

VMS Version 5.4 Release Notes

Order Number: AA-LB22C-TE

August 1990

This document describes changed software features, software problems and restrictions, and changes to documentation for Version 5.4 of the VMS operating system.

Revision/Update Information: This manual supersedes previous versions of VMS Release Notes.

Software Version: VMS Version 5.4

**digital equipment corporation
maynard, massachusetts**

August 1990

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.


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

© Digital Equipment Corporation 1990.

All Rights Reserved.
Printed in U.S.A.

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

The following are trademarks of Digital Equipment Corporation:

CDA	DEQNA	MicroVAX	VAX RMS
DDIF	Desktop-VMS	PrintServer 40	VAXserver
DEC	DIGITAL	Q-bus	VAXstation
DECdtm	GIGI	ReGIS	VMS
DECnet	HSC	ULTRIX	VT
DECUS	LiveLink	UNIBUS	XUI
DECwindows	LN03	VAX	
DECwriter	MASSBUS	VAXcluster	

The following are third-party trademarks:

Display PostScript is a trademark of Adobe Systems Incorporated.

Helvetica and Times are registered trademarks of Linotype AG.

IBM is a registered trademark of International Business Machines, Inc.

Macintosh is a trademark of Apple Computer, Inc.

MS-DOS is a registered trademark of Microsoft Corporation.

PostScript is a registered trademark of Adobe Systems Incorporated.

UNIX is a registered trademark of American Telephone & Telegraph Company.

X Window System, Version 11 and its derivations (X, X11, X Version 11, X Window System) are trademarks of the Massachusetts Institute of Technology.

ZK5482

Production Note

This book was produced with the VAX DOCUMENT electronic publishing system, a software tool developed and sold by Digital. In this system, writers use an ASCII text editor to create source files containing text and English-like code; this code labels the structural elements of the document, such as chapters, paragraphs, and tables. The VAX DOCUMENT software, which runs on the VMS operating system, interprets the code to format the text, generate a table of contents and index, and paginate the entire document. Writers can print the document on the terminal or line printer, or they can use Digital-supported devices, such as the LN03 laser printer and PostScript printers (PrintServer 40 or LN03R ScriptPrinter), to produce a typeset-quality copy containing integrated graphics.

72888

Contents

PREFACE	xxix
---------	------

CHAPTER 1 LICENSE MANAGEMENT	1-1
-------------------------------------	------------

1.1 REGISTERING YOUR LICENSES	1-1
-------------------------------	-----

1.2 REGISTERING A LICENSE USING THE COMMAND PROCEDURE VMSLICENSE.COM	1-3
---	-----

1.3 INFORMATION ON LICENSING FOR SERVICE CUSTOMERS	1-8
--	-----

1.4 MODIFYING LICENSE UNITS WITH THE LICENSE MANAGEMENT UTILITY	1-10
--	------

1.5 LICENSE MANAGEMENT FACILITY NOTES	1-14
---------------------------------------	------

1.6 VMS LICENSE TYPES	1-15
-----------------------	------

1.7 LICENSE ACTIVITY USE DEFINITION FOR VMS LICENSES	1-16
--	------

1.8 VAXCLUSTER LICENSE	1-16
------------------------	------

1.9 DECNET-VAX LICENSE	1-17
------------------------	------

1.10 VAX RMS JOURNALING LICENSE	1-17
---------------------------------	------

1.11 VMS VOLUME SHADOWING LICENSE	1-17
-----------------------------------	------

Contents

CHAPTER 2 GENERAL USER RELEASE NOTES		2-1
2.1	\$ENTRY — NEW SYMBOL	2-1
2.2	BATCH/PRINT FACILITY — PERFORMANCE IMPROVEMENTS	2-1
2.3	COMMAND PROCEDURES — RESTRICTION	2-2
2.4	DECWINDOWS NOTES	2-3
2.4.1	DECterm Terminal Emulator	2-3
2.4.1.1	Conformance Level Change for Support of Terminal State Reports • 2-3	
2.4.1.2	Graphics • 2-3	
2.4.1.3	Initializing DECterm • 2-4	
2.4.1.4	Keyboards and Languages • 2-5	
2.4.1.5	Memory — How to Reduce DECterm Use • 2-6	
2.4.1.6	Memory — Logout Problem • 2-6	
2.4.1.7	Memory — Resize Request Problem • 2-6	
2.4.1.8	PC Interoperability Restrictions • 2-6	
2.4.1.9	Text • 2-7	
2.4.1.10	Window Scrolling Problem • 2-7	
2.4.2	DECW\$COLOR Guidelines for Changes in Achromatic Color (GrayScale) Rendering	2-7
2.4.3	DECwindows Startup Problem	2-8
2.4.4	Desktop Applications	2-8
2.4.4.1	Calendar — Corrected Problems • 2-8	
2.4.4.2	Calendar — Restrictions and Limitations • 2-8	
2.4.4.3	Cardfiler — Scroll Bar • 2-9	
2.4.4.4	Clock — Changing the System Time • 2-9	
2.4.4.5	CDA Viewer — Corrections and New Functionality • 2-9	
2.4.4.6	CDA Viewer — Problems and Unexpected Behavior • 2-9	
2.4.4.7	CDA Viewer — Restrictions • 2-11	
2.4.4.8	DECwindows Mail — Corrected Problems • 2-11	
2.4.4.9	DECwindows Mail — Problems and Restrictions • 2-12	
2.4.4.10	DECwindows Mail — Problem Sending PostScript Files • 2-13	
2.4.4.11	Paint — Corrected Problems • 2-14	
2.4.4.12	Print Screen — LJ250 Problem Corrected • 2-14	
2.4.4.13	Print Screen — Restrictions • 2-14	
2.4.4.14	Print Screen — VAXstation 3520 and 3540 Correction • 2-14	
2.4.4.15	Print Screen — VAXstation 3520 and 3540 Restriction • 2-14	
2.4.5	FileView	2-14
2.4.5.1	Changes to FileView • 2-15	
2.4.5.2	Customizing FileView to View PostScript Files • 2-17	

