBLEBUFFER /SINGLEBUFFER	/DOUBLEBUFFER with MU-type devices only. Otherwise, /SINGLEBUFFER is the default.
/VERIFY	/DOUBLEBUFFER /SINGLEBUFFER	/DOUBLEBUFFER with MU-type devices only. Otherwise, /SINGLEBUFFER is the default.

- Please add the following information to Section 3.3.5 on page 3-13:

The BRU qualifier /COMPARE has two options: DOUBLEBUFFER and SINGLEBUFFER. The options are used only with MU-type devices when 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.

**Note: 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.**

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 format of the /COMPARE qualifier is shown next.

**/COMPARE[:option]**

### Parameter

#### option

Specifies one of the following two options:

Option	Meaning
SINGLEBUFFER	<p>If you use the SINGLEBUFFER option for a compare 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.</p> <p>The SINGLEBUFFER option is the default for devices that are not MU-type devices.</p>
DOUBLEBUFFER	<p>The DOUBLEBUFFER option can be used only on MU-type devices. For most system configurations with MU-type devices, a compare 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 the integrity of the remainder of that block. All the remaining file IDs or file names will be listed as potential errors.</p> <p>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.</p>

### Examples

```
BRU>/COMPARE:SINGLEBUFFER DU0: MU0: Return
```

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

```
BRU>/COMPARE:DOUBLEBUFFER DU0: MU1: Return
```

## Corrections to Documentation

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.

- Please add the following information to Section 3.3.33 on page 3-25:

The BRU `/VERIFY` qualifier has two options: `DOUBLEBUFFER` and `SINGLEBUFFER`. The options are used only with MU-type devices when verifying information from a disk to a tape. The input device must be a disk and the output device must be a tape or tapes.

**Note:** 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 format of the `/VERIFY` qualifier is:

`/VERIFY[:option]`

### Parameter

#### option

Specifies one of the following two options:

Option	Meaning
<code>SINGLEBUFFER</code>	<p>If you use the <code>SINGLEBUFFER</code> option for a verify operation from a disk to a tape, BRU will print the first error found and continue verifying. It prints any other errors found.</p> <p>The <code>SINGLEBUFFER</code> option is the default for devices that are not MU-type devices.</p>
<code>DOUBLEBUFFER</code>	<p>The <code>DOUBLEBUFFER</code> option can be used only on MU-type devices. For most system configurations with MU-type devices, a verify operation from disk to tape will be more efficient if you use the <code>DOUBLEBUFFER</code> option. However, the error messages that are reported when you use the <code>DOUBLEBUFFER</code> option are less explicit than the messages reported when you use the <code>SINGLEBUFFER</code> option. As soon as an error occurs, BRU reports the actual error. If there is any subsequent data in the block, BRU cannot accurately verify the integrity of the remainder of that block. All the remaining file IDs or file names will be listed as potential errors.</p> <p>You must use an MU-type device drive that supports the compare host-data function. The <code>DOUBLEBUFFER</code> option is the default for MU-type devices.</p>

### Examples

```
BRU>/VERIFY:SINGLEBUFFER DU0: MU0: 
```

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

```
BRU>/VERIFY:DOUBLEBUFFER DU0: MU1: 
```

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.

- Please add the following information to Section 3.10.3:

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

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

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

## Corrections to Documentation

### 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-\*FATAL\*—Continuation volumes cannot 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.

- Please add the following information to Section 12.1.3.1 on page 12-7:

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:

```
>dir lb:[rms*]rms*.tsk
```

```
Directory lb:[RMS001001]  
20-JAN-93 09:28
```

```
RMSRESAB.TSK;9      48.      C 30-JAN-93 13:57  
RMSLBL.TSK;9       19.      C 30-JAN-93 13:57  
RMSLBM.TSK;9       31.      C 30-JAN-93 13:57
```

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

```
Directory lb:[RMS001054]  
20-JAN-93 09:28
```

```
RMSCNV.TSK;5       176.     C 26-JAN-93 11:33  
RMSIFL.TSK;5       136.     C 26-JAN-93 11:34  
RMSRST.TSK;5       191.     C 26-JAN-93 11:34  
RMSBCK.TSK;5       187.     C 26-JAN-93 11:34  
RMSDES.TSK;5       248.     C 26-JAN-93 11:34  
RMSDSP.TSK;5       183.     C 26-JAN-93 11:33  
RMSDEF.TSK;5       139.     C 26-JAN-93 11:33
```

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

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

- Please add the following information to Section 14.5 on page 14-9:

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.

**3.16 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	

The examples in Sections J.2 and J.3 are incorrect. The correct examples are as follows:

**Program CRCOPA**

The following sample program is titled, CRCOPA. The CRCOPA program uses a data-set descriptor instead of the default filename block used in CRCOPY.

```

        .TITLE  CRCOPA                ; Card reader copy routine
        ;
        .MCALL  FDBDF$,FDAT$,FDRC$,FDOP$,NMBLK$,FSRSZ$
        .MCALL  OPEN$R,OPEN$W,GET$,PUT$,CLOSE$,EXIT$$
        .MCALL  FINIT$
        ;
        INLUN=3
        OUTLUN=4                      ;Assign CR or file device
        FSRSZ$ 2                      ;Assign to output device
FDBOUT: FDBDF$
        FDAT$  R.VAR,FD.CR
        FDRC$  ,RECBUF,80.
        FDOP$  OUTLUN,OFDSPT
FDBIN:  FDBDF$
        FDRC$  ,RECBUF,80.
        FDOP$  INLUN,IFDSPT
RECBUF: .BLKB 80.
    
```

## Corrections to Documentation

```

OFDSPT: .WORD 0,0 ;Device descriptor
        .WORD 0,0 ;Directory descriptor
        .WORD ONAMSZ,ONAM ;Filename descriptor

IFDSPT: .WORD 0,0 ;Device descriptor
        .WORD 0,0 ;Directory descriptor
        .WORD INAMSZ,INAM ;Filename descriptor

ONAM: .ASCII /OUTPUT.DAT/
      ONAMSZ=-INAM
      .EVEN

INAM: .ASCII /INPUT.DAT/
      INAMSZ=-INAM
      .EVEN

START: FINIT$ ;Init file storage region
      OPEN$R #FDBIN ;Open the input file
      BCS ERROR ;Branch if error
      OPEN$W #FDBOUT ;Open the output file
      BCS ERROR ;Branch if error
GTREC: GET$ #FDBIN ;Note - URBD is all set up
      BCS CKEOF ;Error should be EOF indication
      MOV F.NRBD(R0),R1 ;R1=size of record read
      MOV #RECBUF,R2 ;Strip trailing blanks
      ADD R1,R2
10$: CMPB #40,-(R2)
     BNE PTREC
     SOB R1,10$

;At this point, R1 contains the stripped size of the
;record to be written. If the card is blank,
;a zero-length record is written.

PTREC: PUT$ #FDBOUT,,R1 ;R1 is needed to specify
      BCC GTREC ;the record size

ERROR: JMP ERRSUB ;Jump to ERROR code

CKEOF: DMPB #IE.EFO,F.ERR(R0) ;End of file?
      BNE ERROR ;Branch if other error

      CLOSE$ #FDDIN ;Close the input file
      BCS ERROR
      CLOSE$ #FDBOUT ;Close the output file
      BCS ERROR
      EXIT$$ ;Issue exit directive
      .END START

```

### Program CRCOPB

The following sample program is titled, CRCOPB. The CRCOPB program uses run-time initialization of the File Descriptor Block (FDB).

```

        .TITLE CRCOPB ; Card reader copy routine
;
        .MCALL FDBDF$,FDAT$,FDRC$,FDOP$,NMBLK$,FSRSZ$
        .MCALL OPEN$R,OPEN$W,GET$,PUT$,CLOSE$,EXIT$$
        .MCALL FINIT$
;
        INLUN=3
        OUTLUN=4 ;Assign CR or file device
        FSRSZ$ 2 ;Assign to output device
FDBOUT: FDBDF$
FDBIN: FDBDF$
RECBUF: .BLKB 80.

OFDSPT: .WORD 0,0 ;Device descriptor
        .WORD 0,0 ;Directory descriptor
        .WORD ONAMSZ,ONAM ;Filename descriptor

```

```

IFDSPT: .WORD 0,0 ;Device descriptor
        .WORD 0,0 ;Directory descriptor
        .WORD INAMSZ, INAM ;Filename descriptor
ONAM: .ASCII /OUTPUT.DAT/
      ONAMSZ=.-ONAM
      .EVEN
INAM: .ASCII /INPUT.DAT/
      INAMSZ=.-INAM
      .EVEN
START: FINIT$ ;Init file storage region
      OPEN$R #FDBIN, #INLUN, #IFDSPT, , #RECBUF, #80.
      ;Runtime initialization
      BCS ERROR ;Branch if error
      FDATA$R #FDBOUT, #R.VAR, #FD.CR ;Runtime initialization
      OPEN$W R0, #OUTLUN, #OFDSPT, , #RECBUF, #80.
      BCS ERROR ;Branch if error
GTREC: GET$ #FDBIN ;Note - URBD is all set up
      BCS CKEOF ;Error should be EOF indication
      MOV F.NRBD(R0), R1 ;R1=size of record read
      MOV #RECBUF, R2 ;Strip trailing blanks
      ADD R1, R2
10$: CMPB #40, -(R2)
      BNE PTREC
      SOB R1, 10$

;At this point, R1 contains the stripped size of the
;record to be written. If the card is blank,
;a zero-length record is written.

PTREC: PUT$ #FDBOUT, , R1 ;R1 is needed to specify
      BCC GTREC ;the record size

ERROR: JMP ERRSUB ;Jump to ERROR code

CKEOF: CMPB #IE.EFO, F.ERR(R0) ;End of file?
      BNE ERROR ;Branch if other error

      CLOSE$ #FDDIN ;Close the input file

      BCS ERROR
      CLOSE$ #FDBOUT ;Close the output file
      BCS ERROR
      EXIT$$ ;Issue exit directive
      .END START

```

### 3.17 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.

## Corrections to Documentation

- In Appendix A, 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 Appendix A, 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 Appendix A, 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.

---

### 3.18 RSX-11M/M-PLUS RMS-11 User's Guide

Please add the following corrections to the *RSX-11M/M-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:ROEXSY:ROIMPA-LB: [1,1]RMSDAP/LB:ROAULS
```

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.

---

### 3.19 RSX-11M/M-PLUS RMS-11 Utilities

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

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

## Corrections to Documentation

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 Chapter 2, 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 Chapter 2, 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 Chapter 2, 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 Chapter 2, 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 Chapter 2, 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 Chapter 2, 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 Chapter 4, 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).

## Corrections to Documentation

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 ddn:file.dat;n as output  
-RMS-E-ER$FEX, File already exists
```

## Corrections to Documentation

- 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 display the following error message:

```
?RST-F-CREOUT, Error opening ddmn:file.dat;n as output
```

```
-RMS-E-ER$FEX, File already exists
```



# 4

---

## RMS-11 V2.0 Supplementary Information

This chapter describes RMS-11 software modifications, lists restrictions, and supplementary information for RSX-11M-PLUS.

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.

---

### 4.1 RMS-11 Enhancements

RMS-11 has included the following enhancements:

- Include support for century inferencing within utilities
- Correct input date parsing to allow years beyond 1999.
- Support ISO 8601:1988 standard date formats

---

### 4.2 RMS-11 Corrections

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

---

#### 4.2.1 RMS-11 Local Access

RMS-11 applications, that performed multistreaming 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.

---

#### 4.2.2 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.

## RMS-11 V2.0 Supplementary Information

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

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

---

### 4.2.3 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.

RMSIFL incorrectly reported exception records on packed decimal type records containing negative key values. This problem has been corrected.

Backup sets created on RSX-11M, or octal-based RSX-11M-PLUS systems, were restored with incorrect version numbers. RMSBCK incorrectly assumed that the source file came from a decimal-based system. For example: file FOO.DAT;10 was restored as FOO.DAT;12 even though both the input and output systems used octal version numbers. When this same file was restored to an RSX-11M-PLUS system that supported decimal version numbers, the version number remained ;10 rather than being converted to ;8. This problem did not exist if the input system supports decimal version numbers. This problem has been corrected.

---

## 4.3 Software Restrictions

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

---

### 4.3.1 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.

---

### 4.3.2 Restrictions to RMS-11 Remote Access Methods (RMSDAP)

The following sections describe restrictions to RMS-11 Remote Access Methods.

---

#### 4.3.2.1 DAP Date Restrictions

The DAP specification limits the date range available for remote files. The Data Access Protocol (DAP) specifies a date range of 1970 through 2069. Files which have creation or revision dates outside of this range will not be interpreted correctly.

---

#### 4.3.2.2 Incorrect Interpretation of Keysize Field Values

RMSDAP does not interpret a keysize field value of zero correctly. This causes keyed access to a remote indexed file to fail. The application passes the value zero for access to a nonstring key and issues the following error message:

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

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

---

#### 4.3.2.3 Using RMSDAP on Your System

The RMSDAP provided on your system uses a 576 byte buffer for transfers between your program's record/user buffer and your DECnet-RSX system network buffer. This buffer must be large enough to hold the record and the DAP message overhead (approximately 36 bytes). In certain rare cases, this buffer length may be insufficient.

In particular, if you access a sequential variable file with a Maximum Record Size (MRS) of zero (used when no maximum has been set), and the sum of the largest record length (LRL) and the DAP message overhead exceeds the 576 byte length, your record access will fail with the RMS error ER\$MRS. This occurs because RMSDAP functions at a DAP protocol level that does not allow reconfiguration of buffersize without actually closing and reopening the file (and that is an unacceptable alternative for RMSDAP). If it is within your control to create the remote file, then an explicit MRS should be used. If not, the local RMSDAP buffer can be lengthened by the system manager (prior to run time) depending on the following considerations:

- If your program was built with in-task RMSDAP, consult your map to determine the location of the variable \$BUF.M (current contents 1100 octal) in the module R0NFRT in your application. After saving a copy of your application, ZAP the desired buffer length (including DAP overhead) into your application. This change will only affect your application.
- If your program was built against the DAPRES library, consult the map (LB:[1,34]DAPRES.MAP) and determine the location of \$BUF.M in module R0NFRT. Remove the DAPRES from memory. After saving a copy of LB:[1,1]DAPRES.TSK, ZAP in the desired buffersize. Reinstall the DAPRES library. This change will affect all programs built and run against the DAPRES library.

In both cases the increased size of the buffer will be reflected in the size of your RMSDAP applications.

**Note: Network transfers rely on configuration of local RMSDAP and DECnet-RSX network buffers as well as remote FAL and DECnet-RSX network buffers. In all negotiations, the smaller buffer length**

## RMS-11 V2.0 Supplementary Information

is used. If your local RMSDAP buffer size exceeds the local DECnet-RSX buffer, the remote FAL, or the remote network buffer size the excess will be unusable, thus effectively eliminating the workaround.

---

### 4.3.3 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.COM file:

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

---

## 4.4 RMS-11 Supplementary Information

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

---

### 4.4.1 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.

## 4.4.2 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.

The paragraph in Section 5.4.7 on page 5-43 in the *RSX-11M-PLUS System Generation Guide and Installation Guide* is incorrect. Please replace it with the following paragraph:

RMS-11 Version 2.0 is included on the pregenerated kit disk. The RMS-11 segmented library (RMSRES, and RMSLBL through RMSLBM) and all the RMS-11 utilities are already installed in the system image. No further installation is needed, unless you install the DECnet package on your system and you want to use the RMS-11 remote access facilities. See Section 5.4.1 in the *RSX-11M-PLUS System Generation Guide and Installation Guide* for information on installing the RMS-11 remote access package (DAPRES).

### 4.4.2.1 Location of RMS-11 Files on the Distribution Kit

Table 4-1 describes the location of RMS-11 Version 2.0 files on the RSX-11M-PLUS Version 4.6 distribution kit.

**Note:** Some file names are marked with an asterisk (\*). This indicates that the files are not included on restored RL02 distribution kits. They are excluded because of space limitations on an RL02, and because they are used only to rebuild components of RMS-11 Version 2.0. These files are located in the main distribution backup set on the distribution tape, and if required may be extracted from that set.

**Table 4-1 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	LB:[1,24]	For rebuilding RMSLBL and RMSLBM
RMSROTAB.STB		
RMSRESAB.CMD	LB:[1,24]	For rebuilding RMSRESAB
RMSRESAB.ODL		
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

## RMS-11 V2.0 Supplementary Information

**Table 4-1 (Cont.) RMS-11 Files on RSX-11M-PLUS Distribution Kit**

File Name	Destination	Comments
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
RMSRLX.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
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; <i>not</i> 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*	LB:[1,34]	Maps for segmented library
RMSLBM.MAP*		
DAPRES.MAP*		
RMSRES.TSK	LB:[3,54]	Task image for linking in supervisor mode; <i>not</i> to be installed as the root of the library
RMSRES.STB	LB:[3,54]	STB file for linking supervisor-mode tasks
RMSLBL.CMD*	LB:[1,24]	For rebuilding RMSLBL
RMSLBL.ODL*		
RMSLBM.CMD*	LB:[1,24]	For rebuilding RMSLBM
RMSLBM.ODL*		
DAPRES.CMD*	LB:[1,24]	For rebuilding DAPRES
DAPRES.ODL*		
BCKNON.CMD*	LB:[1,24]	For rebuilding the overlaid version of RMSBCK
BCKNON.ODL*		
BCKNRN.CMD*	LB:[1,24]	For rebuilding the resident-library version of RMSBCK
BCKNRN.ODL*		
BCKNSN.CMD*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSBCK
BCKNSN.ODL*		
CNVNON.CMD*	LB:[1,24]	For rebuilding the overlaid version of RMSCNV
CNVNON.ODL*		

**Table 4-1 (Cont.) RMS-11 Files on RSX-11M-PLUS Distribution Kit**

File Name	Destination	Comments
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
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
RMSUTL.OLB* RMSODL.ODL*	LB:[1,24]	For rebuilding the utilities

## RMS-11 V2.0 Supplementary Information

**Table 4-1 (Cont.) RMS-11 Files on RSX-11M-PLUS Distribution Kit**

File Name	Destination	Comments
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.

### 4.4.2.2 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.

---

#### 4.4.2.3 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.

---

#### 4.4.2.4 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
RMSDES	DESNON.CMD, DESNON.ODL, and RMSDES.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
RMSDES	DESNRN.CMD and DESNRN.ODL

## RMS-11 V2.0 Supplementary Information

Utility	Files
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 CNVNRN.CMD and CNVNRN.ODL. To build RMSCNV using disk-overlaid RMS-11 and RMSDAP, use the files CNVNOO.CMD and CNVNOO.ODL.

### 4.4.2.5 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.

---

# 5 Layered Product Support

---

## 5.1 Correction Files

Table 5-1 lists the layered products and correction files that have been updated for the RSX-11M-PLUS Version 4.6 operating system. The application of the updated correction files to each layered product will result in improved software performance. Table 5-1 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 5-1 Layered Product Correction File References**

Layered Product	Directories Used	Backup Set	Documentation File Name
BASIC-PLUS-2 Version 2.7	[211,200]	BP2	[211,200]BP2.DOC
COBOL-81/RSX * Version 3.1	[215,200]	C81	[215,200]C81.DOC
DATATRIEVE-11/RSX Version 3.3A	[222,200]	DTR	[222,200]DTRREL.DOC
PDP-11 FORTRAN IV Version 2.8	[245,200]	FORTIV	[245,200]FORTIV.DOC
PDP-11 FORTRAN-77 Version 5.4	[246,200]	F77	[246,200]F77.DOC

---

## 5.2 Installing Layered Products Restrictions

The MCR command SET /NONAMED or the DCL command SET DEFAULT/NONAMED should be used before installing layered products because the installation procedures will not install the products correctly with named directory support enabled.