2.4.5.3	FileView Restrictions • 2-17	
2.4.6	Help Widget Enhancement _____	2-18
2.4.7	Session Manager _____	2-18
2.4.7.1	Corrected Problems • 2-18	
2.4.7.2	User-Defined Logo Support Added • 2-20	
2.4.8	SYSGEN Parameter PQL_MPRCLM and Captive Accounts _____	2-20
2.4.9	ULTRIX — Running Applications from a Workstation _____	2-21
2.4.10	ULTRIX — Authorization Guidelines for DECwindows Applications _____	2-21
2.4.11	VT1000 DECwindows Terminal Support _____	2-21
2.4.12	Window Manager (Icon Box) — Corrected Problems _____	2-22
<hr/>		
2.5	DIGITAL COMMAND LANGUAGE (DCL) NOTES	2-22
2.5.1	ANALYZE/ERROR_LOG Command — New Error-Logging Format _____	2-22
2.5.2	DCL Command Verb and Qualifier Length _____	2-22
2.5.3	DCL Lexical Function F\$CONTEXT — Restriction Removed _____	2-23
2.5.4	DEFINE/FORM Command /SHEET_FEED Qualifier — Restriction _____	2-23
2.5.5	DELETE and PURGE Commands — Corrected Problems _____	2-23
2.5.6	ENDSUBROUTINE Command — Correct Usage _____	2-24
2.5.7	MOUNT Command — Correction to OPCOM Message _____	2-24
2.5.8	OPEN Command — Negating Problem Corrected _____	2-26
2.5.9	PRINT and SUBMIT Commands — Changes in Error Conditions _____	2-26
2.5.10	SET ACL Command Qualifiers — Clarifications and Enhancements _____	2-27
2.5.11	SET HOST/DTE Command — Modifications _____	2-28
2.5.12	SHOW LOGICAL/FULL Command — Correction _____	2-28
2.5.13	SHOW MAGTAPE Command Now Obsolete _____	2-28
2.5.14	SUBMIT/DELETE Command — Modifications _____	2-29
2.5.15	Symbol Names — Use Caution When Making Symbol Name Assignments _____	2-29
<hr/>		
2.6	DIGITAL STANDARD RUNOFF (DSR) — VARIANT NAME CHANGE	2-29
<hr/>		
2.7	DIRECTORY ENTRIES — PRIMARY AS OPPOSED TO ALIAS	2-29
<hr/>		
2.8	EDT EDITOR — TMP FILES RENAMING PROBLEM	2-30

Contents

2.9	ETHERNET/802 CONTROLLERS — NEW SUPPORT	2-31
2.10	PROCESS IDENTIFICATION (PID) — ALL SIGNIFICANT DIGITS MUST BE SPECIFIED	2-31
2.11	RA92 DSA DISK SUPPORT	2-32
2.12	SORT/MERGE UTILITY — CHANGES AND ENHANCEMENTS	2-32
2.13	SYSTEM MESSAGES — NEW AND MODIFIED FOR VMS VERSION 5.4	2-32
2.14	TA90E TAPE DRIVE SUPPORT	2-32
2.15	TAPE SUPPORT — ACCEPTANCE OF ANSI INITIALIZED MAGNETIC TAPES	2-33
2.16	TLZ04 TAPE DRIVE NOTES	2-33
2.16.1	TLZ04 Tape Drive Now Supported _____	2-33
2.16.2	TLZ04 Performance Issues _____	2-33
2.17	VAX 6000-SERIES COMPUTERS	2-33
2.18	VAX 9000-SERIES COMPUTERS	2-33
2.19	VAXft 3000 SYSTEM NOTES	2-34
2.19.1	VAXft 3000 System Messages Displayed _____	2-34
2.19.2	VAXft System Services — Required HELP Command Line _____	2-34
2.20	VAX TEXT PROCESSING UTILITY (VAXTPU) NOTES	2-34
2.20.1	/NOWORK Qualifier Problem _____	2-34
2.20.2	/WORK Qualifier Problem _____	2-35
2.20.3	Display Manager Definition Restriction _____	2-35
2.21	VMS MAIL UTILITY — FOLDER NAME PARAMETER NOW SUPPORTS MIXED CASE	2-35