When installing layered products from the console, it is important to log off the console after startup is complete and then log in again as a privileged user. Installations will fail if this is not done, because the console default login during startup does not have certain required logicals defined. These are defined during the conventional login procedure and are taken from the system account file.

---

### 5.3 Invalid BASIC-PLUS-2 Installation Warning

BASIC-PLUS-2 Version 2.6 checks for the presence of the slow Task Builder (STK) even though the installation does not need that utility. You may ignore any warnings displayed when you install BASIC-PLUS-2 Version 2.6 regarding the absence of the slow Task Builder. Version 2.7 of BASIC-PLUS-2 does not exhibit this problem.

---

#### 5.3.1 BASIC-PLUS-2 DATE\$ Function

The BASIC-PLUS-2 DATE\$ function has been updated to correctly display dates in a 2 digit year format for the years 2000 through 2035.

The DATE\$ function treats the value passed to it as an unsigned 16 bit integer. The value passed by BASIC-PLUS-2, however, is always promoted to a 16 bit signed integer. (Note that this behaviour is different from VAX-BASIC in which the DATE\$ function expects a 32 bit integer.)

The calculation for the value to be passed to the DATE\$ function can remain unchanged up to 31-Dec-02. For dates after this, the calculation can be made as follows ...

```
10 REM Y is real and IY% is an integer
   YEAR% is the Calendar year and
   DAY% is the number of days since Jan 1
   ---
20 Y = (YEAR%-1970%)*1000 + DAY%
25 IF (Y > 32767)
   THEN
     IY% = Y - 65536
   ELSE
     IY% = Y
30 PRINT DATE$(IY%)
```

The following are example values for IY% and the corresponding date:

IY	DATE
32001	1-Jan-02
32365	31-Dec-02
-32535	1-Jan-03
-32171	31-Dec-03
-535	1-Jan-35
-171	31-Dec-35

---

### 5.4 FORTRAN-77 DATE/IDATE Functions

The FORTRAN-77 DATE function has been updated to return the correct ASCII year value for years 2000 and beyond.

The page header lines in the FORTRAN list files have been updated to correctly print the system date for years 2000 and beyond.

For compatibility with VAX-FORTRAN, the PDP-11 FORTRAN-77 IDATE function has been modified to return the year of century. This will allow applications which currently display the year as two digits to continue to do so without any change to the application; i.e. 00 through 99.

**CAUTION: This behaviour is different from that which is documented in the FORTRAN-77 Users Guide.**

Previously IDATE returned the number of years since 1900. Some applications may prefer to use this value instead. For example when used in date comparisons, this original behaviour would be desired, so that sequential dates will be in ascending order.

To preserve the original behaviour, the \$IDATE module should be extracted from the FORTRAN OTS library before applying this update.

```
>LBR IDATE.OBJ=[1,1]F77FCS.OLB/EX:$IDATE
```

The update should now be performed and IDATE can be replaced into the appropriate library after the update process is complete, or kept as a separate object module and included in task builds as required.

---

### 5.5 Datatrive-11 Report Writer Heading

The Datatrive-11 Report Writer has been update to correctly display the date at the top of the reports, after 31-December-1999.



# 6

---

## Virtual Device Subsystem Reference

This chapter describes the Virtual Device Subsystem which is included with RSX-11M-PLUS Version 4.6. The subsystem allows the user to create a virtual disk or tape device from a file on a Files-11 volume, and to create a virtual disk device from a set of physical disks bound into a single volume, or from a dynamic common region in memory.

---

### 6.1

#### Introduction

The Virtual Device Subsystem comprises two parts: VFDRV (the Virtual Device Driver) and VCP (the Virtual device Control Program). VCP provides the user interface to create and manage virtual devices.

The Virtual Device Subsystem has the following features:

- Create a virtual disk or tape from a container file on a physical or another virtual disk
- Create a bound volume set from two or more physical devices, accessed as a single virtual disk
- Create a virtual disk from a dynamic common memory region
- Create image copies of disks or tapes
- Hierarchical structure prevents inappropriate deletion or dismounting
- Supports shadowing and caching (permits shadowing unlike physical disks)
- Permits dynamic expansion of the number of virtual units
- Provides diagnostic capabilities

The rest of this chapter will discuss how to invoke VCP, how to use virtual devices, a detailed command list, and a technical discussion of the subsystem.

---

### 6.2

#### VCP Command Line

VCP (Virtual device Control Program) is invoked like standard installed utilities, either by direct command line, or input to the VCP> prompt. VCP accepts a task indirect command file as input.

**Format**

**>VCP command**

**>VCP**

**VCP>command**

**>VCP @commandfile**

## Virtual Device Subsystem Reference

VCP commands are all of the form *verb parameter /switches*. The standard VCP commands are shown below:

**CONnect**  $\left[ \begin{array}{l} \text{ddnn:}[\text{,ddnn:}[\text{,...ddnn:}]] \\ \text{file} \\ \text{main\_partition/MEM} \end{array} \right] \text{[/switches]}[=\text{VFnnn:}]$

**DISConnect**  $\left[ \begin{array}{l} \text{virt\_dev:}[\text{/switches}] \\ \text{/ALL} \\ \text{/USER} \end{array} \right]$

**SHOw** [ddnn:][/switches]

**DUMp** [ddnn:][/switches]

**COPy** ddnn: [/switch] [TO] ddnn: [/switch]

**HELp** command [option]

**CREate** [dd:] /UNITS=ddd[/switches]

**TRAcE** virt\_dev: [/switches]

**STOP** virt\_dev:

**SEt** ddnn: [/switches]

### Options

#### ddnn:

A valid RSX device or logical name.

#### file

A valid Files-11 filespecification.

#### /switch

Optional switch to control command effect.

**Note:** Switches must be specified with 3 characters or fewer.

#### dd:

A valid device, but must not contain a unit number.

#### virt\_dev:

Any virtual device handled by VF (may be a name other than VF)

---

## 6.3 Using Virtual Devices

This section briefly describes typical uses of virtual devices with examples and explains some of the terms. Detailed explanations of the commands with more additional options and functions is described in the *Commands* section that follows.

### 6.3.1 Setup the Virtual Device Units

Virtual Devices, like physical devices, comprise two parts, the drives and the media. The Virtual Device drive is a VF: unit. The Virtual Device media is usually a container file (other options are available and described in the *Commands* section.)

For those planning to use Virtual Devices routinely, it is recommended that the initialization and other setup commands be added to the system startup command file.

To use virtual devices, the virtual device driver VF: must be loaded and the control program VCP must be installed.

```
>LOA VF:/PAR=GEN/VEC
>INS $VCP
```

Next, some virtual device units should be CREATED. These are units for the device type VF: It is best to create as many of these as you are likely to need. While virtual devices can be assigned normal device names like DU3: and MU4:, they all still have equivalent VF: units.

You only need to create as many virtual units as you wish to have in use at any one time. You can create additional units at any time "on the fly", so deciding how many you want at the start does not fix you to this number.

Say you expect to use 4 virtual disk units and 2 virtual tapes.

```
>VCP CRE /UNITS=6
```

would simply create 6 VF: units, but you'd prefer them to have conventional disk and tape names like DU3: or MU2:, so you can enter the following instead

```
VCP CRE DU:/UNITS=4
VCP CRE MU:/UNITS=2
```

The device units are now available for use. The next section will discuss the equivalent of "Loading the media in the Drive".

### 6.3.2 Setting up Virtual Device Container Files

The Virtual Device media is usually a container file. You can create new media (container files) or you can use existing media, and you can give the media special attributes. In each case the procedure to access the container files is to CONNECT the VF: virtual device unit to the container file.

For example, to create a brand new DU type disk of 40,000 blocks in a container file called MYDISK.DSK :

```
>VCP CON MYDISK.DSK/CR:40000./DRV:DU
VCP -- Device VF1: (DU3:) has been assigned.
>
```

Similarly, you can create a disk the specific size of an RX50 diskette ...

```
>VCP CON RX50DISK.DSK/CR:RX50
VCP -- Device VF1: (DU3:) has been assigned.
>
```

You can specify other disk types in the /CR: switch to create disks of the equivalent size.

## Virtual Device Subsystem Reference

To create a tape, the process is similar, although in this case, the size you specify is the initial length of the tape container, and it will be extended as required.

```
>VCP CON MYTAPE.TAP/CR:1000./TAPE/DRV:MU
VCP -- Device VF5: (MU1:) has been assigned.
>
```

When you are done with the virtual disk or tape, after dismounting the device in the normal manner, you must remove the association between the VF: unit and the file. This is done with the DISCONNECT command.

```
>VCP DIS MU1:
>
```

---

### 6.3.3 Using the devices

A virtual disk or tape can be used just like a real disk or tape, by initializing it, mounting it, and it is indistinguishable from using a real device with all the RSX utilities, like PIP, BRU and so on.

There are some other useful things one can do with a virtual disk or tape.

To make an IMAGE copy of a tape loaded on MU0:, use the VCP copy function to copy the tape to a virtual tape container, and then put a blank tape on the real tape drive and copy it back with VCP. VCP will copy most arbitrary tape formats, provided they end with a double tape mark.

Here is an example of copying a tape ... MU0: is the real tape drive, and MU1: is the virtual ...

```
>MOU MU0:/FOR
>VCP CON MYTAPE.TAP/CR:1000/DRV:MU/TAPE/MOU:FOR
VCP - Device VF6: (MU1:) has been assigned.
>VCP COP MU0: MU1:
VCP - Copy started
VCP - Copy finished
>DMO MU0: ! Remove the original tape and load the new tape
>MOU MU0:/FOR
>VCP COP MU1: MU0:
VCP - Copy started
VCP - Copy finished
>DMO MU1:
>VCP DIS MU1:
```

Note the Connect command has the switch /MOU:FOR VCP allows you to specify a subsequent Mount command in the VCP command line and perform the mount.

After disconnecting the tape, it can be connected to later to make another copy, or copied by DECnet to another system and a tape created there, or even copied by FTP to another system and recreated there.

Disks can be copied in a similar manner. This permits you to store diskettes on a large hard drive and recreate them later. A useful way to help preserve distribution media.

You can install software from virtual tapes or disks just as if it was the original media.

---

## 6.4 Commands

The following section describes the commands. Note that some commands are privileged.

---

### 6.4.1 The CONnect command

The connect command is used to connect a file, a set of bound physical devices, or a memory region to a virtual device unit. The device attributes can be tailored, including the device size, type and the device name. Other attributes are described below.

#### Formats

**CON file\_name[/switches][=device]**

**CON device\_1,[,device\_2,device\_n][/switches][=device]**

**CON [main\_partition]/MEM[:size][/switches][=device]**

The first form is to connect a Virtual Device to a file (file mode).

The second form is to bind physical devices as a bound volume.

The last form is to connect a Virtual Device to a memory region.

#### file\_name

This field must contain a valid Files-11 filespec, and the device that the file resides on must be mounted. The user must have read, write, and extend access privileges to the file. The file must not be accessed by another user, or connected as another virtual device.

#### file\_type

The file type for a virtual disk container is ".DSK" and for a virtual tape device is ".TAP".

#### Notes:

1. When accessed in device mode, the target device must be mounted foreign and unallocated. If passthru mode (diagnostic) is selected, neither the virtual nor physical device should be mounted.
2. When accessed in file mode, the container file may be non-contiguous, but in tape mode, must be non-contiguous to be extended. For best performance, the files should have as few fragments as possible.
3. The connect operation checks access for the user. After connection, all access is covered by the RSX I/O mechanism, and if a volume is mounted, by the Files-11 volume access controls. The actual container is locked from file read-write activity, thus preventing deletion. Additionally, because of the hierarchical structure of nested virtual disks, a disk volume closer to the tree of virtual disks cannot be dismantled.
4. On exit, VCP will put the VF unit number which was connected into the upper byte of the exit status word.

#### Switches

##### **/CR—The CReate switch(File mode)**

The CReate switch instructs VCP to create the file specified. The switch takes an optional parameter specifying the size of the file in blocks. If the device

## Virtual Device Subsystem Reference

is not mounted Files-11, or the user does not have the privileges required to create the file, the operation will fail.

The switch also allows a device type to be specified. For virtual disk devices, the file will be created with a size equivalent to the actual size of the physical device type specified. If the device type form of the create function is used, it will also establish defaults for the actions of the /TYPE switch, and whether the /BAD function should create a last track descriptor, or a standard bad block descriptor.

For virtual tape devices, if a device type is specified in the create command, the type specified will define the default for the device name, and the initial size of the file will be set to 40. blocks.

Disk file attributes will be defined as: fixed length, 512.byte records, best try contiguous, with the End-of-File marker set to the end of the highest block allocated.

Tape file attributes will be defined as: undefined record attributes, no record length, best-try contiguous but marked non-contiguous, with the End-of-File marker set to the end of the first block allocated.

### **/TAP—TAPe (File mode)**

The /TAP (tape) switch specifies that the file being connected is a tape format file, and that a virtual tape device should be set up to connect to the file.

### **/TYP—TYPE**

The /TYP (type) switch associates a specific device type with the file being connected, and will be available from the UCB Extension. The argument for this switch must be a string of valid RAD50 characters, which are meaningful to associate with the device.

When this switch is used in conjunction with the /CReate switch, the functionality of the /SAVe switch documented below will be included.

```
CONnect DU:[1,1]RP06/CR:RP06/TYP:XRP06
```

This command will create the file DU:[1,1]RP06.DSK, with 340,670 blocks, and will associate the device type of "XRP06" with the file. Once the file is created, the file will be connected to the next available VF: unit.

### **/SAV—SAVe device info (File mode)**

The /SAV (save) switch will associate a different "device type" with the "volume". It can only be used in conjunction with the /TYP switch, and will update the default volume "device\_type" stored in the file header of the "volume" or "tape".

### **/BAD—BAD block handling (Disk File mode)**

The /BAD switch is used to write a "bad block" descriptor in the last block of the file. Since the "volumes" are stored on a previously initialized Files-11 volume, the use of the BAD utility is considered an overhead to the use of the volume.

This function will make the volume appear that the BAD utility had been run on the volume prior to initialization. The BAD utility can be used to update the empty descriptor created by utilizing the /UPD function of BAD.

If used in conjunction with the /LT switch the bad block descriptor will be created which conforms to the format for a last track device.

This function should only be used with a volume which has not yet been initialized, although this guideline is not enforced.

### **/LT—Last Track (File mode)**

The **/LT** (last track) switch causes the bad block descriptor created to use the format indicated for last-track devices.

The argument for the switch is the actual number of blocks associated with the device type. If the device type was given as an argument to the create command, or the **/ATT** switch is specified, VCP will automatically determine if the device type is a last-track device, and save the last-track information for use when a **/BAD** operation is specified. An example of the command is:

```
CONnect DU:[1,1]RL02/CR:20480/TYP:RL02/BAD/LT:20/DRV:DL:
```

This will set the device attributes for an RL02 device. By using either the **/ATT** switch, or a specific device type as the argument for the **/CR** switch, the last track information can automatically determined.

```
CONnect DU:[1,1]RL02/CR:RL02/BAD
```

Both of the above commands perform an identical function.

### **/SN - Serial Number (File mode)**

The **/SN** (serial number) switch can be used in conjunction with the last track information obtained to write a specific volume serial number into the last track descriptor. The argument to the switch is a decimal number which will be placed into the first word of all last track descriptors created.

Using the example command above, the variation which would create a volume serial number of 400 would be:

```
CONnect DU:[1,1]RL02/CR:RL02/BAD/SN:400
```

### **/RON - Read Only**

The **/RON** (read-only) switch sets the device to read-only mode. Any attempts to write to the volume will be rejected with the status code of "Device write locked".

### **/DRV—Driver**

The **/DRV** (driver) switch will associate the connected volume with a specific device type other than VF. Prior to using this switch, the **CREATE** command must have been issued to create the data structures for the alternate device type.

```
CREATE DB:/UNIT=2 or  
CREATE /UNIT=2/ALTERNATE  
  
CONNECT DU:[1,1]RP06/DRV:DB: or  
CONNECT DU:[1,1]RP06/ATT:RP06
```

These commands will create the data structures to be able to connect two virtual devices either through the device name **VF:**, or the device name **DB:**.

The second command will associate the RP06 file with the first available **DB:**, or alternate, device data structure associated with **VF:**.

### **/NM—No Message**

The **/NM** (no message) function will disable the display of informational messages.

## Virtual Device Subsystem Reference

### **/MOU—MOUNT**

The /MOU (mount) switch instructs VCP to mount the volume after it has been successfully connected. An argument list for the switch can be used to specify how the volume should be mounted. Each argument can be at most 3 characters long, and up to 5 arguments may be included.

The syntax for the command is:

```
CONNECT DU:[1,1]RP06/MOU:argument[:arg_2[:arg_3[... ]]]
```

Valid arguments for this switch are any three character switches which are legal to the MCR MOUNT command. A few examples are:

OVR - Mount the volume Files-11  
FOR - Mount the volume as foreign  
PUB - Mount the volume and set as public

### **/[-]EXT:nnnn—Extend file (Tape files only)**

The /EXT (extend) switch indicates that the tape file being connected should have extend access allowed and when extension is required, the extension size will be the value specified. "nnnn" can be any value between 1 and 32767. If the switch is negated, then the tape file will not be extendable, and will return an IE.EOT error when approaching the end of the usable file space.

The default for this option is /EXT:200, which specifies that tape format files can be extended by 200 blocks per extension request.

### **/ATT—Attributes**

The /ATT (attributes) switch configures the virtual device to appear as the device type indicated. The device type information can be obtained from a number of sources, including the saved device type information in files, a /TYP switch included on the command line, or the device type specified in either /CR or /SIZ switches.

When used, VCP will attempt to connect the device to the device database entry which has the correct device name, indicate the device type in the UCB Extension, and determine what type of bad block information would be needed if the /BAD switch is specified.

### **/PAS—PASsthru (Device mode)**

Passthrough mode is a function of VFDRV where a virtual device is setup identically to the actual target device specified. In passthrough mode, only one device may be specified, and neither the virtual nor physical device should be mounted in any way, due to the implications of passthrough.

Passthrough mode will transfer all I/O requests, both data transfer and control QIO's to the specified device. The primary benefit of passthrough mode is to enable trace functions through the passthrough driver which would normally not be able to be observed. The unit data structures are set up to be exactly as the target device, and normal I/O directed to the device will be redirected through the VF: unit. The connected device must not be mounted, or have any attached tasks at the time the connect is issued.

### **/SIZ—SIZE (Device mode)**

The /SIZ function allows the user to specify the type of device to be emulated. For shadowing purposes, device sizes must match exactly, and this function will allow multiple device bound together to more accurately reflect the intended device type. If the /ATT function is used in conjunction with the /SIZ switch, the other device characteristics such as device type, and device name will also be applied to create the correct environment for the application.