CHAPTER 3	SYSTEM MANAGER RELEASE NOTES	3-1
3.1	\$CREATE_RDB SYSTEM SERVICE — NEW LOCATION FOR RIGHTS DATABASE	3-1
3.2	ANALYZE/DISK_STRUCTURE UTILITY NOTES	3-1
3.2.1	ANALYZE/DISK_STRUCTURE — Corrected Problems	3-1
3.2.2	Facility Code Change	3-2
3.3	AUDIT ANALYSIS UTILITY — /SELECT=STATUS=FAILURE QUALIFIER PROBLEM	3-3
3.4	AUTHORIZE UTILITY NOTES	3-3
3.4.1	Adding Proxy Accounts	3-3
3.4.2	Modifications to the RESTRICTED Flag	3-3
3.4.3	Setting the DISCTRLY Flag	3-4
3.4.4	Submitting Batch Jobs	3-4
3.5	AUTOGEN COMMAND PROCEDURE NOTES	3-4
3.5.1	Feedback Mechanism — Additional Data Files Used	3-4
3.5.2	Hexadecimal Values Processed Correctly in MODPARAMS.DAT	3-4
3.5.3	Mechanism to Control MSCP Server Buffer Size is Obsolete	3-5
3.5.4	OLDSITE Mechanism is Obsolete	3-5
3.5.5	Prefix ADD_ Can No Longer Subtract Values for Parameters Generated by FEEDBACK	3-5
3.5.6	System Files Not Marked for NOBACKUP	3-6
3.5.7	Swap File Size Changes	3-6
3.5.8	Switching Window Systems	3-6
3.6	BACKUP UTILITY NOTES	3-7
3.6.1	Backing Up Files Marked for Recovery Unit Journaling	3-7
3.6.2	Backup Utility Problems	3-8
3.6.3	Corrected Backup Utility Problems	3-8
3.6.4	Forced Error Handling	3-9
3.6.5	Image Save Operation Restriction	3-9
3.6.6	ODS-1 Disk Structures — Corrected Backup Problems	3-10
3.6.7	Postprocessing Backup Problem Corrected	3-10
3.6.8	Reading Multiple Save Sets from a TU81-PLUS Tape Drive	3-10
3.6.9	Standalone Backup — Saving a Bound Volume Set Not Supported	3-11
3.6.10	Using BACKUP with Compound Document Files	3-11

Contents

3.7	BATCH/PRINT FACILITY NOTES	3-11
3.7.1	Generic Queue Restriction Now Enforced	3-11
3.7.2	Print Symbiont Working Set Purge Less Frequent	3-11
3.7.3	START/QUEUE/MANAGER/BUFFER_COUNT Command Enhancement	3-12
3.7.4	START/QUEUE/TOP_OF_FILE Command — Line-Feed Improvement	3-12
3.7.5	Unsynchronized Cluster Time Affects SUBMIT/AFTER Command	3-12
<hr/>		
3.8	BOOTSTRAPPING — SPECIAL FILES HANDLING	3-13
<hr/>		
3.9	CLUSTER_CONFIG.COM CHANGES	3-14
3.9.1	CLUSTER_CONFIG.COM Now Requires VOLPRO Privilege	3-14
3.9.2	PAGEFILE=0 and SWAPFILE=0 No Longer Written to MODPARAMS.DAT	3-15
<hr/>		
3.10	CLUSTER I/O — SUBSYSTEM CHANGES	3-15
3.10.1	Disk Failover	3-15
3.10.2	MSCP Serving Third-Party Disks	3-15
<hr/>		
3.11	DEBUGGER NOTES	3-16
3.11.1	System Management Considerations	3-16
3.11.2	System Resources	3-17
3.11.3	User Quotas	3-17
<hr/>		
3.12	DECDTM SERVICES NOTES	3-17
3.12.1	Adjusting SYSGEN Parameters Before Using DECdtm Services	3-18
3.12.2	DECdtm Services — Startup Processes and How to Inhibit Them	3-18
3.12.3	Defining Logical Names Before Starting DECdtm Services	3-18
3.12.4	File Name Considerations for Transaction Log Files	3-18
3.12.5	File Size Considerations for Transaction Log Files	3-19
3.12.6	Image Files	3-19
<hr/>		
3.13	DECNET-VAX NOTES	3-20
3.13.1	Constraints on Passive Maintenance Functions Relaxed	3-20
3.13.2	DECnet-VAX Objects — New Characteristic Added	3-20
3.13.3	Downline Loading Correction	3-21
3.13.4	Executor Parameters	3-21
3.13.4.1	Executor Path Split Policy — INTERIM Policy Corrected	3-21

3.13.4.2	MAXIMUM LINKS — Maximum Value Increased • 3-22	
3.13.4.3	MAXIMUM PATH SPLITS Default Value • 3-22	
3.13.4.4	PIPELINE QUOTA — Default Value Changed • 3-22	
3.13.5	Local DTE Address Used on Outgoing Call _____	3-22
3.13.6	Maintenance Operations Module — Repeated Operations Error _____	3-22
3.13.7	NCP/NML Requires OPER Privilege to Obtain Service Passwords _____	3-23
3.13.8	NETACP\$BUFFER_LIMIT Logical Name — To Override Default BYTLM Quota _____	3-23
3.13.9	Network Control Program (NCP) — SET/DEFINE CIRCUIT Command _____	3-23
3.13.10	NML Checks for Illegal Address Configurations _____	3-23
3.13.11	Node-Level Access Control Problem Corrected _____	3-24
3.13.12	Proxy Access Parameters — Changes _____	3-24
3.13.13	Session Incompatibility with Phase IV Implementations _____	3-24
3.13.14	Support for X.25 Virtual Circuits Requirement _____	3-25
<hr/>		
3.14	DECWINDOWS NOTES	3-25
3.14.1	DECwindows Startup _____	3-26
3.14.2	DECwindows Startup for Upgrade Only _____	3-26
3.14.3	Tailoring DECwindows _____	3-26
3.14.4	Template File — New DECW\$SYLOGIN.TEMPLATE _____	3-27
3.14.5	Template File — Support for Configuring Multihead Systems _____	3-27
3.14.6	ULTRIX Connection (UCX) and TCP/IP Transport _____	3-27
3.14.6.1	Sequence Lost Errors • 3-27	
3.14.6.2	Starting UCX Before DECnet — Problem • 3-27	
3.14.7	X Servers — Interoperability with Other Vendors' X Servers _____	3-28
<hr/>		
3.15	DIGITAL DISTRIBUTED QUEUING SERVICE — CREATING DQS\$SERVER	3-28
<hr/>		
3.16	DISK HEADER SPACE PROBLEM	3-29
<hr/>		
3.17	DISMOUNT COMMAND — CHANGES REGARDING OPEN FILES	3-31
3.17.1	Closing Files _____	3-32
3.17.2	Clusterwide Support for DISMOUNT _____	3-33
3.17.3	Restoring the Previous Behavior of the DISMOUNT Command _____	3-34
<hr/>		
3.18	DOCUMENTATION — TITLE CHANGES FOR INSTALLATION AND UPGRADE MANUALS	3-35

Contents

3.19	DSDRIVER/LOCKMANAGER PROBLEM CORRECTED	3-35
3.20	DSSI DEVICE NAMING NO LONGER DEPENDENT ON SYSGEN PARAMETER VMS5	3-35
3.21	DUDRIVER AND DSDRIVER — CHANGE TO IMPROVE FAILOVER	3-37
3.22	DUMP FILE NOTES	3-37
3.22.1	Dump File Size Changes	3-37
3.22.2	SAVEDUMP Parameter and PAGEFILE.SYS Size — Interaction Caution	3-38
3.22.3	Selective Crash Dump Files — Caution	3-38
3.22.4	Shared Dump Files	3-39
3.23	ETHERNET NOTES	3-39
3.23.1	DEBNI Ethernet/802 Controller — New Support for the VAXBI Bus	3-39
3.23.1.1	I/O Interface • 3-39	
3.23.1.2	Node ID • 3-40	
3.23.2	DEMNA Ethernet/802 Controller — New Support for the XMI Bus	3-40
3.23.3	DEQNA Ethernet Adapter May Receive Corrupt Data	3-40
3.23.4	DEQTA Ethernet/802 Controller — New Support for the Q-bus	3-40
3.23.5	Ethernet Controllers — Tuning the VMS Operating System	3-41
3.23.5.1	Network Parameters — DEBNA Controllers • 3-41	
3.23.5.2	Network Parameters — DEBNA, DEBNI, and DEMNA Controllers • 3-41	
3.23.6	Second-Generation Ethernet Controller (SGEC) Now Supported	3-42
3.24	INITIALIZE COMMAND — DEFINING VOLUME SERIAL NUMBERS	3-42
3.25	MA780 (MULTIPOINT SHARED MEMORY)	3-43
3.26	MAGNETIC TAPE ACP NOTES	3-43
3.26.1	Correction to I/O Process	3-43
3.26.2	Undocumented Implementation Removed	3-43

3.27	MAIL UTILITY — CHANGES TO PRINT/QUEUE COMMAND	3-44
<hr/>		
3.28	MASS STORAGE CONTROL PROTOCOL (MSCP) SERVER	3-44
3.28.1	Controller Letters — Restriction	3-44
3.28.2	Diskette Devices — Some Not Allowed	3-44
<hr/>		
3.29	MODIFIED-PAGE WRITER — FLUSHING OF MODIFIED-PAGE LIST ELIMINATED	3-44
<hr/>		
3.30	MODULAR EXECUTIVE NOTES	3-45
3.30.1	Effects on Privileged Code	3-45
3.30.2	Effects on SYSGEN	3-45
3.30.3	Effects on System Management	3-45
3.30.3.1	SYS\$LOADABLE_IMAGES Directory on the System Disk • 3-46	
3.30.3.2	System Failure • 3-46	
3.30.4	Introduction to the Modular Executive	3-46
3.30.5	Version Numbers and Version Checking	3-46
<hr/>		
3.31	MONITOR UTILITY NOTES	3-48
3.31.1	Error in Display	3-48
3.31.2	I/O and RMS Problem Corrected	3-48
3.31.3	MONITOR CLUSTER Command Notes	3-48
3.31.3.1	Error Messages When Using MONITOR CLUSTER • 3-48	
3.31.3.2	Using MONITOR CLUSTER Might Introduce a Network Load • 3-49	
3.31.4	RMS Bucket and Multibucket Split Rates Invalid	3-49
<hr/>		
3.32	OPCOM CHANGES	3-49
3.32.1	Log File Operator Classes	3-50
3.32.2	OPCOM Default States	3-50
3.32.3	Overriding the OPCOM Default States	3-50
3.32.4	Removing Old Reply Commands — Requirement	3-52
<hr/>		
3.33	PRINTER EXECUTION QUEUE PROBLEM CORRECTED	3-52
<hr/>		
3.34	PSEUDOTERMINAL DRIVER	3-52