For example, two RK07 disk drives, and one RL02 can be merged to create an equivalent number of blocks as provided on an RM02/3 disk drive. Using the /SIZ and /ATT switches together can create a suitable device to shadow an RM02/3 disk drive; i.e.

```
CREATE DR:/UNIT=1
CONNECT DM0:,DM1:,DL2:/SIZ:RM02/ATT
```

This will create an environment suitable for shadowing a physical RM02 with the merged devices DM0:,DM1:, and DL2:.

### **/MEM—Memory**

The /MEM (memory) switch defines that the device will utilize a portion of the system's main memory as a mass storage device. The size can be given as either a physical type of device, or as a physical block size which must be smaller than the largest available "hole" in memory.

```
CREATE DU:/UNIT=1
CONNECT /MEM:RX50/ATT/BAD
```

In this example, an alternate device structure is created, with a device type of "DU", and the second command allocates an 800. block segment of memory, and utilizes it as an RX50 type device.

### **/SYS—SYStem (Privileged)**

The /SYS switch will connect a virtual disk with no defined owner (essentially the virtual device is owned by the system). Similar to the MOU /PUB, it will prevent disconnection of a virtual device which are intended to be a permanent device on the system. The difference between MOU/PUB and CON/SYS is that the non-disconnectable status of the virtual device for MOU/PUB only applies while the device is mounted. As soon as it is dismounted it is eligible for disconnection.

---

## 6.4.2 The DISconnect command

The disconnect command will disconnect a file, unbind a set of bound physical devices, or disconnect a memory region from a virtual device unit. The file associated with the device will be closed and the virtual device marked as offline. If the device was connected by an alternate device name, the unit data structures will be restored. The format of the Disconnect command is shown below:

### **Format**

**DIS [ddnn:]/[switches]**

### **Switches**

#### **/DEV, Sub-devices**

The /DEVice switch controls whether VCP will scan the other VF: data structures looking for a hierarchical device structure below the target device to be disconnected. If so, the devices below the target device will be dismounted and disconnected prior to the target device being dismounted and disconnected.

#### **/DMO**

The /DMO (Dismount) switch is used to request VCP to dismount the target device prior to disconnecting it. If the device is shadowed, this will also be stopped prior to the actual dismount operation. Once the device is dismounted, the disconnect operation will continue.

## Virtual Device Subsystem Reference

### **/ALL**

The **/ALL** option is used to disconnect all VF: devices in a hierarchical manner. All devices connected to the top level virtual disk will be disconnected, and the next device still connected will be done, until all devices are disconnected, or an error occurs.

### **/USE—USER**

The **/USE (USER)** switch will disconnect all VF: devices connected by the user during his current terminal session. For Privileged users the **/USE** switch will attempt to disconnect all VF: devices they own regardless of the mount status or the status of any recursive virtual devices. (See **/SYS** below)

---

### 6.4.3 The SHOW command

The show command will show the status of any virtual devices on the system.

#### **Format**

**SHO [device:][/switches]**

#### **device:**

is the virtual device (VFnn:) for which to show status. If omitted, the status of all devices will be displayed.

#### **switches**

are one or more of the switches described below.

#### **Status display for a connected disk and tape virtual drive**

```
VF1: 07226 Public,Mounted,Label=RSX11MPLBL82,Connected,Type=VRP06
      (DB3:)      File: DU2:[001001]RP06.DSK;1, Base_LBN: 3
                  FCB extension: 33670, Size: 340670
```

```
VF2: 07234 Tape,TT3: -Private,Mounted,Foreign,Connected,Type=VTS11
      (MS0:)      File: DU2:[001001]PLUSNETV4.TAP;1, Noncontiguous
                  Mapped_VBNs: 0,9287, Currebt_VBN: 2
                  Tape_Status: R/O,BOT, Current_idx/obj: 1./1 (DAT)
                  Largest_record: 512.
```

### **/ALL**

The **/ALL** switch includes all the other SHOW switches.

### **/DISK**

The **/DISK** switch is used to display all active virtual tape devices

### **/FULL**

The **/FULL** switch is used to include the device information for the selected virtual devices from their respective Unit Control Block Extensions

### **/TAPE**

The **/TAPE** switch is used to display all active virtual tape devices

### **Binary Dumps**

The following switches produce binary dumps of the specified data structures:

- /DCP** - Device Control Block
- /UCB** - Unit Control Block
- /SCB** - Status Control Block
- /FCB** - File Control Block Extension and Window Block
- /UCBX** - UCB Extension

/MAP - Noncontiguous file mapping pointers  
 /ODB - Tape object impure area  
 /IDX - Tape object descriptor buffer

A listing of all of the binary data structures which correspond to the display for VF1: shown above would be:

```

Device Control Block, VF: @ 052234
052234 / 065354 072226 043126 001001 000054 120052 177477 000070
052254 / 000000 177200 000377 000000 000000 000377 032224

Unit Control Block, VF1: @ 072226
072220 / 033670 000000 000000

072226 / 052234 072226 000721 002001 140110 000005 031276 001000
072246 / 072576 000000 006400 137154 001000 002241 070604 074054
072266 / 000000 000000 000000

FCB_Extension @ 033670
033670 / 115014 036134 000000 000003

File Window Descriptor @ 036134
036134 / 015402 000004 124774 000000 000000 000000 000000

UCB Extension @ 224100
140000 / 000000 000000 000000 000000 000000 000000 000000 000000
140020 / 000000 106140 140440 000126 000000 000000 000000 053000
140040 / 000000 000000 000000 000000 000000 000000 000000 000000
140060 / 000000 000000 000000 000000 000000 000000 000000
  
```

---

### 6.4.4 The DUMP command

The dump command is similar to the SHOW command, except that the display output is sent to a specified file, and that all VF: devices are shown.

#### Format

**DUMP file\_name[/switches]**

file\_name is a valid Files-11 output file spec and switches are described under the SHOW command. If an output file name is not specified, output is to TI:

---

### 6.4.5 The COPY command

The copy command is used to copy disks or tapes to another device, providing it is as large, or larger than the input device. The format of the command is as follows:

#### Format

**COPY indevice:[/switches] [TO] outdevice:[/switches]**

**COPY outdevice:[/switches]=indevice:[/switches]**

Valid switches are:

/FILE = Copy device in Files-11 mode (copy only allocated blocks)

/DEVICE = Copy all logical blocks

/WRITE = Perform write checking on the output device

## Virtual Device Subsystem Reference

`/BUFFER:size` =(\*Tape\*) Utilize buffer of "size" bytes

`/STATUS` = Display copy status in 1000. block increments

### Description

The COPY command can be used to duplicate disk/tape devices which performs a block by block copy operation. The command can be done in both Files-11 mode, or device mode, to perform the copy operation.

In Files-11 mode, the RSX-11M-Plus shadow catchup functionality is duplicated, which will utilize the input devices bitmap to determine which blocks should be copied.

In device mode, a block by block copy is performed, until the end of volume is reached, which will handle foreign device structures, tapes and non-Files-11 disks.

---

### 6.4.6 The CREate command (Privileged)

The CREATE command is used to dynamically add additional device structures. Each time the command is issued, an additional Device Control Block (DCB) data structure is created, however the original Status Control Block (SCB) is utilized by all units connected using the VF: driver.

Create is a privileged command, due to the potential impact it could have on primary pool, if a sufficiently large number of units are created. Each time the command is issued, one DCB (30. bytes), and "n" UCB's (46. bytes) are created. If a device name other than "VF:" is specified, or the "/ALternate" switch is specified, an additional DCB is also required.

Optionally, this command can be used to connect one of the virtual units to a another type of device, as specified in the command line. In this case, two DCB data structures are created which both point to the same set of Unit Control Blocks (UCB). Until the device is connected to the alternate device name, the device will not appear in the system device list, and the logical unit range will not be determined until a device is connected using the alternate DCB. This provides the interlock which will allow the VF: driver to be unloaded while alternate device data structures have been allocated, and eliminate any possible race condition which could occur while loading or unloading a physical device driver.

### Format

**CREate** [`dd:`]/UNIts=`ddd`[/switches]

#### **dd:**

The device name of the alternate device to connect. If omitted the command will only create additional VF: data structures. Legal device names are VF, Dx, and Mx, where Dx device types imply disk type data structures, and Mx device types imply tape device data structures. If the system supports deferred or runtime binding, the /ALternate switch should be used instead.

#### **ddd**

The number of devices to create from 1 to 255 (decimal), inclusive. The total number of VF: units cannot exceed 256.

### Switches

#### **/DEVICE=VFnn:**

The optional switch "/DEVICE" can be used to control whether an actual create function is performed. In the case of a startup command file, the units would only be created if the unit specified in the "/DEVICE" argument would be the first device created as a result of the CREATE command.

#### **/TAPE**

The /TAPE switch is used to configure the device data structure for a tape type device. For systems which support deferred or runtime binding, this determination is made dynamically at CONnect time, and need not be specified during the CREATE option.

#### **/ALternate**

The /ALternate qualifier is used on systems which support deferred or runtime binding to specify that an alternate DCB should be created, and the device attributes will be defined at the time the device is connected.

If the /TAPE or optional "dd:" elements are specified on a system which supports runtime or deferred binding, these indicators will imply the /ALternate qualifier.

---

### 6.4.7 The SET command (Privileged)

The SET command allows the user to dynamically change the attributes of a connected device which would be done through another series of commands.

Its primary functionality is to allow the user an increased flexibility in controlling the functions and states of a connected device. The format of the command is:

#### **Format**

**SET device:/switch\_1[/switch\_n]**

For some options, the device specified may be a device other than a VF: unit, and for other options only a VF: device is allowed to be altered. If an option is available for all devices on the system, it will be indicated in the description.

- ONLine - Set device as being ready
- OFLine - Set device into not ready status
- RON - Set device as being readonly
- RW - Set device as being read/write
- EXT - Set tape device as being able to extend
- ZERO - Clear all device counters
- REWind - Issue rewind command to tape device
- EOF - Update tape file <EOF> from tape region

## Virtual Device Subsystem Reference

### **/ONLine**

The ONLINE function is used to "ready" a device. For virtual disks, it will set volume valid, and for virtual tapes, it will reset the select error bit.

```
VCP>SET MM3:/ONLINE
```

### **/OFFline**

The OFFLINE function is used to "unready" a virtual device. For disks, it will clear volume valid, and for tapes, it will issue a rewind and unload QIO\$.

```
VCP>SET DR7:/OFF
```

### **/RON**

The ReadOnly function sets a device into read-only mode, by setting the software write lock bit in the device's unit characteristics word.

```
VCP>SET DR7:/RON
```

### **/RW**

The ReadWrite function sets a device into read/write mode, if it was disabled by software. This function cannot be used to enable a device which is hardware write protected.

```
VCP>SET MM4:/RW
```

### **/EOF**

The EOF function force an update of the file attributes for a virtual tape container file. This function will write the user file attributes area of the file, which will reflect the device characteristics at the time the command is issued.

```
VCP>SET MS2:/EOF
```

### **/EXtend**

The EXTend function is used to enable or disable extend access for a virtual tape. The switch allows an argument which specifies the number of blocks to be allocated when an extend function is required. By negating the switch, extend access is denied, and a task writing to the tape will receive an end-of-tape error when there are 5 blocks left in the file.

```
VCP>SET MM3:/EXT:2000
```

Set MM3: to allow extends, and when required, extend the file by 2000 blocks.

### **/ZEro**

The ZERO command is used to zero all counters associated with a device. These include both VF: specific counters, and RSX counters.

```
VCP>SET MS2:/ZE
```

### **/REwind**

The REWIND function will issue a rewind command to a specific magtape unit, return the device to <BOT>.

```
VCP>SET MS2:/REW
```

---

### 6.4.8 The STOp command (Privileged)

The STOP command terminates trace operations on a specific unit, or all active traced devices. The format of the command is:

**Format**

**STOP [device:][/ALL]**

where either "device:" or "/ALL" can be specified, depending on whether all trace activity is to be stopped, or only trace operations on a single unit.

---

### 6.4.9 The TRAcE command (Privileged)

The trace function allows I/O requests to be traced through the system, and recorded in a log file for later analysis. Within the trace facility are functions which can limit the number of packets logged, based upon the logical block referenced, the task issuing the I/O, the type of I/O function code, or the particular path taken by an I/O packet.

In determining whether a packet can be logged, the driver will create a packet which is interpreted by the trace logging task, and if any condition is satisfied for logging, the packet will be written to the logfile. If the logfile is written to a disk which has tracing enabled, the actual I/O issued by the trace task will not be logged in order to eliminate the possibility of an infinite logging loop.

Below are listed the areas which can be tailored to limit the total number of packets logged, and how they are validated within VCP.

- Log I/O generated by a specific task, or tasks
- Log I/O which is specified for a particular LBN, or range of LBN's
- Log I/O which is using a specific I/O function code.
- Log I/O which has a specific status attribute associated with the packet

---

#### 6.4.9.1 Tasks

When considering I/O which is issued by a specific task, the task name is compared against the task name which is stored within the trace packet, which is the task whose TCB pointer is contained in I.TCB of the I/O packet.

---

#### 6.4.9.2 Logical Block Numbers/Ranges

Specific logical block number ranges can be compared at the output level dispatching of the I/O packet from the driver. These LBN values will be the block numbers which will be directed to the next device. If, for example, a virtual device began at block 3 of a physical device, and the user wished to monitor I/O activity to LBN 0 of the virtual device, the trace function would need to specify that LBN 3 was the desired LBN to monitor at the virtual device level. If the connected device is setup as a passthru driver, the logical blocks on the virtual device would have a one to one correspondence with a block on the physical device.

A range of logical block number can be specified by specifying the ending LBN value as a negative number. To specify a block range of 0 through 100, the LBN switch would be specified as "/LBN:0:-100".

## Virtual Device Subsystem Reference

---

### 6.4.9.3 I/O function monitoring

The trace function allows for the monitoring of primary I/O function codes. These are the function codes defined through the high byte of the actual I/O function code. Examples of these "primary" I/O function codes are IO.KIL, IO.RLB, IO.WLB, IO.ATT, IO.DET, and IO.STC.

---

### 6.4.9.4 Status attribute monitoring

Status attribute monitoring allows observing I/O which has gone through a specific path before arriving at the virtual driver. Examples of the attributes which can be monitored are Internal I/O Completion packets, which can be generated by VFDRV, or the Data Cache Manager (DCM11M). Due to the nature of how VFDRV behaves, other internal functions of the driver can be monitored to determine if the driver is behaving incorrectly during a particular sequence of events, which otherwise would not be able to be observed.

The specific status attributes available are:

- 1 MDV, I/O which spans physical devices
- 2 SHA, I/O which is associated with a shadow pair
- 3 DUP, I/O which caused VFDRV to duplicate the I/O packet
- 4 INT, I/O which is nested more than one level deep
- 5 IIC, I/O which utilizes a Kernel AST completion routine
- 6 ADA, I/O which VFDRV uses I.AADA as internal pointers

---

### 6.4.9.5 Examples

Below are examples of the use of each of the TRACE switches listed above:

```
TRACE VF3:/TSK:XXXT4:F11ACP
```

```
TRACE DB4:/FN:WLB/TYP:IIC
```

```
TRACE VF5:/LBN:0:-5
```

```
TRACE VF:/APP/TYP:MDV
```

The first example enables trace operations for all requests posted by task XXXT4, and F11ACP. The second example will monitor all write functions, as well as all I/O packets which specify an Internal Completion routine (Kernel AST). The third example will trace all I/O directed to LBN 0 through 5 at the output from the virtual device. The last example will trace all I/O requests which were forced to be split by VFDRV, and the output will be appended to a previous logfile if one currently exists.

---

## 6.5 Technical Overview

This section describes the basic algorithms used to implement the virtual disk driver, and the protection mechanisms which allow the driver to insure data integrity of the system as a whole, and of the disk data structures.

---

### 6.5.1 General

In general, the driver uses the RSX-11M/M-Plus I/O hierarchy to allow a very flexible usage of devices in the system. Each virtual device accessed as a file on a previously mounted device uses the Files-11 File Control Block to preserve the data integrity of the device at each higher level. By removing the File window block from the task header at the time the file is accessed, this keeps the volumes transaction count at a non-zero level while the virtual device is accessed. Since the volume is considered to be an open file, the space on the disk cannot be deleted until the file is deaccessed, and effectively, the space is reserved for the driver.

For devices bound together as a volume set, each member of the set is redirected to the target device, and as such any I/O directed to the member volume will be handled by the virtual disk unit. If the volume is not mounted as passthru, the volume must be mounted foreign, since control I/O functions will be not passed on to each of the member volumes.

If this concept is followed down to multiple levels, each device which resides on a higher level device structure also has the space "reserved" at each level, and due to the transaction counts being maintained, none of the volumes in the hierarchy can be dismounted until each volume at a lower level is also released.

---

#### 6.5.1.1 Access rights, and privileges

In order to connect a volume to a virtual device, the user must have full access to the volume and file which is to be connected. If the user does not have the access, the file access will fail, and the connect request will be terminated. Once the user has connected the volume to a virtual device, the protection mechanism again returns to the normal device protection available from RSX-11M/M-Plus. If the volume is allocated, or mounted public, the user must be privileged to affect a system volume. Similarly, if a non-privileged user connects, and subsequently mounts the volume private, that user will have full access rights to the volume. In order to disconnect a volume from active use, it must be dismounted, and for a non-privileged user, this function will be reserved to volumes which were previously mounted by the user, and which are not public devices.

---

#### 6.5.1.2 Create Function

Since the function has consequences on a system wide basis, and can affect system primary pool, this function is reserved to privileged users. The basic function of the create command is to allow the system to dynamically expand the device data structures used by VFDRV. The driver utilizes a common Status Control Block (SCB) for all units created on the system, so the data structures which are created by this function are the additional DCB which will be used by the added devices, and the Unit Control Blocks (UCB's) which are specified by the argument of the command. If an optional device name is included in the create command, an additional DCB will be added which will reference the same set of unit control blocks under a different device name.

Alternate device DCB's will not be linked into the system's device list until the unit is actually connected. This allows flexibility in the actual handling of physical device drivers. If an alternate DCB existed prior to the actual loading of the physical device driver, the actual device driver would attempt to utilize the data structures created by VCP, which would not contain real

## Virtual Device Subsystem Reference

controller references. As such, the final result would be unpredictable, with a system crash being the most likely outcome.

All DCB's created by the utility will fall in ascending sequential order from the last DCB included in the system for the specific device type. In this case, the logical unit number of the alternate device name may not be the same as the logical unit number under which the unit is accessed as a VF: device.

---

### 6.5.2 Connect Function

The CONNECT function is the principal "work" function performed by the control program. Its purpose is to allow a virtual device to use a specific file as the target for the device operations. The connect function can also create the target file, and assign specific attributes to the virtual device in the process of the connect operation.

---

#### 6.5.2.1 File access

The first step in connecting the virtual device is to ensure that the file can be accessed for use. The user must have read, write, and extend access to the file in order to connect it.

---

#### 6.5.2.2 Device access

For merged devices, the target device must not be mounted Files-11, and not allocated to another user. If these conditions are not met, an error message will be generated. For RSX-11M-Plus, the volume must be mounted foreign.

When passthrough mode is selected, the device must not be mounted.

---

#### 6.5.2.3 Virtual Unit selection

After the file is accessed, the program searches the volatile data base to determine if there are any virtual units which can be used to access the file. The user may specify a specific unit to be used, or add a condition that the device be connected using a specific device name.

If the device selected is to be accessed using a device name other than VF:, the UCB must have the DCB back pointer redirected such that the device will appear to be the correct device name.