Contents

3.35	RQDX3 CONTROLLER NOTES	3-52
3.35.1	Device Unit Number Changed	3-53
3.35.2	Frequent Controller Resets — Restriction	3-53
<hr/>		
3.36	SECURITY FEATURES — NOTES	3-53
3.36.1	Department of Defense (DoD) Erase Pattern Corrected	3-53
3.36.2	NETCONFIG.COM Security Enhancements	3-54
3.36.3	OPCOM on Nonclustered MicroVAX Systems	3-55
3.36.4	Passwords — New Security Alarms	3-55
3.36.5	Reestablishing Security Environment	3-55
3.36.6	Security Audit Alarm Settings Preserved Between System Boots	3-56
3.36.7	Security-Auditing Failure Mode Settings Preserved Across Initializations	3-56
3.36.8	Suppressing Duplicate Logging of Security Alarms by OPCOM	3-56
3.36.9	SYSECURITY.COM Command Procedure — New Site-Specific Configuration File	3-57
3.36.10	SYSUAF Template File — Change	3-57
3.36.11	User Authorization File (UAF) Notes	3-58
3.36.11.1	Captive Accounts — Batch and Network Restrictions Removed • 3-58	
3.36.11.2	Captive Accounts — Security and Application PRINT Commands • 3-58	
3.36.11.3	Changes to the CAPTIVE Flag • 3-59	
3.36.11.4	DISIMAGE Flag • 3-62	
3.36.11.5	RESTRICTED Accounts — Incompatibility Problems Corrected • 3-62	
3.36.11.6	UAF Record Length Enforcement • 3-62	
3.36.11.7	UAF Template File Changes • 3-62	
<hr/>		
3.37	SECURITY REVIEW RECOMMENDED	3-63
<hr/>		
3.38	SET HOST (CTDRIVER/REMACP/RTPAD) NOTES	3-64
3.38.1	\$CANCEL is Asynchronous	3-64
3.38.2	CTDRIVER Enforces SETMODE/SENSEMODE Buffer Size	3-64
3.38.3	CTDRIVER's Output Buffering	3-64
3.38.3.1	Captive Command Procedures and Ctrl/Y • 3-65	
3.38.3.2	Ctrl/C, Ctrl/Y, and Out-of-Band Abort Character Processing • 3-65	
3.38.3.3	Extra Input Prompt Displayed Following Ctrl/C, Ctrl/Y, or an Out-of-Band Abort Character • 3-66	
3.38.3.4	Output Line Not In Sequence Following Ctrl/C, Ctrl/Y, or an Out-of-Band Abort Character • 3-66	
3.38.4	Ctrl/C Processing	3-67
3.38.5	New REMACP Image and RTTLOAD Command File	3-67

3.38.5.1	Running Without RTTDIVER • 3-67	
3.38.5.2	Setting the Maximum Number of Remote Users • 3-68	
<hr/>		
3.39	SHOW CLUSTER UTILITY — INCN_TIME FIELD NOW OBSOLETE	3-69
<hr/>		
3.40	SHUTDOWN.COM — CHANGE IN DISK DISMOUNT REPORTING	3-69
<hr/>		
3.41	STARTUP.COM — NEW SEQUENCE OF OPERATIONS	3-69
<hr/>		
3.42	SYSGEN UTILITY NOTES	3-70
3.42.1	RECNXINTERVAL Parameter _____	3-70
3.42.2	New PRIORITY_OFFSET Parameter _____	3-71
3.42.3	New SCSI_NOAUTO Parameter _____	3-71
3.42.4	Symmetric Multiprocessing (SMP) Not Supported on Uniprocessor Systems Running DECwindows Server _____	3-71
3.42.5	TAPE_ALLOCLS Parameter _____	3-72
<hr/>		
3.43	SYSLOA — PAGE FAULT PROBLEM CORRECTED	3-72
<hr/>		
3.44	SYSMAN UTILITY NOTES	3-72
3.44.1	DO Command — Changes to Symbol Substitution and DCL Verification _____	3-72
3.44.2	PARAMETERS SET/STARTUP Command — Problem Corrected _____	3-73
3.44.3	SET PROFILE Command — Problem Corrected _____	3-73
<hr/>		
3.45	SYSTARTUP_V5.COM — STARTING THE QUEUE MANAGER	3-73
<hr/>		
3.46	SYSTEM DISK SIZE RECOMMENDATION: 100 MBYTES	3-74
<hr/>		
3.47	SYSTEM DUMP ANALYZER (SDA) UTILITY — REQUIREMENTS FOR VIRTUALPAGECNT SYSTEM PARAMETER	3-74
<hr/>		
3.48	TA90E TAPE DRIVE — USAGE GUIDELINES	3-75
3.48.1	Identifying the TA90E with SHOW DEVICE _____	3-75
3.48.2	Limitation of Using BACKUP with the TA90E Tape Drive _____	3-76

Contents

3.49	TDRIVER.MAR — CORRECTIONS	3-76
3.50	UNIBUS — FLOATING INTERRUPT VECTOR CHANGE	3-76
3.51	UPGRADE NOTES	3-77
3.51.1	Layered Product Cautions	3-77
3.51.1.1	ALL-IN-1 Shareable Images Requirement for CDA Support • 3-78	
3.51.1.2	DIGITAL Distributed File Service (DECdfs) Update Requirements • 3-78	
3.51.1.3	VAX Public Access Communications Requirements • 3-78	
3.51.1.4	VAX TU70/72 Device Driver Not Compatible • 3-79	
3.51.1.5	VAX Workstation Software (VWS) Requirement • 3-79	
3.51.2	Rolling Upgrades from VMS Version 5.3 to Version 5.4	3-79
3.51.3	VAXstation 8000 Upgrade Unsupported	3-79
3.51.4	VMS Kits Provided on RX33 Diskettes	3-79
3.52	VAX 8000-SERIES COMPUTER NOTES	3-79
3.52.1	DMB32 Product Software Required for DMB32 Communications Controller	3-80
3.52.2	Halting a VAX 8530, 8550, 8700, or 8800 System	3-80
3.52.3	SET TIME/CLUSTER Command	3-80
3.52.4	SET TIME Command — Problem	3-80
3.52.5	VAX 8800 Systems Running SMP — Problem	3-80
3.52.6	VAXBI 5 Restriction	3-81
3.53	VAX 9000 COMPUTER — AUTOGEN WITH FEEDBACK REQUIREMENT	3-81
3.54	VAX COMPUTERS — VMS SUPPORT	3-82
3.55	VAXCLUSTER RECONFIGURATION TIME REDUCED	3-82
3.56	VAXft 3000 COMPUTER NOTES	3-82
3.56.1	\$GETDVI and F\$GETDVI — Use to Obtain VAXft 3000 Device Information	3-83
3.56.2	DELTA/XDELTA Utility — How to Bootstrap a VAXft 3000 System	3-83
3.56.3	System Dump Analyzer (SDA) Utility — How to Cause a VAXft 3000 System Failure	3-83