---

#### 6.5.2.4 File linkage to virtual device

At this point, the File Control Block Extension (FCBX) is allocated, and pointed to by the UCB pointer U.FCBX. Then the control word of the target device must be copied, such that if the device is an NPR device, the buffer words are set up correctly to allow the driver to transfer the data correctly. Finally, the Unit Control Block Extension is allocated, and the device type information is copied to the secondary pool extension.

Once the UCB data structures are set up, the UCB is set to an online state, and the driver is called at the power fail entry point, to ensure that it is ready to accept the subsequent I/O requests which will follow.

Host Reconfiguration task (HRC...) is notified of the change, if active, and the user is notified of the unit which was assigned for use.

---

### 6.5.3 Disk Data Caching, and Volume Shadowing

The driver handles both device shadowing, and disk data caching, in a way which incurs minimum overhead on the operating system. In both cases, the I/O packet is tracked by the UCB field of the I/O packet, which is generally translated to the target device UCB. Since this is not reasonable under these cases, both conditions are handled as a special case by the driver.

---

#### 6.5.3.1 Multiple Device Transfers

The driver is capable of breaking an individual task I/O request into multiple individual requests. Each I/O packet is queued to the actual physical device which will then perform the actual I/O, and returned to VFDRV after the function is complete. The packet totals are updated in the original I/O packet to reflect the actual number of bytes transferred, and if the byte count has not been satisfied, the next device in the volume set is used, and the base logical block number is reset to zero.

---

#### 6.5.3.2 Shadowing

Shadowing is implemented at the executive level, within the subroutine \$GTPKT. Within this subroutine, a data structure allocates a duplicate I/O packet, and points to the primary I/O packet from within this secondary data structure. The routine \$DRQRQ is then called recursively with the primary I/O packet.

The driver handles this situation by a) not allowing recursive entry into the driver, and b) making a duplicate I/O packet for the request. In the case of a write request, both packets are sent to their respective devices.

---

#### 6.5.3.3 Disk Data Caching

Disk data caching is implemented within the executive routine \$DRQRQ, where an I/O packet is checked against the blocks which are cached. If the packet is not found, and also meets the requirements to be cached, the I/O packet will be duplicated, and the duplicate packet will be forwarded on, with an internal I/O completion specified.

Since the cacher specifies an internal I/O completion routine, the driver must duplicate the I/O packet so that it can also specify an internal completion to unwind the UCB addresses associated with the specific device.

In addition, the driver maintains the maximum usability of cache by only allowing the top level request to be cached. All indirect references to different devices along the hierarchy cannot be cached.

---

### 6.5.4 Device Data Structures

This section describes the device data structures associated with the driver. Most entities are standard RSX-11M/M-Plus device structures, with a few additional data structures added. The UCB structure is extended backwards up to a maximum of 10. bytes. The data structures described are:

- Unit File Control Block Extension (U.FCBX)
- Unit I/O Outstanding Queue (U.IOPQ)

## Virtual Device Subsystem Reference

- Unit Device Control Block Pointers (U.DCBP,U.DCBS)

The definition which describes these offsets is:

```
.ASECT
= -12
U.DCBS: .BLKW 1 ; Secondary DCB pointer
U.DCBP: .BLKW 1 ; Primary DCB pointer
U.FCBX: .BLKW 1 ; File Control Block Extension
U.IOPQ: .BLKW 1 ; I/O Pending Queue
U.IOPQ: .BLKW 1 ; I/O Pending Queue
U.MUP: .BLKW 1 ; Multi-user Protection Word
U.DCB: .BLKW 1 ; Back pointer to DCB (=0)
```

Each of the structures pointed to be the UCB is described in the sections below. The offsets U.MUP, and U.DCB are standard offsets within the RSX device data structures. The offsets U.DCBS, and U.DCBP can only exist on the first UCB of the associated data structure, and all other units will have at most three negative offsets.

### U.FCBX

The Unit File Control Block Extension is used to describe the relationship of the device to the next higher level data structure, namely the target device on which the device lives. The File Control Block Extension (FCBX) is a 6 word structure, with the following structure:

```
.ASECT
=0
FX.UCB: .BLKW 1 ; Destination UCB
FX.WDB: .BLKW 1 ; File Window Block (@I.LN2)
FX.LBN: .BLKW 2 ; Starting LBN on device
FX.TCB: .BLKW 1 ; TCB address of trace task
FX.REL: .BLKW 1 ; Trace buffer logging bias
FX.SIZ: .BLKW 0 ; Size of this data structure
```

for merged devices, the list format is slightly different;

```
.ASECT
=0
FX.UCB: .BLKW 1 ; Destination UCB
FX.LNK: .BLKW 1 ; Link word to next device FCBX
.BLKB 1 ; Unused
FX.FLG: .BLKB 1 ; .GT. 0 if merged device list
.BLKW 1 ; Unused
FX.SZ0: .BLKW 0 ; Size of subsequent device FCBX's
FX.TCB: .BLKW 1 ; TCB address of trace task
FX.REL: .BLKW 1 ; Trace buffer logging bias
FX.SIZ: .BLKW 0 ; Size of this data structure
```

for tapes, the list offset include a bias value used to access the tape dynamic region:

## Virtual Device Subsystem Reference

```
.ASECT
=0
FX.UCB: .BLKW 1 ; Destination UCB
FX.WDB: .BLKW 1 ; File Window Block (@I.LN2)
        .BLKB 1 ; Unused byte
FX.FLG: .BLKB 1 ; Flag indicating file type
FX.ODB: .BLKW 1 ; Starting Bias of tape region
FX.TCB: .BLKW 1 ; TCB address of trace task
FX.REL: .BLKW 1 ; Trace buffer logging bias
FX.SIZ: .BLKW 0 ; Size of this data structure
```

and for memory regions, the following offsets describe access to the dynamic region:

```
.ASECT
=0
FX.UCB: .BLKW 1 ; Destination UCB (=0 for memory)
FX.PCB: .BLKW 1 ; PCB address of memory partition
FX.PTY: .BLKW 1 ; Address of ICB parity block
FX.FRK: .BLKW 1 ; Address of possible fork block
FX.TCB: .BLKW 1 ; TCB address of trace task
FX.REL: .BLKW 1 ; Trace buffer logging bias
FX.SIZ: .BLKW 0 ; Size of this data structure
```

### **FX.UCB (Destination UCB)**

The UCB field of this structure describes the target unit on which the virtual device actually lives. This device may be a physical device or another virtual device. The I/O packet will be forwarded to this device, after adjusting the destination LBN to be in line with the starting LBN of this virtual unit mapped to the target UCB.

For memory regions, this field is set to zero.

### **FX.WDB (Window Block)**

The window block is the mechanism whereby the driver locks the virtual device to the next level device structure. This is the window block which is created when VCP accesses the file during the connect operation. The format of the data structure is that of a Files-11 Window Block, described in the system macro F11DF\$.

### **FX.PCB (Region PCB address)**

The PCB address is utilized for memory regions to describe the partition which is being utilized as the disk region. This PCB is then linked into the appropriate main partition to reserve the memory for use as a memory disk.

### **FX.LNK (FCBX Link word)**

For bound volume sets, the link word serves to point to the next FCBX data structure associated with the device. The end of the list is terminated with a zero link word.

## Virtual Device Subsystem Reference

### **FX.LBN (Starting LBN)**

The starting LBN is that value which is the address of virtual block 1 of the file accessed when the unit was connected. This 24-bit base LBN is added to the logical block values stored in parameters four and five of the QIO\$ directive parameter block passed to the driver when the QIO\$ directive is issued.

### **FX.FLG (Flags byte)**

The flags byte overlaps the high order byte of the starting logical block number. Since the logical block cannot exceed 24 bits, this flags byte will be zero for a volume which is actually a file.

### **FX.ODB (Object descriptor tracking area)**

The object descriptor bias is used by the driver to handle current tape context, and current object information. For RSX-11M-Plus systems, this region is called VF.TAP, and is loaded in high memory when the tape device is actually connected.

### **FX.SZ0 (Secondary FCBX size)**

FX.SZ0 represents the number of bytes which will be used for all subsequent FCBX's associated with a merged volume set. Only the primary FCBX will maintain the data structures associated with trace operations.

### **FX.TCB (Trace task TCB)**

This word is used to indicate the task which is currently associated with trace operations. The driver will verify that the task is active before attempting to allocate a trace buffer for use. If the last buffer is taken, the driver will use this value to flag the associated unit as attached, which will stall all other I/O requests.

### **FX.REL (Trace buffer bias)**

This word contains the starting 32.wd block bias which contains the actual trace buffers used for trace logging. The first 32.wd block is used to maintain listheads and counters, and all other 64.byte blocks are used for containing the actual traced I/O packets.

---

## **6.5.5 U.DCBP,U.DCBS**

The offsets are defined only for the first unit of a multi-accessed unit, and point to the two Device Control Blocks (DCB's) possible for these units. In order for this data structure to exist, the status bit UU.MUN is set in the utility word (U.UTIL) of the UCB. These words allow VCP to determine if the DCB is in use, or that the DCB is available for that unit.

---

## **6.5.6 U.IOPQ**

The Unit I/O pending Queue is used to be able to determine what I/O packets are currently queued to other drivers (including VFDRV). The queue is only used when one of the following conditions are in effect:

- The unit is being shadowed

## Virtual Device Subsystem Reference

- The unit has an I/O packet which has an internal I/O completion queued
- The packet crosses actual physical device boundaries

The internal I/O completion function is currently used to implement disk data caching, where the UCB field of the I/O packet is required to be correct for the I/O function. When an I/O packet is defined to be one of the above functions, the driver will either a) Duplicate the I/O packet to set up an internal I/O completion function itself, or b) attach a rider packet to a previously duplicated packet created by the driver. The "packet" is then attached to the newly created I/O packet through the I.AADA packet offset, and the current UCB address is placed into offset I.AADA+2. These packets allow for a normal "unwinding" of the I/O packet during the I/O Done processing.

The offsets defined for this use are:

```
.ASECT
      = I.AADA
I.ILNK: .BLKW  1          ; I/O Rider link
I.IUCB: .BLKW  1          ; Next level UCB to complete
      = 0
R.ILNK: .BLKW  1          ; Next rider link
R.IUCB: .BLKW  1          ; Next level UCB address
```

When the rider link word (R.ILNK) field becomes zero, then the rider packet is actually the original I/O packet queued to VFDRV at the first level.

At that point, the duplicated I/O packet is released back to primary pool, and the completion status is passed back through \$IODON, to return the final status to the caller.



# 7

---

## Extended Security and Related Features

The following features have been added to the Account File Maintenance Utility (ACNT), the Login task (HELLO) and the Set Password facility (PSW).

- Password lifetime
- Password minimum length
- Inactivity logout timer
- Inactivity warning message initial time remaining
- Inactivity warning message time interval
- Password expiration
- Disable account
- Remote dialup login disabled
- Network login disabled
- Account expiration

Each of these functions is described in the sections below.

Also described is an enhancement to the Last Login message to provide a foreign language and alternate format date display and also enhancements to BYE to optionally give "silent" logouts.

---

### 7.1

#### Account File Maintenance Utility (ACNT) changes

Along with the functional changes described in the following sections, the Account File Maintenance Utility (ACNT) dialogue has been changed when prompting for the entry of an account in the Delete, Examine and Modify functions.

For Delete and Examine, either the UIC (Group,Member) or the user's Last Name can be entered to find the account. The user is now prompted with the following message:

```
Enter account (group,member, or last name) :
```

For Modify, the UIC, Last name, or the keyword ALL can be specified as a response to the prompt:

```
Enter account (group,member, last name or ALL) :
```

The keyword ALL is used to change those parameters which can be universally applied to all accounts, such as Password Lifetime.

---

### 7.2 Password Expiration

Password Expiration is an ACNT parameter which will force selected users to change their password at their next login. It is settable on a per-user basis (during the ACNT Add or Modify Account functions) or for all users (using the ACNT Modify Account function). If the password expired flag is set, when the user next logs in, he will be prompted to change his password as a part of the login process.

**Note: If the user is logging into a batch job or virtual terminal, password expiration will not be enforced since there are no means available to change the user's password from a batch job.**

The normal use for this feature is to allow the System Manager to specify the initial password for a user when an account is set up for convenience, and then allow the user to change his password.

---

### 7.3 Password Lifetime

Password Lifetime is an ACNT parameter which will force users to change their password an interval determined by the system manager. It is settable on a per-user basis (during the ACNT Add or Modify Account functions) or for all users (using the ACNT Modify Account function). When the password lifetime has date has been reached or passed, the password expired flag is set and the user is forced to change his password at his next login.

**Note: If the user is logging into a batch job or virtual terminal, password expiration will not be enforced since there are no means available to change the user's password from a batch job.**

The password lifetime value is settable in a range from zero to one hundred-twenty seven days (0-127) inclusive. A value of zero will disable password lifetime checking. For all other valid values, the date of the user's last password change will be used to calculate the new expiry date.

As the expiration date approaches, the user will be warned that the password will expire at an interval defined system wide. If that interval is less than the password lifetime for an account, warnings will not be given for that account. The parameter may be selected either as a task build parameter when building the Login (HELLO) task or as a system defined logical, "SYS\$PWD\_WARN". The value should be a decimal number in the range 0 - 127, inclusive.

For example, to define the logical for a password warning interval of 7 days:

```
>DFL SYS$PWD_WARN=7/GBL  
or  
$ DEFINE/SYSTEM SYS$PWD_WARN 7
```

The task build command file, HELBLD.TKB, documents how to set this parameter during taskbuild.

Below is an example of the use of the "MODIFY" option, where all accounts are modified to select password lifetime, followed by examination of an account:

## Extended Security and Related Features

```
Enter option: M
Enter account (group,member, last name or ALL): ALL
Slave terminal? [Y/N]: N
Disable login/logout messages? [Y/N]: N
Password lifetime [D R:0-255 0=Disable]: 60
Password minimum length [D R:0-39 0=Disable]: 9
Inactivity logout timer [D R:0-32767 0=Disable]:
Warning message initial time remaining [D R:1-255 D:10]:
Warning message time interval [D R:1-255 D:5]:
Expire password? [Y/N]: Y
Disable account from login? [Y/N]:
Remote dialup login disabled? [Y/N]:
Network login disabled? [Y/N]:
Enter account (group,member): ^=Z
Account Utility options are:
  Add  Add an account to file
  Delete Delete an account file entry
  Examine Examine existing account
  List  List account file
  Modify Modify account file
  Sort  Sort account file
  CTRL/Z Terminate utility session
Enter option: E
Enter account (group,member, or last name): 1,174
14-MAR-91 20:59:06  RSX-11M-PLUS Multiuser Account File List  Page 1
Owner = [001,174] Login_defaults = US00:[SMITH]
L_name = SMITH  F_name = JOHN  Password = (ENCRYPTED)
Def_CLI = MCR  Session ID = USR Account # = 4397
Total_logins = 13966 Last_login = 14-MAR-91 18:48:18, Login_failures = 0
Characteristics = NOSlave  Def_dir_string  NOSilent  Pwd_min = 9
Login_Flags = Pwd_expired
Password_lifetime = 60 days
Def_Protection = [RWED,RWED,RWE,]
Inactivity timer = 240 min., Warning time/interval = 10./5. min.
Key RETURN to continue, CTRL/Z for ACNT Options Menu :
```

---

### 7.4 Password Minimum Length

Password Minimum Length is an ACNT parameter which will force users to specify a password of a minimum number of characters. It is settable on a per-user basis (during the ACNT Add or Modify Account functions) or for all users (using the ACNT Modify Account function).

When a user attempts to change his password with the "SET PASSWORD" command (the PSW Utility) the new password will be checked for length and if less than the specified minimum, the user will be notified and requested to try again.

If the new password is not long enough, the following example displays the result:

```
$ SET PASSWORD
Old password:
New password:
Verification:
SET -- Password must be between 9 and 39 characters - retry
$
```

## Extended Security and Related Features

---

### 7.5 Disabling Account Access

Access to the system for a particular account can be disabled, either completely, or from selected sources. These functions are enabled from the ACNT utility and require a Yes or No response:

- Disable account
- Remote dialup login disabled
- Network login disabled

On an attempted Login (HELLO), each of these is checked and access allowed or denied as appropriate.

---

#### 7.5.1 Account Disabled

When an account is flagged as disabled, all logins through that account are denied. This is true for batch, interactive, and network logins.

---

#### 7.5.2 Remote Login Disabled

When an account is flagged with the remote login disabled bit, all attempts to login to the system from a device which is set "REMOTE" will be disabled.

A remote device is one which is set with the "REMOTE" terminal characteristic set on the system (typically a modem dial-up port). To see which ports are set remote, enter the following command:

```
>SET /REMOTE
```

**Note: All Local Area Transport (LAT) terminals are classed as remote, so users who access the system using LAT terminals must not have Remote Access denied.**

---

#### 7.5.3 Network Login Disabled

When a user has Network Login disabled, access will be denied when attempting to log in from any of the following sources:

- DECnet RT: devices
- DECnet HT: devices
- Any DECnet application which has verification enabled; i.e. FAL

**Note: Support to completely enable this feature is in NVP included with DECnet-11M+ Version 4.6. For systems with earlier versions of DECnet, only SET HOST type logins will be disabled, and all other access will still be permitted.**

## 7.5.4 Account expiration

Account expiration is an ACNT parameter settable on a per-user basis, and When an expiration date is defined, the Examine or List ACNT functions will display the expiration date, otherwise the entry will be omitted. The expiration date is the last day that access to the account will be permitted. No access will be permitted on the following day.

Below is an example which sets up an account for expiration on 13-MAR-1991, and the subsequent "EXAMINE" command for the account.

```
>RUN $ACNT

*** RSX-11M-PLUS Account File Maintenance Utility ***
Account Utility options are:
  Add Add an account to file
  Delete Delete an account file entry
  Examine Examine existing account
  List List account file
  Modify Modify account file
  Sort Sort account file
  CTRL/Z Terminate utility session

Enter option: M

Enter account (group,member, last name or ALL): 1,174

Owner = [001,174] Login_defaults = US00:[SMITH]
L_name = SMITH      F_name = JOHN      Password = (ENCRYPTED)
Def_CLI = MCR      Session ID = USR Account # = 4397
Total_logins = 12483 Last_login = 18-MAR-91 16:02:40, Login_failures = 0
Characteristics = NOSlave      Def_dir_string NOSilent Pwd_min = 9
Login_Flags = None
Password_lifetime = 30 days, Date_last_changed = 14-MAR-91
Def_Protection = [RWED,RWED,RWE,]
Inactivity timer = 240 min., Warning time/interval = 10./5. min.

This entry? [Y/N]: Y
Type <ESC> or <=><RET> to leave entry unchanged
Password [S R:0-39]: =
Default system device ( DDU ): =
First name [S R:0-12]: =
Last name [S R:0-14]: =
Default file protection ([SY,OW,GR,WO]): =
Default directory string ([group,member] or [name]) : =
Session ident [S R:0-3]: =
Account number [D R:0-9999]: =
Enter user CLI [S R:1-6 D:"MCR"] : =
Account expiration date (MM/DD/YY or DD-MMM-YY) <CR>=None: 3/13/91
Slave terminal? [Y/N]:
Disable login/logout messages? [Y/N]:
Password lifetime [D R:0-255 0=Disable]: =
Password minimum length [D R:0-39 0=Disable]: =
Inactivity logout timer [D R:0-32767 0=Disable]: =
Expire password? [Y/N]: =
Disable account from login? [Y/N]: =
Remote dialup login disabled? [Y/N]: =
Network login disabled? [Y/N]: =

Enter account (group,member): ^Z
```

## Extended Security and Related Features

```
Account Utility options are:
  Add Add an account to file
  Delete Delete an account file entry
  Examine Examine existing account
  List List account file
  Modify Modify account file
  Sort Sort account file
  CTRL/Z Terminate utility session

Enter option: E

Enter account (group,member, or last name): 1,174

18-MAR-91 16:04:11  RSX-11M-PLUS Multiuser Account File List  Page 1

Owner = [001,174] Login_defaults = US00:[SMITH]
L_name = SMITH      F_name = JOHN      Password = (ENCRYPTED)
Def_CLI = MCR      Session ID = USR Account # = 4397
Total_logins = 12483 Last_login = 18-MAR-91 16:02:40, Login_failures = 0
Characteristics = NOSlave      Def_dir_string  NOSilent  Pwd_min = 6
Login_Flags = None
Password_lifetime = 30 days, Date_last_changed = 14-MAR-91
Def_Protection = [RWED,RWED,RWE,]
Inactivity timer = 240 min., Warning time/interval = 10./5. min.
Account expiration date = 13-MAR-91
```

---

## 7.6 System Inactivity Logout

System Inactivity Logout is enabled for individual or all users in ACNT. System inactivity or Idle Session is checked by the system task SYSLOG. It will scan all active users once a minute, provide warning messages, and force a logout for a user who has this parameter enabled.