3.57	VAXSTATION 3520 AND 3540 COMPUTER NOTES	3-84
3.57.1	Console Support on the Graphics Terminal _____	3-84
3.57.2	Software Products Information _____	3-84
3.58	VMS EXECUTIVE NOTES	3-84
3.58.1	Changes to Process Paging File Assignment _____	3-84
3.58.2	Extended Working Set and Virtual Address Space Sizes _____	3-84
3.58.3	PAGEFILE.SYS — Changes _____	3-85
3.58.4	Paging File Recommendation _____	3-85
3.59	VMSINSTAL — DCLTABLES LOGICAL NAME NOT USED	3-85
3.60	VOLUME SHADOWING PHASE II NOTES	3-86
3.60.1	AUTOGEN Operations — Volume Shadowing Adjustment Required _____	3-86
3.60.2	Batch and Print Jobs — Reentering after Conversion _____	3-86
3.60.3	Hierarchical Storage Controller (HSC) Revision Levels Required _____	3-87
3.60.4	SHOW DEVICES Command _____	3-87
3.60.5	Specify a Virtual Unit Name to Obtain \$GETDVI FREEBLOCK Information _____	3-87
3.60.6	SYSGEN Parameter VMSD3 — Special Parameter to Control Failover _____	3-88
3.61	VWS WORKSTATIONS — SETTING OF MULTIPROCESSING SYSGEN PARAMETER	3-88
3.62	YFDRIVER TERMINAL PORT DRIVER SUPPORTS NEW BAUD RATE	3-88
CHAPTER 4 PROGRAMMER RELEASE NOTES		4-1
4.1	\$CANCEL IS AN ASYNCHRONOUS OPERATION	4-1
4.2	\$CHANGE_ACL SYSTEM SERVICE — LOCK CORRECTION	4-1
4.3	\$CMKRNL — FAILURE OF CUSTOMER-WRITTEN PROGRAMS	4-1

Contents

4.4	ACL EDITOR NOTES	4-2
4.4.1	Use of Protected Entries — Correction	4-2
4.5	ACTIVATING AN IMAGE WITH SYSTEM VERSION MISMATCH — CHANGE	4-2
4.6	DCL SUBROUTINE — MODIFICATIONS TO SUBROUTINE ENTRY POINT SCOPING	4-2
4.7	DCL SUBSTRING ASSIGNMENT PROBLEM CORRECTED	4-3
4.8	DEBUGGER NOTES	4-4
4.8.1	Corrected Problems or Restrictions	4-4
4.8.2	Debugger Problems or Restrictions	4-5
4.8.2.1	\$WAKE Call Followed by \$HIBER Call • 4-6	
4.8.2.2	Debugger Commands Disabled in DECwindows Interface • 4-6	
4.8.2.3	Examining LABEL[n] for a Code Location • 4-6	
4.8.2.4	MACRO Source Correlation Problem • 4-6	
4.8.2.5	Register Window in DECwindows Interface • 4-7	
4.8.2.6	RUN/DETACHED Command Entered After LINK/DEBUG Command • 4-7	
4.8.2.7	SET IMAGE Command Limitation • 4-7	
4.8.2.8	Using Concealed Rooted-Directory Logical Names for Source Files • 4-8	
4.8.2.9	Using Debugger Commands in DCL Command Procedures • 4-8	
4.8.2.10	Using the Abort Key or Stop Button After a SPAWN Command • 4-8	
4.8.2.11	Using the Debugger on a VAXstation Running VWS • 4-9	
4.8.2.12	Using the DECwindows Interface • 4-9	
4.8.2.13	Vector-Support Restrictions and Problems • 4-11	
4.8.2.14	Watchpoints in Installed Writable Shareable Images • 4-11	
4.8.3	Obsolete Debugger Commands	4-11
4.8.4	Use of Single-Process (Pre-Version 5.2) Debugging Configuration	4-12
4.9	DECNET — FILE ACCESS PROTOCOL EXTENSIONS	4-13
4.10	DECWINDOWS NOTES	4-13
4.10.1	CDA Toolkit Notes	4-13
4.10.1.1	Base Converters — Analysis Back End • 4-13	
4.10.1.2	Base Converters — PostScript Back End • 4-14	
4.10.1.3	Base Converters — Text Back End • 4-14	

4.10.1.4	Corrections to CDA Toolkit • 4-14	
4.10.1.5	New Attribute Rules for Files (Input Only) — UCX and VMS Services for PCS Servers • 4-15	
4.10.1.6	New Functionality Added to CDA Toolkit • 4-16	
4.10.1.7	New Messages to Clarify Errors • 4-16	
4.10.1.8	NEXT_AGGREGATE Routine — Correct Usage • 4-17	
4.10.1.9	REMOVE_AGGREGATE Routine — Correct Usage • 4-17	
4.10.2	DECterm Terminal Emulator _____	4-17
4.10.2.1	Color Table Report — Reporting Problem Corrected • 4-17	
4.10.2.2	CREATE/TERMINAL Command — Negative Values Problem Corrected • 4-18	
4.10.2.3	DECterm Fonts • 4-18	
4.10.2.4	ReGIS Locator Report — Omitted Coordinates • 4-18	
4.10.2.5	VT52-Mode Cursor Addressing — Restriction Removed • 4-18	
4.10.3	DECwindows Server and Driver Notes _____	4-18
4.10.3.1	Client-Number Problem Corrected • 4-19	
4.10.3.2	Misspelled Cursor Screen Boundary \$QIO Constant • 4-19	
4.10.3.3	Problems and Restrictions • 4-19	
4.10.3.4	Setting a Cursor Pattern \$QIO Call • 4-19	
4.10.3.5	Virtual Memory Space Problem with Large Pixmaps • 4-20	
4.10.3.6	VAXstation Configurations • 4-20	
4.10.4	Display PostScript Notes _____	4-21
4.10.4.1	Ada Bindings Not Available • 4-22	
4.10.4.2	VAX Calling Standard Bindings — DPS\$PRINTF Routine Not Implemented • 4-22	
4.10.5	SET DISPLAY Command _____	4-22
4.10.6	User Interface Language (UIL) Compiler Notes _____	4-22
4.10.6.1	Built-In Tables — Additions • 4-22	
4.10.6.2	Convenience Translation Files Made Public • 4-23	
4.10.6.3	Corrected Problems • 4-23	
4.10.6.4	Specifying Multiline Compound Strings • 4-24	
4.10.7	VAX C Definition File Requirements _____	4-24
4.10.8	Xlib Routines — Notes _____	4-25
4.10.8.1	Corrected Xlib AST Reentrance Problem • 4-25	
4.10.8.2	X\$DISPLAY_STRING Routines — Correction • 4-25	
4.10.8.3	Xlib Programming Restriction • 4-25	
4.10.9	XUI Toolkit Notes _____	4-25
4.10.9.1	Changes from VMS Version 5.1 to VMS Version 5.2 • 4-26	
4.10.9.2	Corrections to the XUI Toolkit • 4-26	
4.10.9.3	Discrepancies Between DECwindows Toolkit and MIT R3 Intrinsics • 4-26	
4.10.9.4	DRM Routines — Unavailable VAX Bindings • 4-26	
4.10.9.5	Font-Unit Values — Change in Properties • 4-26	
4.10.9.6	Internal Format of Compound Strings • 4-27	
4.10.9.7	List Box Dynamic Sizing • 4-27	
4.10.9.8	Problems and Restrictions in the XUI Toolkit • 4-27	

Contents

4.10.9.9	Redrawing Widgets • 4-29	
4.10.9.10	Selection Push Buttons • 4-29	
<hr/>		
4.11	DEVICE DRIVER DEBUGGING — POOLCHECK ENHANCEMENTS	4-30
<hr/>		
4.12	DEVICE SUPPORT NOTES	4-30
4.12.1	INVALIDATE_TB Macro — New Version _____	4-30
4.12.2	SPI\$MAP_BUFFER Macro — PRIO=HIGH Parameter Added ____	4-30
4.12.3	SYSGEN — New Order for Unit Control Blocks (UCBs) _____	4-31
<hr/>		
4.13	DIRECTORY SIZE LIMITATION REMOVED — EFFECT ON RSX-11 COMPATIBILITY MODE	4-31
<hr/>		
4.14	DSA DISK DRIVERS — ALTERNATE HOST INFORMATION CHANGE	4-31
<hr/>		
4.15	GBBDRIVER — NEW OUTPUT DRIVER	4-32
<hr/>		
4.16	I/O DEVICE DRIVER NOTES	4-32
4.16.1	Logical End-of-Volume Detection Now Always in Effect _____	4-32
4.16.2	Opening a Sequential-Media File Now More Efficient _____	4-32
4.16.3	User EOT Mode Correction _____	4-33
<hr/>		
4.17	IF-THEN-ELSE CONSTRUCT AND \$STATUS SYMBOL	4-33
<hr/>		
4.18	LANGUAGES — REINSTALLATION REQUIRED	4-33
<hr/>		
4.19	LIBRARIAN ROUTINES — LOCATE MODE CAUTION	4-34
<hr/>		
4.20	LINKER UTILITY — OPEN IMAGE LIBRARY SUPPORT	4-34
<hr/>		
4.21	MESSAGE ROUTER VERSION 3.0 INSTALLATION	4-34
<hr/>		
4.22	MODULAR EXECUTIVE NOTES	4-35
4.22.1	Instructions for Loading a Site-Specific Executive Loaded Image _____	4-36
4.22.2	New Description for \$MTACCESS _____	4-38