The task checks the number of directives issued by a user since the last scan, and if unchanged, it declares that user "idle" for the previous minute. When the user has been idle for the specified number of minutes, a logout will be forced.

**Caution:** For those familiar with OpenVMS inactivity logouts, the RSX inactivity logout mechanism differs in one very specific way. Under Open VMS, if a user enters a carriage return, it will constitute system activity. Input solely to MCR, the RSX command dispatcher, does not appear to the system as if the user has been active. A carriage return at the terminal falls into this category.

---

### 7.6.1 Parameters

There are three parameters associated with inactivity logout:

- total inactive time allowed
- number of minutes before logout before the initial warning is displayed
- timer interval at which logout pending warnings are issued

These parameters are selected at the following ACNT utility prompts:

- Inactivity logout timer [D R:0-32767 0=Disable]:
- Warning message initial time remaining [D R:1-255 D:10]:
- Warning message time interval [D R:1-255 D:5]:

---

### 7.6.1.1 Inactivity logout timer

This value specifies the total time that a user is allowed to be inactive prior to the user being logged off. The value given is specified in minutes, with a value of zero being used to disable inactivity logout. The valid range for this parameter is from 0 - 32767, inclusive.

---

### 7.6.1.2 Warning message initial time remaining

This value specifies, in minutes, when the user should begin to receive warning messages prior to logout. Once the initial time remaining value has been reached, the inactive user will continue to receive warning messages at the interval defined as the "Warning message time interval". The valid range for this parameter is from 1 - 255, inclusive.

---

### 7.6.1.3 Warning message time interval

Once the initial time remaining value has been reached, the user will receive warnings at timed intervals based on this value until the total time allowed has expired. The valid range for this parameter is from 1 - 255, inclusive.

---

## 7.6.2 Messages

When SYSLOG issues messages to a user, it will disable the following terminal characteristics in an attempt to "breakthrough" any current activity:

- Cancel control-O
- Cancel control-S
- Issue I/O request using the "write breakthrough" function code

When the initial time has elapsed the user will be warned that they have been inactive and will be logged out after remaining time has elapsed.

After the first message is displayed, the user will again be warned that they have been inactive and will continue to be warned according to the message interval timer.

---

## 7.6.3 Other conditions

If the user initiates a "SET HOST" session to a remote node, the task which controls the user's activity is active at the console pseudo-device (CO0:), so the activity will not be charged to the user's terminal. In order to prevent an unintended logout, any terminal which is attached to a task running from the console pseudo-device will not have inactivity checked.

If system logins are disabled, no system inactivity checking is also disabled for all users.

### 7.7 Login Failure Disconnect

The login task, HELLO, will disconnect a session which was connected from a remote line after encountering a specific number of login failures. This applies only to terminals which have the "REMOTE" characteristic enabled. If the terminal is connected using a modem on an asynchronous controlled, all modem control signals must be enabled to to disconnect the phone line when data terminal ready (DTR) is deasserted by the RSX host system.

The number of login failures allowed is a system-wide parameter, which is set either in the login task (HELLO) task build command file or as a system defined logical.

In the following example the logical is defined to disconnect remote sessions after 3 login failures.

```
>DFL SYS$HEL_HANGUP=3/GBL  
or  
$ DEFINE/SYSTEM SYS$HEL_HANGUP 3
```

The task build command file, HELBLD.TKB, also documents how this parameter is selected at taskbuild time.

#### 7.7.1 Login Date Display Options

This enhancement allows the day of the week and the month in Login's "Last interactive login" message to be displayed in any selected language. Since the month and date are usually transposed outside the USA, when this option is selected the message will be displayed, for example, like this:

```
Last interactive login on Mardi, 25 Avril,1995 14:27:26 (TT20:)
```

To change the names of the Days, define the logicals SYS\$DAY\_0 through SYS\$DAY\_6 starting with Sunday.

```
DEF/SYS "SYS$DAY_0" "Dimanche"  
DEF/SYS "SYS$DAY_1" "Lundi"  
...  
DEF/SYS "SYS$DAY_6" "Samedi"
```

To change the names of the Months, define the logicals SYS\$MONTH\_01 through SYS\$MONTH\_12 starting with January. Note that SYS\$MONTH is from the Alternate Date routines (See the System Library Routines Manual)

```
DEF/SYS "SYS$MONTH_01" "Janvier"  
DEF/SYS "SYS$MONTH_02" "Fevrier"  
...  
DEF/SYS "SYS$MONTH_12" "Decembre"
```

Defining the logical SYS\$MONTH\_01 will change the display of the Last login message to use the alternate date format. So, for example, in the UK and Canada, a single logical SYS\$MONTH\_01 could be defined to January, which would change the date presentation to the more generally accepted Day-Month-Year in those countries.

Note that the SYS\$DAY\_n definitions only apply to Login messages.

---

### 7.7.2 BYE Enhancements

BYE now has an additional switch. Its two switches are described below.

BYE [/S][/H]

/S[ilent] will stop the display on the user's terminal of much of the information output during a BYE operation.

/H[old] will hold the connection on a non-local line. This is useful when wanting to login again to the same system after logging off.

The DCL LOGOUT command has corresponding qualifiers:

LOGOUT [/SILENT][/HOLD]



# 8

---

## Year 2000 Certification for RSX-11M-PLUS Version 4.6

RSX-11M-PLUS Version 4.6 was released in January of 1999, and has been thoroughly tested for Year 2000 Compliance as described in this chapter.

RSX-11M-PLUS Version 4.6 V4.6, as tested, is compliant with both ISO 8601:1988 and ANSI X3.30 standards with regard to date presentation, when configured for ISO compliant date presentation.

RSX-11M-PLUS Version 4.6 is also compliant with the BSI-DISC PD2000-1 standard with regard to date handling.

Based on DISC PD2000-1, when years are specified as two-digits, the century inferencing rules will specify that years greater than seventy-seven (77) will be in the twentieth century (preceded by 19), and those less than sixty-four (64) will be in the twenty-first century (preceded by 20).

All date related functions provided by the operating system have been extensively tested to insure compliance with the BSI, ISO, and ANSI standards, through the year 2064.

---

### 8.1 Date Presentation and Input

All dates are presented by RSX-11M-PLUS Version 4.6, its components and utilities in one of the following 3 formats based on a selection at system generation, or dynamically changed with the System Reconfiguration Utility:

DD-MMM-YY      2 Digit Format

DD-MMM-YYYY    4 Digit Format

YYYY-MM-DD     ISO Format

All components of RSX-11M-PLUS Version 4.6 will accept dates in any of the three defined formats. Dates entered in two digit format are subject to inferencing. (See the section following titled *RSX and Inferencing*)

---

### 8.2 RSX and Inferencing

In the discussions of Year 2000, the term inferencing occurs widely in discussing 2 digit year formats. When a date is input to RSX-11M-PLUS Version 4.6 with a 2 digit year, it is necessary to determine a full 4 digit year that this 2 digit year is intended to represent. This process is referred to as inferencing. Defined inferencing does meet the international standards.

On input, RSX-11M-PLUS Version 4.6, as shipped, will infer 2 digit years 65 through 99 as 1965 through 1999, and 2 digit years 00 through 64 as 2000 through 2064. It is possible to change the base year of 1965 (described later.)

On output, RSX-11M-PLUS Version 4.6 will obey the same rules when outputting 2 digit dates, and the correct century can be inferred on sight.

### 8.3 Internal Date Storage

Internally, the date is stored in a data structure which includes both time and date. This structure is normally retrieved by the use of the Executive Directive GTIM\$ or application level function which invokes this directive.

Shown below is the format of this structure, which is common to all members of the RSX family.

```

      15                               00
+-----+
!           Year - 1900           !
+-----+
!           Month                 !
+-----+
!           Day                   !
+-----+
!           Hour                  !
+-----+
!           Minute                !
+-----+
!           Second                !
+-----+
!           Tick of Second        !
+-----+
!           Ticks per Second     !
+-----+

```

By the use of a single common executive time directive, this structure is used internally by all RSX components referencing the system date, except where otherwise noted.

### 8.4 Disk File System

The RSX Family uses Files-11 Structure level 1. This volume structure contains an index file, which contains file headers for all files on the volume. The file header contains all of the information relating to the file's creation date, revision date, and expiration date. RSX-11M-PLUS does not utilize the file expiration date for any purpose.

#### 8.4.1 Files-11 File Headers

The file header block is organized into four areas, of which the first three are variable in size.

- Header Area
- Identification Area
- Map Area
- Checksum

The Identification Area includes the applicable date fields.

**8.4.1.1 ODS-I File Header Identification Area Description**

The ident area of the file header begins at the word indicated by H.IDOF. It contains identification and accounting data about the file.

**Table 8-1 Identification Area**

Offset	Size	Description
I.FNAM	6 Bytes	File Name, These three words contain the name of the file, packed three Radix-50 characters to the word. This name usually, but not necessarily, corresponds to the name of the file's primary directory entry.
I.FTYP	2 Bytes	File Type, This word contains the type of the file in the form of three Radix-50 characters.
I.FVER	2 Bytes	Version Number, This word contains the version number of the file in binary form.
I.RVNO	2 Bytes	Revision Number, This word contains the revision count of the file. The revision count is the number of times the file has been accessed for write.
I.RVDT	7 Bytes	Revision Date, The revision date is the date on which the file was last deaccessed after being accessed for write. It is stored in ASCII in the form "DDMMYY", where DD is two digits representing the day of the month, MMM is three characters representing the month, and YY is the last two digits of the year.
I.RVTI	6 Bytes	Revision Time, The revision time is the time of day on which the file was last deaccessed after being accessed for write. It is stored in ASCII in the format "HHMMSS", where HH is the hour, MM is the minute, and SS is the second.
I.CRDT	7 Bytes	Creation Date, These seven bytes contain the date on which the file was created. The format is the same as that of the revision date above.
I.CRTI	6 Bytes	Creation Time, These six bytes contain the time of day at which the file was created. The format is the same as that of the revision time above.
I.EXDT	7 Bytes	Expiration Date, These seven bytes contain the date on which the file becomes eligible to be deleted. The format is the same as that of the revision and creation dates above.
	1 Byte	(unused) This unused byte is present to round up the size if the ident area to a word boundary.
S.IDHD	46 Bytes	Size of Ident Area This symbol represents the size of the ident area containing all of the above entries.

## Year 2000 Certification for RSX-11M-PLUS Version 4.6

### 8.4.1.1.1 Date format to support year 2000 and beyond

Files-11 only allows for dates to be presented as "DDMMYY", which presents a problem for files created or revised after the year 2000.

ODS-1 provides a 7-byte ASCII field for creation and other dates. In order to accommodate this restriction in a way that is consistent with RSX date handling, the 2-bytes of year in the date attribute will be used to represent the year since 1900.

The two bytes will be encoded as follows:

- Low byte—Low order decimal digit of year, in ASCII.
- High byte—Quotient of years since 1900 divided by 10, plus 60<sub>8</sub>. This represents the ASCII high digit of the year from 1900 – 1999. In 2000, however, this will result in ":" being stored for the high digit of the year. For years 2010 through 2019, ";" will be stored, and so forth. This change should result in another 200 years or more of file representations.

**Table 8-2 Storage of Date Fields after 1990**

Calendar Year	Year Field Representation
1990 – 1991	90 – 99
2000 – 2009	:0 – :9
2010 – 2019	;0 – ;9
2020 – 2029	<0 – <9
2030 – 2039	=0 – =9
2040 – 2049	>0 – >9
2050 – 2059	?0 – ?9
2060 – 2069	@0 – @9
2070 – 2079	A0 – A9
2080 – 2089	B0 – B9
2090 – 2099	C0 – C9
2100 – 2109	D0 – D9
2110 – 2119	E0 – E9
2120 – 2129	F0 – F9
2130 – 2139	G0 – G9
2140 – 2149	H0 – H9
2150 – 2159	I0 – I9

### 8.4.2 File system attributes

The following sections describe how application interface with the Files-11 Ancillary Control Processor (F11ACP). In most cases, the system provides date information in the format described in the previous section.

In some cases, the information returned is formatted in a format which is compatible with the OpenVMS operating system, and is a 64-bit quantity expressed in 100ns intervals since 17–November–1858. Within this document, this quantity is referred to as a VMS quadword.

## Year 2000 Certification for RSX-11M-PLUS Version 4.6

The table below describes the various date related functions processed by F11ACP, and the type of date data which is returned:

**Table 8-3 F11ACP Date attributes**

Attribute code	Return Field(s)	Format
8.	File expiration date	DDMMYY
10.	Full file header	DDMMYY
13.	Creation/Revision date	DDMMYY
17.	Creation date	VMS quadword
18.	Revision date	VMS quadword
19.	Expiration date	VMS quadword

Both the ASCII date format, and VMS quadword format will support file dates well beyond 2100.

## 8.5 Magnetic Tape Standards

Date information is stored within the ANSI Header 1 label (HDR1), to identify the creation date of a given file. This date is specified to be of the form `_yyddd`, where `_` represents a blank, `yy` represents a year within the 1900s, and `ddd` represents a day of that year. Each digit is stored as a single ASCII digit `"0-9"`. This format conforms with a level 3 implementation of the ANSI standard (X3.27-1978). With the level 4 standard the `_` character is a blank for a year within the 1900s or a `"0"` for a year in the 2000s.

RSX-11M-PLUS Version 4.6 includes support for ANSI X3.27-1978 level 4 magnetic tape labels. This includes full support for the Year 2000, although the same ASCII attributes exist for accessing the date fields stored in the HDR1 label on tape.

MTAACP accepts the same date attributes as F11ACP, except for the revision date. MTAACP supports dates from 1900 through 2099.

RSX-11M-PLUS Version 4.6 implements ANSI compliant level 4 records for all the applicable utilities tested. This insures that all media transfers will be compatible with other systems implementing the ANSI level 4 standard.

**Table 8-4 RSX-11M-PLUS ANSI tape dates**

Date	ANSI
10-JAN-1998	_98010
10-JAN-2000	000010
10-JAN-2010	010010

---

## 8.6 Foreign System Interchange

The FLX utility is the only RSX component which deals with foreign system interchange.

The FLX utility will handle DOS, and RT-11 volumes. For DOS, the valid date range is 1970 through 2035, and for RT-11 volumes, the valid date range is 1972 through 2099.

For both file systems, the entire valid date range is supported via FLX, with no restrictions.

---

## 8.7 Backup Utilities

Three utilities are supplied with RSX-11M-PLUS for the purpose of backup. These are RMSBCK, RMSRST, and BRU. Each of these utilities will properly handle the full supported date range, and include support for century inferencing. When a year is specified as two digits, the inferred date range will be from 1965 through 2064. Four digit years are also supported, and must be specified as a value greater than 1900.

RMSBCK, and RMSRST also properly handle the full supported date range, from 1900 through 2064 inclusive. Internally, both utilities are limited to 2155, although extensive testing has not been done beyond 2099 on RSX-11M-PLUS.

All of the backup utilities also support ISO format dates as input for date range qualifiers and switches.

---

## 8.8 Networking

The network architecture of DECnet, includes two specifications which limit the overall supported dates for various operating systems. These are Network Management Protocol (V4.2.0), and Data Access Protocol (V7.1).

---

### 8.8.1 Data Access Protocol V7.1

The Data Access Protocol (DAP) used for file transfers, includes a file date and time (DTM) message. The DTM message stores the date as "DDMMMYY", and only includes two ASCII digits for the year. The date field is followed by a time field, in a similar ASCII format which does not have any limitations.

This message has three sub-groups, which, if present, are indicated in a menu byte. These are the creation date, revision date, and expiration date. Each of these date groups has the same "DDMMMYY" formatted date.

Since only two digits are used, century inferencing must be applied to the dates provided. All two digit fields in the range 70 through 99 are treated as being in the twentieth century, while those in the range 00 through 69 are treated as being in the twenty-first century, resulting in a valid date range of 1970 through 2069.

---

## 8.8.2 Network Management Protocol V4.2.0

Network Management protocol used for logging DECnet events, includes a date and time stamp, which is calculated as Julian half-days since 1-Jan-1977. The formula for the conversion is:

$$\begin{aligned} \text{JULIAN} = & \quad (3055 * (\text{MONTH} + 2) / 100 - (\text{MONTH} + 10) / 13 * 2 - 91 \\ & \quad + (1 - (\text{YEAR} - \text{YEAR} / 4 * 4 + 3) / 4) * (\text{MONTH} + 10) / 13 + \text{DAY} - 1 \\ & \quad + (\text{YEAR} - 1977) * 365 + (\text{YEAR} - 1977) / 4 * 2 \end{aligned}$$

If the resulting value is considered a signed 16-bit integer, the date range available is from 1-January-1977 through 9-November-2021, or 44 years.

If the resulting value is considered to be unsigned, the date range can be expanded to 2065.

The RSX-Family treats the resulting value as unsigned, and will support network event logging until 2065.

Once the limiting date is passed, events will continue to be logged, however the time stamp will rollback to 1977, and will be relative to that year.

---

## 8.9 File System Utilities

All file system utilities have been extensively tested for Year 2000 Readiness, and no issues have been identified over the supported range of dates, 1977 through 2064.

---

### 8.9.1 Utility Date Format

All utilities accept dates in one of the format(s) described below:

- DD-*MMM*-YY
- DD-*MMM*-YYYY
- YYYY-MM-DD, ISO 8601 Compliant

Field	Description
DD	Two-digit day of the month
MM	Two-digit month of year
MMM	Three character English abbreviation of the month of the year; i.e. JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC
YY	Two-digit year inferred to span between 1965-2064
YYYY	Four-digit year in the range 1900-2155

---

## 8.10 Assemblers, Compilers, and Linkers

The Macro-11 assembler has been corrected with regard to date presentation.

Affected are the output list, map, and cross reference files produced by the assembler and task builder. The MACRO assembler considers the year 1900 to indicate that there is no valid date in the system, and will not output a date if the current system date is set to 1900.

## Year 2000 Certification for RSX-11M-PLUS Version 4.6

The task creation date which is stored in the task label block is stored in a binary format, and the year is stored relative to 1900, and is identical to the internal system time format used by RSX-11M/11M-PLUS.

PDP-11 Layered Products are subject to separate Y2K certification. The list and details of the products updated and tested for use with V4.6 are listed in Chapter 5, along with brief notes on the changes made to the product.

---

### 8.11 Batch and Queue

The RSX-11M-PLUS Queue Manager and Command Line Interface can correctly handle dates in the range from 1977 through 2155, although it considers the year 2100 to be a leap year.

---

#### 8.11.1 Queue Manager Processing Options

In RSX-11M-PLUS V4.4, the structure for the Queue Manager was changed, with regard to date processing. Below is a description of the changes which were incorporated, and how this may affect applications.

Prior to RSX-11M-PLUS V4.4, the Queue Manager and spooling tasks (QMGCLI) provided 4 bits for storing the year specification in a PRINT/AFTER job. Also, the data structures passed between the Queue Manager and QMGCLI that were stored in queue file QUEUE.SYS provided 4 bits for storing year specification in PRINT/AFTER jobs. The information has been stored as an increment from 1977.

Now, the year specification is stored as an 8-bit value, as an increment from 1900. Along with this change, the storage of month, day, hour, and minute have been changed.

Prior to starting the new Queue Manager, the system manager should assure that the old QUEUE.SYS file is empty, that is, the results of a SHOW/QUE instruction shows all queues empty of held, pending, stopped, paused, etc. jobs. A new QUEUE.SYS file will be created by the Queue Manager when it starts.

Although the spooler tasks included with this version have been modified in accordance with the new formats, application programs may include spoolers that have not been updated. The former syntax for packets from a spooler task to the Queue Manager remains acceptable, although certain assumptions are made.

If a packet is received with function QM.OPJ (former Open Job function) or QM.MDJ (former Modify Job function), the Queue Manager modifies the packet to the new format before storing it in QUEUE.SYS. The former 4-bit Year increment value is assumed to be a modulo-16 delta from 1977, specifying a date subsequent to current system time. This is implemented by adding 16 to the increment until the year is greater than or equal to that of the current system time.

Any tasks that directly access the Job Entry structure in QUEUE.SYS will have to be modified accordingly. Symbolic offset names are maintained so, unless the task accesses the After Time or Job Time Stamp fields, they probably only require you to assemble and link them again. Tasks that do make use of the Time fields should be recoded for consistency with the new format.

---

## 8.12 System Management

All of the RSX-11M-PLUS Version 4.6 System Management Utilities have been updated to include the three complaint date formats, in particular, in the error logging subsystem, and the system date and time utilities.

For volumes initialized after 2000, the volume creation date is stored and interpreted as described for other date fields used in ODS-I. This is also true for the volume revision date.

When using RSX-11M-PLUS Version 4.6 with the KDJ11-E processor (used with the PDP-11/93 and PDP-11/94 processors), to ensure correct use of the Time of Year (TOY) clock, the console firmware should be at Version 2.01 or later.

---

## 8.13 Certification

RSX-11M-PLUS Version 4.6 is compliant with the following standards within the date range from 1-January-1977 through 31-December-2064, subject to the conditions noted in this chapter, including the use of ISO date format where applicable, and year inferencing.

- BSI-DISC PD2000-1
- ANSI X3.30
- ISO 8601:1988

In those cases where inferencing rules are required for use within the warranted date range, functions which use two digits shall infer that years greater than 77 are in the twentieth century, and years which are less than or equal to 64 are in the twenty-first century.

The following is a table showing the components and functions which have been tested, and their tested date limitations:

# Year 2000 Certification for RSX-11M-PLUS Version 4.6

**Table 8-5 RSX-11M-PLUS Utilities Tested**

Utility	Tests	Limiting Date
Directives	ABRT\$, ALTP\$, ALUN\$, ASTX\$, ATRG\$, CINT\$, CLEF\$, CLOG\$, CMKT\$, CNCT\$, CRAW\$, CRGF\$, CRRG\$, CSRQ\$, DECL\$, DLOG\$, DSAR\$, DSCP\$, DTRG\$, ELAW\$, ELGF\$, EMST\$, ENAR\$, ENCP\$, EXIF\$, EXIT\$, EXST\$, EXTK\$, GCII\$, GLUN\$, GMCR\$, GMCX\$, GPRT\$, GREG\$, GTIM\$, GTSK\$, MAP\$, MRKT\$, QIO\$, QIOW\$, RCST\$, RCVX\$, RDAF\$, RDEF\$, RDXF\$, RPOI\$, RQST\$, RREF\$, RSUM\$, RUN\$, SCAA\$, SDAT\$, SETF\$, SDRC\$, SDRP\$, SFPAS\$, SMSG\$, SNXC\$, SPEA\$, SPND\$, SPRA\$, SPWN\$, SREF\$, SRDA\$, STIM\$, STLO\$, STOP\$, STSE\$, SVDB\$, SVTK\$, UMAP\$, USTP\$, WSIG\$, WTLO\$, WTSE\$	Dec-2155
SYSLIB—Date routines	\$DATx, \$EDMSG	Dec-2155
System—File System	Creation/Revision Date	Dec-2099
System—Drivers	No date dependencies	Dec-2155
System—Errorlogging	Date/time stamps	Dec-2155
ACS	No date dependencies	Dec-2155
ALL	No date dependencies	Dec-2155
ASN	No date dependencies	Dec-2155
BAD	No date dependencies	Dec-2155
BOO	No date dependencies	Dec-2155
BRO	Date/time stamp	Dec-2155
BYE	Date/time stamp	Dec-2155
BRU	/SINCE, /THROUGH, /CREATED, /MODIFIED	Dec-2099
CDA	Crash date	Dec-2155
CFL	Listing files	Dec-2155
CLQ	Date time due	Dec-2155
CMP	No date dependencies	Dec-2155
CON	Date/time stamp	Dec-2155
COT	Date/time stamp	Dec-2155
CRF	Listing files	Dec-2155
DCL	Date parsing	Dec-2155
DMO	No date dependencies	Dec-2155
DMP	Header ID area	Dec-2155

## Year 2000 Certification for RSX-11M-PLUS Version 4.6

**Table 8-5 (Cont.) RSX-11M-PLUS Utilities Tested**

Utility	Tests	Limiting Date
EDI	No date dependencies	Dec-2155
EDT	DATE command	Dec-2155
ELI	Date/time stamp	Dec-2155
ERRLOG	Date/time stamp	Dec-2155
F11ACP	Header date information	Dec-2155
FLX	DOS volumes	Dec-2035
FLX	RT-11 volumes	Dec-2099
FMT	No date dependencies	Dec-2155
FTB	Map files	Dec-2155
HEL	Date/time stamps	Dec-2155
HOM	Home block creation/revision dates	Dec-2155
HRC	Date/time stamp	Dec-2155
ICP	Date/time stamps, date symbol	Dec-2155
INI	Home block creation dates	Dec-2155
INS	No date dependencies	Dec-2155
IOX	Date/time stamps	Dec-2155
LOA	No date dependencies	Dec-2155
LBR	library headers	Dec-2155
LPP	Date/time stamps	Dec-2155
MAG	No date dependencies	Dec-2155
MCR	CLQ,TIM	Dec-2155
MOU	No date dependencies	Dec-2155
MTAACP	Header date information	Dec-2099
MACRO-11	Assembly listing files	Dec-2155
PAT	No date dependencies	Dec-2155
PIP	DIR/THROUGH, DIR/FULL, DIR/SINCE, COPY/SINCE, COPY/THROUGH, COPY/PRESERVE_DATES, APPEND/SINCE, APPEND/THROUGH, DELETE/SINCE, DELETE/THROUGH	Dec-2155
PMD	Date/time stamps	Dec-2155
PMT	Date/time stamps	Dec-2155
QMGCLI	After date processing	Dec-2099
QMG	After date processing	Dec-2099
RCT	No date dependencies	Dec-2155
RMD	Date/time stamps	Dec-2155

Year 2000 Certification for RSX-11M-PLUS Version 4.6

Table 8-5 (Cont.) RSX-11M-PLUS Utilities Tested

Utility	Tests	Limiting Date
RMS-11	RMSDSP, RMSIFL, RMSCNV, RMSBCK, RMSRST, RMSDEF, RMSDES	Dec-2099
RPT	/DATE:TODAY, /DATE:YESTERDAY, /DATE:RANGE:	Dec-2155
SAV	No date dependencies	Dec-2155
SHA	No date dependencies	Dec-2155
SHF	No date dependencies	Dec-2155
SHUTUP	Date/time stamps	Dec-2155
SLP	No date dependencies	Dec-2155
TIME	TOY, system	Dec-2064
TKTN	Date/time stamps	Dec-2155
TKB	Map files	Dec-2155
UFD	No date dependencies	Dec-2155
UNL	No date dependencies	Dec-2155
VCP	Date/time stamps	Dec-2155
VFY	No date dependencies	Dec-2155
VMR	TIME command	Dec-2155
ZAP	No date dependencies	Dec-2155

# A

---

## Reporting Problems

This appendix describes submitting problem reports.

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

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

An SPR can be used for:

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

An SPR cannot be used for:

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

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

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

### Categories of SPRs:

- Problem/Error SPR

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

- Suggested Enhancements/Other SPR

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

## Reporting Problems

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

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

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

- **CRASH**—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**—Controller/device information, software options, error log output, a copy of device registers, and a sample program.
- **UTILITIES**—A copy of your terminal output, showing setup commands, before and after effects, and relevant file information.
- **TASK BUILDER**—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: output from the File Structure Verification Utility (VFY) and dump of the volume; for improper results: the error code, a file header dump, and a sample program.
- **ERROR LOG REPORT GENERATOR (RPT)**—A copy of the report file generated by RPT, either a hardcopy listing or 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.

---

## B 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.

---

### B.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:

```
>SET /UIC=[11,40] 
>EDI REQSB.COR 
[Creating new file]
Input
REQSB.MAC;2/AU/-BF=REQSB.MAC;1
.
.
*EX
[Exit] 
>SET /UIC=[11,10] 
>SLP @[11,40]REQSB.COR 
```

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:

```
>SET /UIC=[11,24] 
>MAC REQSB, [11,34]REQSB/-SP=[1,1]EXEMC/ML, [11,10]RSXMC/PA:1,REQSB 
```

- 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:

```
>SET /UIC=[1,24] 
>LBR RSX11M/RP/-EP=[11,24]REQSB 
```

- 3 Perform the following sections of SYSGEN:
  - Building the Executive and Drivers

## Applying Corrections to Source Files

- 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:

```
>SET /UIC=[11,24] 
>MAC ZZDRV=[1,1]EXEMC/ML,[11,10]RSXMC/PA:1,ZZDRV
>SET /UIC=[1,24] 
>LBR RSX11M/RP/-EP=[11,24]ZZDRV
```

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:

```
>ASN SY:=OU: 
>TKB @[200,200]ZZDRVBLD
>SET /SYSUIC
SYSUIC=[g,m]
>SET /UIC=[g,m] 
>PIP /NV=[1,54]ZZDRV.TSK,ZZDRV.STB
```

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:

```
>VMR 
Enter filename:RSX11M
VMR>UNL ZZ: 
VMR>LOA ZZ: 
VMR> 
>RUN $SHUTUP
```

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>UNL ZZ:/VEC 
VMR>LOA ZZ:/VEC 
VMR> 
```

Hardware boot the modified system.

---

## B.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:

## Applying Corrections to Source Files

```
>SET /UIC=[12,40] [Return]
>EDI SETOV.COR [Return]
[Creating new file]
Input
SETOV.MAC;2/AU/-BF=SETOV.MAC;1
.
.
*EX [Return]
[Exit]

>SET /UIC=[12,10] [Return]
>SLP @[12,40]SETOV.COR [Return]
```

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

```
>SET /UIC=[12,24] [Return]
>MAC SETOV=[1,1]EXEMC/ML, [11,10]RSXMC/PA:1, [12,10]SETOV [Return]
```

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.COM) 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:

```
>SET /UIC=[1,24] [Return]
>LBR MCR/RP/NOEP=[12,24]SETOV [Return]
>PIP SETOV.OBJ;*/DE [Return]
>ASN SY:=MP: [Return]
>ASN SY:=IN: [Return]
>TKB @MCRBLD [Return]
>VMR [Return]
Enter filename:RSX11M [Return]
VMR>REM ...MCR [Return]
VMR>INS [3,54]MCR [Return]
VMR> [Ctrl/Z]
```

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

```
>SET /UIC=[1,24] [Return]
>LBR MCR/RP/NOEP=[12,24]MCRDIS [Return]
>PIP MCRDIS.OBJ;*/DE [Return]
>ASN SY:=MP: [Return]
>ASN SY:=IN: [Return]
>TKB @MCRBLD [Return]
>VMR [Return]
Enter filename:RSX11M [Return]
VMR>REM MCR... [Return]
VMR>INS [3,54]MCD/XHR=NO [Return]
VMR> [Ctrl/Z]
>RUN $$SHUTUP [Return]
```

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.

## Applying Corrections to Source Files

- 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:

```
>SET /UIC=[1,24] 
>LBR INS/RP=[12,24] INSLB 
```

- 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:

```
>SET /UIC=[1,24] 
>ASN SY:=MP: 
>TKB @INSBLD 
>VMR 
Enter filename:RSX11M 
VMR>REM ...INS 
VMR>INS [3,54] INS/IOP=NO 
VMR> 
>RUN $SHUTUP 
```

Hardware boot the system.

---

## B.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.

```
>LBR [1,24]DCLO/LI 
$LIBRARY/LIST [1,24]DCLO 
```

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.

## Applying Corrections to Source Files

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

```
>SET /UIC=[23,24] Return
>MAC xxx=[11,10]RSXMC/PASS:1,[23,10]DCLMAC/PASS:1,xxx Return
>SET /UIC=[1,24] Return
>LBR DCLO/RP=[23,24]xxx Return
>TKB @DCLBLD Return
```

If your CLI is DCL, use the following commands:

```
$ SET DEFAULT [23,34] Return
$ MACRO [11,10]RSXMC/PASS:1,[23,10]DCLMAC/PASS:1,xxx Return
$ SET DEFAULT [1,24] Return
$ LIBRARY/REPLACE DCLO [23,24]xxx Return
$ LINK @DCLBLD Return
```

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):

```
>CLI /ELIM=DCL Return
>REM DCL Return
>INS $DCL/CLI=YES Return
>CLI /INIT=DCL Return
```

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).

---

## B.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:

```
>HELLO SYSTEM Return
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):

```
>UFD DB0:[13,10] Return
>UFD DB0:[13,40] Return
>BRU Return
BRU>/NOINITIALIZE/SUPERSEDE/NOPRESERVE/BACKUP_SET:MPBL40SRC Return
From: MM0:[13,10]F11PRE.MAC,WTRN1.MAC Return
To: DB0: Return
BRU--Completed
BRU> CtrlZ
```

3 Create the correction file, as follows:

```
>SET /UIC=[13,40] Return
>EDT WTRN1.COR Return
```

4 Apply the correction, as follows:

```
>SET /UIC=[13,10] Return
>SLP @[13,40]WTRN1.COR Return
```

## Applying Corrections to Source Files

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

```
>SET /UIC=[13,24] 
>MAC WTRN1=[1,1]EXEMC/ML, [11,10]RSXMC, [13,10]F11PRE,WTRN1
```

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

```
>SET /UIC=[1,24] 
>LBR FCP/FP=[13,24]WTRN1 
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:

```
>ASN NL:=MP: 
>TKB @FCPxxxBLD
```

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

```
>RUN $VMR 
Enter filename: RSX11M
VMR>REM F11ACP 
VMR>INS [3,54]FCPxxx/PAR=GEN/IOP=NO/CKP=NO 
VMR> 
>RUN $SHUTUP 
```

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

---

## B.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.

## Applying Corrections to Source Files

- 2 If necessary, restore the required files from the distribution tape as follows:

```
>UFD DB0:[27,10] [Return]
>UFD DB0:[27,40] [Return]
>UFD DB0:[27,24] [Return]

>BRU [Return]
BRU>/NOINITIALIZE/SUPERSEDE/NOPRESERVE/BACKUPSET:MPBL40SRC [Return]
From:MM0:[27,10]CNPRE.MAC,CNCMR.MAC [Return]
To:DB0:[Return]
BRU -- Completed
BRU> [CTRLZ]
```

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

```
From:MM0:[27,10]HRPRE.MAC,HRONL.MAC [Return]
```

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

```
>SET /UIC=[27,40] [Return]
>EDT CNCMR.COR [Return]
```

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

```
>SET /UIC=[27,10] [Return]
>SLP @[27,40]CNCMR.COR [Return]
```

- 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:

```
>SET /UIC=[27,24] [Return]
>MAC CNCMR=[1,1]EXEMC/ML,[11,10]RSXMC/PA:1,[27,10]CNPRE,CNCMR [Return]
```

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

```
>SET /UIC=[1,24] [Return]
>LBR OLR/RP/NOEP=[27,24]CNCMR.OBJ [Return]
```

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

```
>ASN DB0:=IN: [Return]
>ASN DB0:=OU: [Return]
>ASN NL:=MP: [Return]

>TKB @CONBLD [Return]
```

**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 the following:

```
>RUN $VMR [Return]
Enter filename:RSX11M [Return]
VMR>REM ...CON [Return]
VMR>INS [3,54]CON [Return]
VMR> [CtrlZ]
```

## Applying Corrections to Source Files

For HRC, type the following:

```
>RUN $VMR   
Enter filename:RSX11M   
VMR>REM HRC...   
VMR>INS [3,54]HRC/IOP=NO   
VMR> 
```

- 9 In order to use the new reconfiguration tasks, follow these steps:
  - a. Shut the system down using the SHUTUP utility as follows:

```
>RUN $SHUTUP 
```
  - b. Reboot the system.

---

## B.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.

---

## B.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.

## B.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:

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

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,FCSEBIGBUF/PA:1,FCSPRE,CLOSE
```

`FCSANSI.MAC` enables the ANSI tape conditional assembly directives, and `FCSEBIGBUF.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 `FCSEMULTIBUF.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`,

## Applying Corrections to Source Files

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.

---

### B.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:

```
>SET /UIC=[1,1] Return  
>PIP /NV=SYSLIB.OLB Return  
>LBR SYSLIB/RP=[50,24]DELETE.OBJ, PUT.MTA, PUTSQ.MTA Return  
>LBR SYSLIB/RP=[50,24]FINIT.OBJ, FINTSL.SUP/-EP Return
```

If you use NOANSLIB.OLB, also use the following procedure:

```
>PIP /NV=NOANSLIB.OLB Return  
>LBR NOANSLIB/RP=[50,24]DELETE.OBJ, PUT.NMT, PUTSQ.NMT Return  
>LBR NOANSLIB/RP=[50,24]FINIT.OBJ, FINTSL.SUP/-EP Return
```

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

---

- ABO (MCR) command
  - restriction • 1–21
- ABORT (DCL) command
  - restriction • 1–21
- Account
  - access • 7–4
  - Disabled • 7–4
    - Network login • 7–4
    - Remote login • 7–4
  - Expiration • 7–5
- Account Maintenance Program
  - See ACNT
- ACNT • 7–1
  - blank account fields • 3–49
  - Enhanced Security • 2–7
- ALLOCATION field
  - RMS–11 • 3–64
- ALT command
  - keywords
    - /RPRI • 3–9
    - /TERM • 3–9
- Alternate record key field
  - RMS–11 • 3–62
- ASN (MCR) command • 3–9 to 3–11
- ASSIGN (DCL) command
  - /QUEUE • 3–2
- AST (Asynchronous system trap)
  - user completion routines • B–11

---

## B

---

- BACKUP (DCL) command
  - corrected example • 3–5
  - qualifiers
    - /CREATED • 3–4 to 3–5
    - /ERROR\_LIMIT • 3–7
    - /MODIFIED • 3–4 to 3–5
  - restrictions
    - /LENGTH qualifier • 3–7
    - /NEW\_VERSION qualifier • 3–7
    - /REWIND/APPEND qualifiers • 3–8
    - /TAPE\_LABEL qualifier • 3–8
    - /VERIFY qualifier • 3–7

- BAD
  - /READ switch • 1–12
- Bad Block Replacement Control Task
  - See RCT
- BASIC-PLUS–2
  - correction file • 5–1
  - DATE\$ update for years past 2003 • 5–2
  - invalid installation warning • 5–2
- BOO
  - Corrections • 1–19
  - /HW switch • 1–9
- Boot
  - Hardware Boot • 1–9
  - /HW switch • 1–9
  - virtual disks • 1–9
- /BR switch
  - RMS–11 • 3–66
  - RMSDSP utility • 3–66
- BRU
  - messages
    - error • 3–57 to 3–58
  - qualifiers
    - /COMPARE • 3–54 to 3–58
    - /DOUBLEBUFFER • 3–54 to 3–58
    - /SINGLEBUFFER • 3–54 to 3–58
    - /VERIFY • 3–54 to 3–58
  - restrictions • 1–20, 3–7 to 3–8
    - /ERROR\_LIMIT qualifier • 3–7
    - /LENGTH qualifier • 3–7
    - /NEW\_VERSION qualifier • 1–20, 3–7
    - /REWIND/APPEND qualifiers • 3–8
    - /TAPE\_LABEL qualifier • 3–8
    - use of wildcards • 1–20
    - /VERIFY qualifier • 3–7
  - supplemental information • 3–54 to 3–58
  - Year 2000 dates • 1–7
- BUCKETSIZE field
  - RMS–11 • 3–65
- BYE • 7–9

---

## C

---

- CALL GINABO • 3–22
- CALL GINAPR • 3–23
- CALL GINDEF • 3–23

## Index

CALL GINDEV • 3-23  
CALL GINDVJ • 3-24  
CALL GINFMK • 3-24  
CALL GINGAS • 3-24  
CALL GINQMC • 3-25  
CALL GINREN • 3-25  
CALL GINSPR • 3-26  
CALL GINTSK • 3-26  
CALL GINUAB • 3-27  
CALL GINUIC • 3-27  
CALL GINUPD • 3-27  
CALL GINVEC • 3-28  
Cancel Mark Time Request (CMKT\$) directive • 3-28  
CINT\$ directive • 3-31  
CLEAR ALL command  
    RMSDES utility • 3-64  
CLI/INIT command • B-5  
CMKT\$ directive • 3-28  
COBOL-81  
    correction file • 5-1  
Communication driver  
    example • 3-44  
    programming hints • 3-42  
Communication mode  
    setting full duplex (communication driver) • 3-40  
Communication set operation  
    IO.SYN function (communication driver) • 3-40  
/COMPARE qualifier  
    BRU • 3-55, 3-56  
CON  
    SET command • 1-16  
    SYS attribute • 1-16  
Connect to Interrupt Vector (CINT\$) directive • 3-31  
Contiguous file areas  
    RMS-11 • 3-63  
Continuation buckets  
    RMSDES utility • 4-4  
Controller identification  
    DSA driver • 3-45  
Correction files  
    BASIC-PLUS-2 • 5-1  
    COBOL-81 • 5-1  
    DATATRIEVE-11/RSX • 5-1  
    FORTRAN-IV • 5-1  
    layered products • 5-1  
    PDP-11 FORTRAN-77 • 5-1  
CPU  
    supported processors  
        M1 • 1-2  
        M11 • 1-2  
Crash device  
    restriction on specifying during SYSGEN • 1-20

Crash driver  
    memory dump • 3-20  
Crash dump  
    loadable support • 3-20  
Create Regions (CRRG\$) directive • 3-28  
CRRG\$ directive • 3-28

---

## D

---

Data caching  
    write loading temporary files • 3-49  
Datatrieve-11  
    Report Writer headings past year 2000 • 5-3  
DATATRIEVE-11/RSX correction file • 5-1  
Date  
    BRU dates • 1-7  
    Components Modified • 1-6  
    Date references • 1-8  
    DMP Header ID Dates • 1-8  
    EDT Date Command • 1-7  
    ERRLOG Date handling updated • 1-7  
    FLX  
        DOS Year 2000 dates • 1-7  
        RT-11 Year 2000 dates • 1-7  
    Help • 1-8  
    INI Home block dates • 1-7  
    Macro-11 listings • 1-8  
    PIP Date Range comparison • 1-7  
    QUEUE command modification • 1-7  
    RPT Date Range Switch • 1-7  
    SYSLIB • 1-15  
    TIM Command in VMR • 1-8  
    Time command modification • 1-6  
DATE (FORTRAN-77)  
    update for years past 2000 • 5-2  
DATE\$ (BASIC-PLUS-2)  
    update for years past 2003 • 5-2  
DCL  
    command  
        SET  
            SYSTEM/POOL/LIMITS • 3-7  
    Command  
        SHOW DEVICE enhancements • 1-13  
    commands  
        ABORT restriction • 1-21  
        ASSIGN/QUEUE • 3-2  
        BACKUP  
            /CREATED • 3-4 to 3-5  
            /MODIFIED • 3-4 to 3-5  
            qualifiers • 3-7  
            restrictions • 3-7

- DCL
  - commands (cont'd)
    - DEASSIGN/QUEUE • 3-3
    - DELETE
      - /QUEUE • 3-3
    - FORMAT
      - WRITE:n • 3-4
    - INITIALIZE • 3-3 to 3-4
      - /HEADERS qualifier • 3-3 to 3-4
      - /MAXIMUM\_FILES:n qualifier • 3-3 to 3-4
      - /QUEUE • 3-3
    - PRINT • 3-1, 3-2
    - SET
      - SYSTEM/CRASH\_DEVICE • 3-7
      - TERMINAL VT300 and later series restriction • 3-3
      - TERMINAL/*model* • 3-3
    - SHOW
      - ASSIGNMENTS • 3-5 to 3-7
      - LOGICALS • 3-5 to 3-7
      - QUEUE/ALL • 3-1
      - START/QUEUE • 3-3
      - STOP/QUEUE • 3-3
  - Date formats • 1-13
  - DUMP
    - New Command • 1-13
  - DUMP command • 1-13
  - updating source file • B-4
- Deadlock
  - RMS-11 • 3-62
- DEASSIGN/QUEUE (DCL) command • 3-3
- DELETE/QUEUE (DCL) command • 3-3
- DEV
  - /FILES switch • 1-13
  - /FULL switch • 1-13
- Device
  - allocation failure • 3-62
  - disconnecting (communication driver) • 3-41
  - DU
    - Large disk support • 1-16
  - initializing (communication driver) • 3-41
  - KDJ11-E
    - TOY clock restrictions • 1-2
  - list of supported devices • 1-2
  - MENTEC M-series processors • 1-2
    - M1 • 1-2
    - M11 • 1-2
    - M70,M80,M90,M100 • 1-2
  - MSCP devices • 1-12
  - RX33 drive • 3-37
  - RZ29B • 1-12
  - RZ29B disk drive • 1-2
- Device (cont'd)
  - TK25 cartridge tape • 3-21
- Device driver
  - changes • B-1
  - updating loadable • B-2
- DFL (MCR) command • 3-9 to 3-12
- DIGITAL Command Language
  - See DCL
- Directives • 3-21 to 3-35
  - Abort (GINABO) • 3-22
  - Cancel Mark Time Request (CMKT\$) • 3-28
  - CINT\$ • 3-31
  - CMKT\$ • 3-28
  - Connect to Interrupt Vector (CINT\$) • 3-31
  - Create Region (CRRG\$) • 3-28
  - CRRG\$ • 3-28
  - \$EDMSG • 3-46
  - Extend Task (EXTK\$) • 3-29, 3-32 to 3-33
  - EXTK\$ • 3-29, 3-32 to 3-33
  - EXTM\$ • 3-29
  - FEAT\$ • 3-29
  - Find and Return Task Information (GINTSK) • 3-26
  - Get Assigned Device Name (GINGAS) • 3-24
  - Get Device Information (GINDEV) • 3-23
  - Get Device Information Junior (GI.DVJ) • 3-34 to 3-35
  - Get Feature Mask Words (GINFMK) • 3-24
  - Get LUN Information (GLUN\$) • 3-38
  - Get Partition Parameters (GPRT\$) • 3-35
  - Get Region Parameters (GREG\$) • 3-35
  - Get Son of Device (GINDVJ) • 3-24
  - Get System APR (GINAPR) • 3-23
  - Get System UIC Information (GINUIC) • 3-27
  - Get User Account Block (GINUAB) • 3-27
  - GI.DVJ • 2-4 to 2-5
  - GI.VEC • 2-5 to 2-7
- GIN
  - GINABO call • 3-22
  - GINAPR call • 3-23
  - GINDEF call • 3-23
  - GINDEV call • 3-23
  - GINDVJ call • 3-24
  - GINFMK call • 3-24
  - GINGAS call • 3-24
  - GINQMC call • 3-25
  - GINREN call • 3-25
  - GINSPR call • 3-26
  - GINTSK call • 3-26
  - GINUAB call • 3-27
  - GINUIC call • 3-27
  - GINUPD call • 3-27
  - GINVEC call • 3-28
- GIN\$

## Index

### Directives

- GIN\$ (cont'd)
    - GI.DVJ • 3-34 to 3-35
  - GLUN\$ • 3-38
  - GPRT\$ • 3-35
  - GREG\$ • 3-35
  - Queue (MCR) Command Line (GINQMC) • 3-25
  - Recursive Translation of Logical Name (RLOG\$) • 3-30
  - Recursive Translation of Logical Name (RLON\$) • 3-30
  - Rename Task (GINREN) • 3-25
  - RLOG\$ • 3-30
  - RLON\$ • 3-30
  - SDUN\$ • 3-35
  - Set Task Default UIC (GINDEF) • 3-23
  - Set Task Privilege (GINSR) • 3-26
  - Test for Specified System Feature (FEAT\$) • 3-29
  - Test for Specified Task Feature (TFEA\$) • 3-31
  - TFEA\$ • 3-31
  - TLOG\$ • 3-31
  - TLO\$ • 3-31
  - Translate Executive Entry Point Vector (GINVEC) • 3-28
  - Translate Logical Name String (TLOG\$) • 3-31
  - Translate Logical Name String (TLO\$) • 3-31
  - Update UICs and Default Directory (GINUPD) • 3-27
  - VSUN\$ • 3-35
- Disk Save and Compress utility  
See DSC
- Distribution Kits • 1-3
  - Pregenerated System • 1-3
  - RL02 • 1-3
  - TK50 • 1-3
- DMC11 (communication driver)
  - message send • 3-44
  - powerfail • 3-43
  - synchronous line interface • 3-38
- DMP
  - Header ID Dates • 1-8
- DSA driver
  - controller identification • 3-45
  - error log file • 3-45
  - error log support • 3-45
  - error logging support • 3-45
  - error packets • 3-45
  - I/O sequence number • 3-44, 3-45
  - offset P.SEQ • 3-45
  - powerfail recovery
    - type • 3-45
  - Read Serial Number function • 3-45
  - time delay • 3-45

- DSC • 2-1
- DU driver
  - Read Serial Number function • 3-44
  - setting to volume valid • 3-44
- DUP11 (communication driver)
  - sync character • 3-38
  - synchronous line interface • 3-38

---

## E

---

- \$EDMSG directive • 3-46
  - file name string conversion • 3-46
- ELI
  - restrictions
    - /NOLIM switch • 1-21
    - /SH switch • 1-21
- Enhanced security
  - See ACNT, PSW, HELLO, BYE
- /ER switch
  - RMS-11 • 3-65 to 3-66
  - RMSCNV utility • 3-65 to 3-66
- ER\$FUL error • 3-62
- ER\$KEY error • 3-61
- ER\$MRS error • 3-62
- ER\$NOD error • 3-62
- Error Logging
  - See also ELI
  - ERRLOG
    - Date handling updated • 1-7
  - file for DSA driver • 3-45
  - restrictions • 1-20
    - DU devices
      - improper display of register fields • 1-20
    - improper display of register fields • 1-20
    - MU devices
      - improper display of register fields • 1-20
    - to history summaries • 1-20
- RPT
  - Date Range Switch • 1-7
- support
  - for DSA driver • 3-45
  - for RA70 disk drives • 3-36
  - for RA90 disk drives • 3-36
- Error packets
  - DSA driver • 3-45
- Errors
  - ER\$FUL • 3-62
  - ER\$KEY • 3-61
  - ER\$MRS • 3-62
  - ER\$NOD • 3-62
  - RMSDES utility • 4-4

/ERROR\_LIMIT qualifier  
     BACKUP (DCL) command • 3-7  
 Executive • 1-18  
     Data Structures  
         HWDDF\$ • 1-17  
         Vectored symbols • 1-17  
     modification to IOSUB.MAC • 3-32  
     modifying source file • B-1  
     MSCPDRV • 1-18  
     RDDR • 1-18  
     restriction on shared regions • 3-32  
     support for volume valid • 3-36  
     TDSCH • 1-18  
 Executive Debugging Tool  
     See XDT  
 EXIT\_SUPERSEDE command  
     RMSDES utility • 3-64, 4-4  
 Extend Task (EXTK\$) directive • 3-29  
     APR protection mask • 3-32 to 3-33  
 EXTENSION field  
     RMS-11 • 3-65  
 EXTK\$ directive • 3-29, 3-32 to 3-33  
 EXTM\$ directive • 3-29

---

## F

---

FAL  
     support for remote record access to indexed files • 3-63  
 FCS  
     ANSI • B-8  
     assembly files • B-9  
     code size • 2-2  
     modifying modules • B-8  
     multibuffered • B-8  
     non-ANSI • B-8  
     READ\$ function • B-11  
     source file  
         conditional assembly • B-8  
         updating • B-8  
     VMS ACP function support • 2-1  
     WRITE\$ function • B-11  
 FCSRES  
     now uses EIS • 1-17  
     Y2K Enhancements • 1-15  
 FEAT\$ directive • 3-29  
 File Access Listener  
     See FAL  
 File Control Services  
     See FCS

File loading  
     RMS-11 restriction • 4-4  
 File specification  
     RMS-11 • 3-66  
         wildcard • 3-66  
     RMSBCK utility • 3-66  
     RMSDSP utility • 3-66  
 Files-11 ACP  
     Additional buffer space • 1-10  
 FLX  
     DOS and RT-11 Year 2000 dates • 1-7  
     Multi-partitioned RT-11 disks • 1-11  
 FMS  
     command file example • 3-50  
 .FORM directive  
     terminal support • 3-8  
 FORTRAN IV-PLUS • 3-50  
 FORTRAN-77  
     DATE, IDATE • 5-2  
     update for years past 2000 • 5-2  
 FORTRAN-77 correction file • 5-1  
 FORTRAN-77 support • 3-50  
 FORTRAN-IV  
     correction file • 5-1  
 Full-duplex mode  
     considerations (communication driver) • 3-42  
     setting mode (communication driver) • 3-40

---

## G

---

GEN region name • 3-35  
 Get Device Information Junior function (GI.DVJ) • 3-34 to 3-35  
 Get LUN Information (GLUN\$) directive • 3-38  
 Get Partition Parameter (GPRT\$) directive • 3-35  
 Get Region Parameters (GREG\$) directive • 3-35  
 GI.DVJ directive • 2-4 to 2-5  
 GI.VEC directive • 2-5 to 2-7  
 GIN directives  
     GINABO call • 3-22  
     GINAPR call • 3-23  
     GINDEF call • 3-23  
     GINDEV call • 3-23  
     GINDVJ call • 3-24  
     GINFMK call • 3-24  
     GINGAS call • 3-24  
     GINQMC call • 3-25  
     GINREN call • 3-25  
     GINSPR call • 3-26  
     GINTSK call • 3-26  
     GINUAB call • 3-27

## Index

GIN directives (cont'd)  
  GINUIC call • 3-27  
  GINUPD call • 3-27  
  GINVEC call • 3-28  
GIN\$ directive  
  GI.DVJ function • 3-34 to 3-35  
GLUN\$ directive • 3-38  
GLUN\$ macro • 3-38  
  get information  
    communication driver • 3-38  
GPRT\$ directive • 3-35  
GREG\$ directive • 3-35

---

## H

---

Half-duplex mode  
  considerations (communication driver) • 3-42  
  set mode (communication driver) • 3-41  
Hardware support  
  See Device and CPU  
HELLO  
  corrections  
    day-of-week routine • 1-19  
  Enhanced Security • 2-7  
HELLO task • 3-12  
HELP • 3-12  
HFT • 2-2  
HOM  
  Home block date updated • 1-7  
Host Reconfiguration  
  See CON  
HRC  
  dual-ported device handling • 1-16  
Hybrid Fast Task Builder  
  See HFT

---

## I

---

I/O sequence number  
  DSA driver • 3-44, 3-45  
ICM  
  Date symbols • 1-14  
Idle Session Logout • 7-6  
Indexed file  
  RMS-11 • 3-65  
Indirect  
  Date symbols • 1-14  
  4 digit years • 1-14  
  preprocessor (IPP)

Indirect  
  preprocessor (IPP) (cont'd)  
    restriction on .END statements • 1-22  
  restriction on .ONERR statements in .BEGIN/.END  
  blocks • 1-22

## INI

  Home block date updated • 1-7  
INI (MCR) command  
  /INF keyword • 3-12  
  guidelines • 3-13  
  /MXF keyword • 3-12  
  guidelines • 3-13  
INITIALIZE (DCL) command  
  /FORMAT  
    DL-type device restriction • 3-4  
  /HEADERS qualifier • 3-3 to 3-4  
  /MAXIMUM\_FILES qualifier • 3-3 to 3-4  
  guidelines • 3-4  
  /QUEUE • 3-3  
  /WRITE:n restriction • 3-4

## INS command

  Corrections • 1-19  
  UTLUIC search • 1-12

## Interface

  message-oriented (communication driver) • 3-38  
IO.FDX function (communication driver) • 3-40  
IO.HDX function  
  set mode (communication driver) • 3-40  
IO.INL function (communication driver) • 3-41  
  after sync • 3-43  
IO.RLB function (communication driver) • 3-41  
IO.RNS function (communication driver) • 3-41  
IO.SYN function (communication driver) • 3-42  
  set operation • 3-40  
  with IO.INL • 3-43  
IO.TRM function (communication driver) • 3-41  
IO.WNS function (communication driver) • 3-42  
IOSUB.MAC module  
  modification • 3-32

---

## K

---

KDJ11-E  
  TOY clock • 1-18  
  TOY clock restrictions • 1-2, 1-18

---

## L

---

Layered products  
 BASIC-PLUS-2  
   invalid installation warning • 5-2  
 correction files • 5-1  
   BASIC-PLUS-2 • 5-1  
   COBOL-81 • 5-1  
   DATATRIEVE-11/RSX • 5-1  
   FORTRAN-77 • 5-1  
   FORTRAN-IV • 5-1  
 Restriction  
   install not using named directories • 5-1  
   installing from logged in console • 5-1  
 LOA (MCR) command • 3-13  
 Logical unit number  
   See LUN  
 Login Date  
   Options • 7-8  
 Login Failure Disconnect • 7-8  
 Low-traffic sync (communication driver) • 3-43  
 LUN  
   get information  
     communication driver • 3-38

---

## M

---

Macro-11  
   Date in listings • 1-8  
 Macros  
   GLUN\$ • 3-38  
   QIO\$ • 3-39 to 3-42  
     functions  
       IO.FDX • 3-40  
       IO.HDX • 3-40  
       IO.INL • 3-41  
       IO.RLB • 3-41  
       IO.RNS • 3-41  
       IO.SYN • 3-42  
       IO.TRM • 3-41  
       IO.WLB • 3-42  
       IO.WNS • 3-42  
   QIO\$C • 3-37  
 MCR  
   Command  
     DEVICE enhancements • 1-13  
   commands  
     ABO restriction • 1-21  
     ALT  
       /RPRI keyword • 3-9  
       /TERM keyword • 3-9  
     ASSIGN (ASN) • 3-9 to 3-11  
     DEFINE LOGICALS (DFL) • 3-9 to 3-12

MCR  
   commands (cont'd)  
     INITIALIZE VOLUME (INI)  
       /INF qualifier • 3-12, 3-13  
       /MXF qualifier • 3-12, 3-13  
     LOAD (LOA) • 3-13  
     OPENREGISTER (OPE)  
       /TASKD qualifier • 3-14  
       /TASKI qualifier • 3-14  
     PIP • 3-18  
     PRI • 3-1, 3-2  
     QUE  
       /AS • 3-2  
       /CR • 3-3  
       /DEA • 3-3  
       /DEL • 3-3  
       /STA • 3-3  
       /STO • 3-3  
     RUN  
       /PRI keyword • 3-15  
       /UIC keyword • 3-14, 3-15  
     SET  
       /CKP qualifier • 3-16  
       /CRASHDEV qualifier • 3-16, 3-20  
       /CRASH\_DEVICE qualifier • 3-16, 3-20  
       /CTRLC qualifier • 3-17  
       /DPRO qualifier • 3-17  
       /INQUIRE qualifier • 3-16, 3-17  
       /INTRUSION qualifier • 3-16  
       /PLCTL qualifier • 3-17  
       /SECPOL qualifier • 3-18  
       /TERM qualifier • 3-18  
     TIM  
       SETTOY switch • 3-18  
       SYNC switch • 3-18  
       updating source file • B-3  
 MENTEC M-Series  
   support • 1-2  
 Message-oriented interface (communication driver) • 3-38  
 Mode  
   maintenance (communication driver) • 3-41  
   set (communication driver) • 3-40  
 Monitor Console Routine  
   See MCR  
 MSCP support  
   RCT • 3-48  
 MSCPDRV  
   Shadowing • 1-18

## Index

---

### N

---

- /NV switch
  - RMSBCK utility • 3-66
  - RMSRST utility • 3-67

---

### O

---

- Object Time System
  - See OTS
- ODL files
  - RMS-11 • 4-9
- Offset P.SEQ
  - DSA driver • 3-45
- OPE (MCR) command
  - /TASKD keyword • 3-14
  - /TASKI keyword • 3-14
- OTS Fast Map routine
  - IOT restriction • 2-2
- Overlaid libraries
  - Manually loading • 3-50

---

### P

---

- Password
  - expiration • 7-2
  - length • 7-3
  - lifetime • 7-2
- PIP
  - Date Range comparison • 1-7
  - /PAGE switch • 1-15
  - qualifiers
    - /DD • 3-18
  - support
    - for wildcards • 3-58
- Pool
  - ICB pool expansion • 1-11
- Powerfail
  - DMC11 (communication driver) • 3-43
- Powerfail recovery
  - DSA driver • 3-45
- Pregenerated kit
  - Restoring from tape • 1-3
  - RMS-11 • 3-19, 4-5
- PRINT command
  - DCL • 3-1, 3-2
  - MCR • 3-1
    - PRI command enhancements • 3-2

- Problem reporting • A-1
  - information to be supplied • A-2
- Programming hints
  - communication driver • 3-42
- PSW
  - Enhanced Security • 2-7

---

### Q

---

- QIO\$ macro • 3-39 to 3-42
  - communication driver • 3-39
  - device-specific function
    - communication driver • 3-40
  - functions
    - IO.FDX • 3-40
    - IO.HDX • 3-40
    - IO.INL • 3-41
    - IO.RLB • 3-41
    - IO.RNS • 3-41
    - IO.SYN • 3-42
    - IO.TRM • 3-41
    - IO.WLB • 3-42
    - IO.WNS • 3-42
  - standard function
    - communication driver • 3-39
- QIO\$C macro • 3-37
- QUE (MCR) command • 3-1
  - /AS • 3-2
  - /CR • 3-3
  - /DEA • 3-3
  - /DEL • 3-3
  - /STA • 3-3
  - /STO • 3-3
  - Year 2000 dates • 1-8
- Queue Manager • 3-46

---

### R

---

- RCT
  - MSCP support • 3-48
  - write-protected disks • 3-48
- RDDRV
  - Powerfail • 1-18
- Read operation
  - logical block (communication driver) • 3-41
  - sync character (communication driver) • 3-41
- Rebuilding utilities
  - RMS-11 • 4-10

- Record Management Services
  - See RMS-11
- Recursive Translation of Logical Name (RLOG\$)
  - directive • 3-30
- Recursive Translation of Logical Name (RLON\$)
  - directive • 3-30
- Redundancy checking (communication driver) • 3-42
- Report Generator Task
  - /HISTORY qualifier • 3-21
- Report Writer (Datatrieve-11) headings past year
  - 2000 • 5-3
- Reporting problems • A-1
  - information to be supplied • A-2
- Resource Monitoring Display
  - See RMD
- Restriction
  - Layered products
    - install not using named directories • 5-1
    - installing from logged in console • 5-1
- Restrictions
  - KDJ11-E TOY clock • 1-2
- RL02 installation kit • 4-5
- RLOG\$ directive • 3-30
- RLOG\$ macro expansion • 3-30
- RLON\$ directive • 3-30
- RLON\$ macro expansion • 3-30
- RMD
  - BOTTOM command • 3-48
  - display restriction • 1-22
  - Free space on big disks • 1-14
  - TOP command • 3-48
  - Writes Load Rates • 3-47
- RMS-11
  - accessing files on remote nodes • 4-10
  - alternate record key field • 3-62
  - building utilities • 4-9
  - contiguous file areas • 3-63
  - creating a single-area indexed file • 3-65
  - creating indexed file • 3-65
  - deadlock • 3-62
  - errors
    - ER\$DNF • 4-2
    - ER\$FUL • 3-62
    - ER\$KEY • 3-61
    - ER\$MRS • 3-62
    - ER\$NOD • 3-62
  - FAL support for remote record access to indexed files • 3-63
  - fields
    - ALLOCATION • 3-64
    - BUCKETSIZE • 3-65
    - EXTENSION • 3-65
  - file loading restriction • 4-4
- RMS-11 (cont'd)
  - file specification • 3-66, 4-1
  - files on the distribution kit • 4-5 to 4-8
  - installing • 4-5
  - local access • 4-1
  - ODL files • 4-9
  - pregenerated kit • 3-19, 4-5
  - rebuilding utilities • 4-10
  - remote access • 4-3
  - remote access methods
    - dates • 4-3
  - restrictions
    - nonamed directory mode • 4-2
    - remote access methods • 4-2 to 4-4
    - RMSCNV utilities • 4-4
    - RMSDEF utilities • 4-4
    - to file loading • 4-4
    - to the DEF command • 4-4
    - to the DEFINE command • 4-4
  - RMSDAP utility
    - configuring for your system • 4-3
    - small systems • 4-4
    - startup command procedure • 4-8 to 4-9
    - task building against resident library • 3-63
    - TKB cluster option syntax • 3-63
    - UPDATE operation • 3-62
    - utilities • 4-2
      - command files • 4-9
      - configurations • 4-9
      - ODL files • 4-9
      - rebuilding • 4-10
  - RMSBCK
    - file specifications • 3-66
    - /NV switch • 3-66
  - RMSCNV
    - /ER switch • 3-65 to 3-66
    - restrictions • 4-4
  - RMSDAP
    - configuring for your system • 4-3
    - keysize field • 4-3
  - RMSDEF
    - restrictions • 4-4
  - RMSDES
    - CLEAR ALL command • 3-64
    - continuation buckets • 4-4
    - enhancements • 4-4
    - error messages • 4-4
    - EXIT\_SUPERSEDE command • 3-64, 4-4
    - SAVE command • 3-64
    - SAVE\_SUPERSEDE command • 3-64, 4-4
  - RMSDSP

## Index

RMS-11  
utilities  
RMSDSP (cont'd)  
/BR switch • 3-66  
command line format • 3-66  
file specification • 3-66  
/SU switch • 3-66  
RMSIFL  
enhancements • 4-2  
RMSRST  
command line format • 3-67  
/NV switch • 3-67  
RMSBCK utility  
file specification • 3-66  
/NV switch • 3-66  
RMSCNV utility  
/ER switch • 3-65 to 3-66  
file loading restriction • 4-4  
RMSDAP utility  
keysize field • 4-3  
RMSDEF utility  
DEF command restriction • 4-4  
DEFINE command restriction • 4-4  
RMSDES utility  
CLEAR ALL command • 3-64  
continuation buckets • 4-4  
error messages • 4-4  
EXIT\_SUPERSEDE command • 3-64, 4-4  
SAVE command • 3-64  
SAVE\_SUPERSEDE command • 3-64, 4-4  
RMSDSP utility  
/BR switch • 3-66  
command line format • 3-66  
file specification • 3-66  
/SU switch • 3-66  
RMSIFL utility  
enhancements • 4-2  
RMSRST utility  
command line format • 3-67  
/NV switch • 3-67  
RPT  
Date Range Switch • 1-7  
/HISTORY qualifier • 3-21  
RUN (MCR) command  
/PRI keyword • 3-15  
/UIC keyword • 3-14, 3-15  
RX33 drive • 3-37  
RZ29B disk drive • 1-2

---

## S

---

SAVE command  
RMSDES utility • 3-64  
SAVE\_SUPERSEDE command  
RMSDES utility • 3-64, 4-4  
SDUN\$ directives • 3-35  
Send directive • 3-35  
SET (DCL) command  
SYSTEM/CRASH\_DEVICE command • 3-7  
SYSTEM/POOL/LIMITS command • 3-7  
TERMINAL  
/INQUIRE command  
VT300-series support • 3-3  
VT400-series support • 3-3  
VT500-series support • 3-3  
/model • 3-3  
VT300 and later series support • 3-3  
SET (MCR) command  
/CKP • 3-16  
/CRASHDEV • 3-16, 3-20  
/CRASH\_DEVICE • 3-16, 3-20  
/CTRLC • 3-17  
/DPRO • 3-17  
/INQUIRE • 3-16, 3-17  
/INTRUSION • 3-16, 3-47  
/PLCTL • 3-17  
/SECPOL • 3-18  
/TERM • 3-18  
SET command  
UTLUIC • 1-12  
Set mode  
half duplex (communication driver) • 3-41  
IO.HDX function (communication driver) • 3-40  
SHA  
Command  
ACP switch • 2-9  
ERROR switch • 2-9  
LOG switch • 2-9  
SHARE switch • 2-9  
DISPLAY Command • 2-10  
Introduction • 2-8  
MERGE command • 2-7  
New qualifiers • 2-7  
SET command • 2-7  
SET Command • 2-9  
START/MERGE Command • 2-8  
Shadow Recording  
See SHA  
Shared regions  
I- and D-space restriction • 3-32  
SHOW (DCL) command  
ASSIGNMENTS • 3-5 to 3-7  
LOGICALS • 3-5 to 3-7  
QUEUE

SHOW (DCL) command  
   QUEUE (cont'd)  
     /ALL qualifier • 3-1  
 SHUTUP utility • B-8  
 SLP  
   correcting source files • B-1  
 Software Performance Report  
   See SPR  
 Source file  
   correcting • B-1  
   updating • B-1  
 Source Language Input Program  
   See SLP  
 SPR  
   category • A-1  
   completing • A-1  
   priority • A-2  
 START/QUEUE (DCL) command • 3-3  
 STOP/QUEUE (DCL) command • 3-3  
 /SU switch  
   RMS-11 • 3-66  
   RMSDSP utility • 3-66  
 SUBMIT (DCL) command • 3-1  
 Sync character (communication driver) • 3-41  
   specifying • 3-42  
 SYSGEN  
   corrections • 3-18  
     No FPP support • 2-12  
     to RMS-11 pregenerated kit • 3-19  
     to selecting FCPMDL • 3-18  
   restrictions • 1-19  
     incorrect displays for multiple MM devices •  
       1-19  
     to selecting FCPMDL • 1-19  
   specifying nonexistent device as the crash device •  
     1-20  
 SYSLIB  
   DARITH  
     \$CDDMG routine • 1-15  
   EDDAT  
     \$DAT routine • 1-15  
     \$DAT4 routine • 1-15  
     \$DAT1 routine • 1-15  
     \$DATS routine • 1-15  
   EDTMG  
     Date formats • 1-15  
     \$EDMSG routine • 1-15  
   Y2K Enhancements • 1-15  
 System Activity Logout • 7-6  
 System Inactivity Logout  
   Messages • 7-7  
   Timer • 7-7  
   Warning • 7-7

System library  
   using with TKB • 2-2

---

## T

---

Task  
   extending • 3-32 to 3-33  
   hanging when attaching to a terminal • 1-21  
   Universal Receiver Task • 2-11  
   URT • 2-11  
 Task Builder  
   See TKB  
 TDSCH  
   Date processing • 1-18  
 Technical Overview • 6-16  
 Terminals  
   unused ports • 1-21  
 Terminals VT300,VT400,VT500 • 1-20  
 Test for Specified System Feature (FEAT\$) directive •  
   3-29  
 Test for Specified Task Feature (TFEA\$) directive •  
   3-31  
 TFEA\$ directive • 3-31  
 TIM (MCR) command • 3-18  
 Time delay  
   DSA driver • 3-45  
 TKB  
   /-SH restriction • 1-22  
   allocating library APRs • 3-52  
   building new FCP • B-6  
   CLSTR option • 3-53  
   cluster option syntax • 3-63  
   command file • B-3  
   compatible library requirement • 2-2, B-11  
   corrections  
     to error message • 3-54  
     to the /INCLUDE qualifier • 3-51  
     to the /SAVE qualifier • 3-51  
   EXTK\$ task directives • 3-29  
   EXTM\$ task directives • 3-29  
   FMS command file example • 3-50  
   illegal APR reservation message • 3-54  
   /LI subswitch • 3-53  
   LIBR option • 3-53  
   RESLIB option • 3-53  
   RESSUP option  
     new /SW parameter code • 3-52  
   restrictions • 2-2  
     OTS Fast Map routine • 2-2  
   RMS-11 resident library • 3-63  
   RNSDCT option • 1-14

## Index

### TKB (cont'd)

- SUPLIB option
    - new /SW parameter code • 3-52
  - task-build command file • B-6
  - using system library • 2-2
  - VSECT option • 3-52
  - TLOG\$ directive • 3-31
  - TLOG\$ macro expansion • 3-31
  - TLON\$ directive • 3-31
  - TLON\$ macro expansion • 3-31
  - Translate Logical Name String (TLOG\$) directive • 3-31
  - Translate Logical Name String (TLON\$) directive • 3-31
  - Transmission
    - validation (communication driver) • 3-42
- 

## U

---

- Universal Receiver Task • 2-11
  - Unstop directive • 3-35
  - UPDATE operation
    - RMS-11 correction • 3-62
  - URT • 2-11
- 

## V

---

- VCP
  - Create Virtual Device • 6-12
- /VERIFY qualifier
  - BRU • 3-55, 3-56
- VFY
  - aborting • 3-58
  - switches
    - /DE • 3-58
    - /DH • 3-58
    - /RE • 3-58
    - /UP • 3-58
  - warning message • 3-58
- Virtual Device
  - Connect
    - /ATT • 6-5
    - /BAD • 6-5
    - /CREATE • 6-5
    - /DR • 6-5
    - /EXT • 6-5
    - /LT • 6-5
    - /MEM • 6-5
    - /MOU • 6-5
    - /NM • 6-5

### Virtual Device

- Connect (cont'd)
    - /PASS • 6-5
    - /RO • 6-5
    - /SAVE • 6-5
    - /SIZE • 6-5
    - /SN • 6-5
    - /TAPE • 6-5
    - /TYPE • 6-5
  - Create • 6-12
    - /ALTERNATE • 6-13
    - /DEVICE • 6-13
    - /TAPE • 6-13
  - Disconnect
    - /ALL • 6-9
    - /DEV • 6-9
    - /DMO • 6-9
    - /USE • 6-9
  - Set • 6-13
    - /EOF • 6-14
    - /EXT • 6-14
    - /OFF • 6-14
    - /ONL • 6-14
    - /RE • 6-14
    - /RON • 6-14
    - /RW • 6-14
    - /ZE • 6-14
  - Setup • 6-3
  - Stop • 6-15
  - Trace • 6-15
    - IO • 6-16
    - Logical Block Nos. • 6-15
    - Status Attribute • 6-16
    - Switch use • 6-16
    - Tasks • 6-15
  - Using • 6-4
  - Virtual disk
    - Additional changes • 1-9
  - Virtual Monitor Console Routine
    - See VMR
  - VMR
    - INSTALL command • B-5
    - REMOVE command • B-5
    - TIM Command • 1-8
    - updating a loadable driver • B-2
  - VMS ACP function support
    - FCS • 2-1
  - VSUN\$ directives • 3-35
- 

## W

---

Write operation  
no sync (communication driver) • 3-42

---

## X

---

XDT  
loadable support • 3-20

---

## Y

---

Year 2000  
ANSI Tape • 8-5  
Assemblers and Linkers • 8-7  
Backup • 8-6  
BASIC-PLUS-2 • 5-2  
Batch and Queue • 8-8  
BRU dates • 1-7  
Certification • 1-3, 8-9  
Components Modified • 1-6  
Datatrive-11 • 5-3  
date format • 1-4  
Date references • 1-8  
2 Digit years • 1-5

4 Digit years • 1-5  
DMP Header ID Dates • 1-8  
EDT Date Command • 1-7  
Enhancements • 1-4  
ERRLOG Date handling updated • 1-7  
Files-11 • 8-2  
FLX • 8-6  
DOS Year 2000 dates • 1-7  
RT-11 Year 2000 dates • 1-7  
Foreign tapes • 8-6  
FORTRAN-77 • 5-2  
Help • 1-8  
Inference base date • 1-5  
Inferencing 2 digit years • 1-3, 8-1  
INI Home block dates • 1-7  
Internal Date Storage • 8-2  
ISO Format Date • 1-6  
Macro-11 listings • 1-8  
ODS-1 • 8-2  
PIP Date Range comparison • 1-7  
QUEUE command modification • 1-7  
RPT Date Range Switch • 1-7  
SYSLIB • 1-15  
System management • 8-9  
TIM Command in VMR • 1-8  
Time command modification • 1-6





digital