4.23	NATIONAL CHARACTER SET (NCS) CHANGE	4-38
<hr/>		
4.24	PROCESSOR REGISTER DEFINITION SYMBOLS	4-39
<hr/>		
4.25	RA92 DSA DISK — DEFINING SYMBOL DT\$_RA92	4-39
<hr/>		
4.26	RECORD MANAGEMENT SERVICES (RMS) NOTES	4-39
4.26.1	Appending to Shared Sequential Files — Restriction Removed	4-40
4.26.2	Expiration of RMS Files — Change	4-40
4.26.3	RAB\$_ASY Qualifier Now Supported for Process-Permanent Files	4-40
4.26.4	RMS Indexed File Local Buffers — New Default	4-41
4.26.5	RMS Statistics Restrictions	4-41
4.26.6	XAB\$_NUL Option — Clarification	4-41
<hr/>		
4.27	RECOVERY UNIT JOURNALING NOTES	4-42
4.27.1	Appending to Write-Shared Sequential Files	4-42
4.27.2	Exclusive Access to Recovery Unit Journalled Files — Restriction	4-42
4.27.3	Moving Recovery-Unit Journalled RMS Indexed Files to Systems Running VMS Version 4.7 and Earlier	4-42
4.27.4	SET FILE/AI_JOURNAL or SET FILE/BI_JOURNAL Command — Correct Usage	4-43
4.27.5	SYSGEN Parameter PIOPAGES — Change in Usage	4-43
4.27.6	VFC Format Sequential Files Partially Supported for Before-Image or Recovery Unit Journaling	4-43
4.27.7	WRTJNL_BIJ Error Message	4-44
<hr/>		
4.28	RUN-TIME LIBRARY (RTL) NOTES	4-44
4.28.1	LIB\$CREATE_VM_ZONE Routine — New Flags Added	4-44
4.28.2	LIB\$DECODE_FAULT Use with Vector Processor	4-44
4.28.3	LIB\$GET_VM Routine Performance Degradation	4-45
4.28.4	LIB\$VERIFY_VM_ZONE and LIB\$SHOW_VM_ZONE Zone Analysis Problem	4-45
4.28.5	Obsolete PPL\$ Routines and Replacements	4-46
4.28.6	PPL\$TRIGGER_EVENT Routine Memory Problem Corrected	4-46
4.28.7	RTL Language Support Enhancements	4-46
4.28.8	RTL Screen Management—SMG\$CREATE_PASTEBOARD and SMG\$CREATE_VIRTUAL_KEYBOARD Restriction	4-47
4.28.9	Scalar Math Routines — SYS\$SHARE:UVMTHRTL.EXE Now Obsolete	4-47

Contents

4.29	SELF-MODIFYING ITEM LISTS WITH \$GETXXX SERVICES	4-47
4.30	SET HOST/DTE/DIAL COMMAND — DMF-32 CONTROLLER PROBLEM	4-48
4.31	TLZ04 TAPE DRIVE — DEFINING SYMBOL DT\$_TLZ04	4-48
4.32	VAX 9000 COMPUTER — BI DEVICE DRIVERS CONFORMANCE REQUIREMENT	4-48
4.33	VAX ADA RUN-TIME LIBRARY NOTES	4-49
4.33.1	'STORAGE_SIZE Attribute Change	4-49
4.33.2	CALENDAR.SPLIT Improvement	4-49
4.33.3	CLOSE Procedures — Change in Implementation	4-49
4.33.4	CONSTRAINT_ERROR Now Raised in Place of NUMERIC_ERROR	4-49
4.33.5	Procedures to Improve AST Handling and Time Slicing	4-50
4.33.5.1	Ada SYSTEM_RUNTIME_TUNING Package •	4-50
4.33.5.2	EXPAND_AST_PACKET_POOL Procedure •	4-50
4.33.5.3	REQUEST_TIME_SLICE Procedure •	4-51
4.33.6	Restriction on END_OF_FILE Function	4-52
4.33.7	VAX Vector Registers Not Saved During Task Switches	4-53
4.34	VAX C NOTES	4-53
4.34.1	Mixing D_FLOAT and G_FLOAT Modules	4-53
4.34.2	VAX C Run-Time Library — Changes	4-53
4.34.3	VAX C Run-Time Library — Error Checking Enhanced	4-54
4.34.4	VAX C Run-Time Library — Socket Routines	4-54
4.35	VAX MACRO NOTES	4-55
4.35.1	Caution on Using NOP Instruction as a Delay Mechanism	4-55
4.35.2	Use of TIMEDWAIT Macro on MicroVAX and VAXstation 3100 Series Systems	4-56
4.35.3	VAX MACRO Corrected Problems	4-56
4.35.4	VAX MACRO Problems	4-56
4.35.5	VAX MACRO Product Requirements	4-59
4.36	VAX PASCAL RUN-TIME LIBRARY ENHANCEMENTS	4-59

4.37	VAX PL/I RUN-TIME LIBRARY CORRECTIONS AND ENHANCEMENTS	4-59
<hr/>		
4.38	XDELTA NOTES	4-60
4.38.1	Invoking XDELTA	4-60
4.38.1.1	Bootstrapping a VAX-11/730 or VAX-11/750 Computer Using the VMS Console TU58 • 4-61	
4.38.1.2	Bootstrapping XDELTA on a MicroVAX 2000, VAXstation 2000, MicroVAX 3400-Series, VAXstation 3520, VAXstation 3540, MicroVAX/VAXstation 3600-Series, MicroVAX 3900, VAX 4000 Model 300, MicroVAX II, or VAX-11/750 Computer • 4-62	
4.38.1.3	Bootstrapping XDELTA on a VAX-11/780 or a VAX-11/785 Computer • 4-63	
4.38.1.4	Bootstrapping XDELTA on a VAX 6000-Series Computer • 4-63	
4.38.1.5	Bootstrapping XDELTA on a VAX 8200, 8250, 8300, or 8350 Computer • 4-64	
4.38.1.6	Bootstrapping XDELTA on a VAX 8530, 8550, 8810 (8700), 8820, 8820-N (8800), 8830, or 8840 Computer • 4-65	
4.38.1.7	Bootstrapping XDELTA on a VAX 8600 or 8650 Computer • 4-65	
4.38.1.8	Bootstrapping XDELTA on a VAX 9000 Computer • 4-66	
4.38.1.9	Bootstrapping XDELTA on a VAXft 3000 Computer • 4-67	
4.38.1.10	Bootstrapping XDELTA on a VAXstation 3100 or MicroVAX 3100-Series Computer • 4-68	
4.38.2	Requesting an Interrupt	4-68
<hr/>		
CHAPTER 5	DOCUMENTATION RELEASE NOTES	5-1
<hr/>		
5.1	<i>GUIDE TO DECNET-VAX NETWORKING</i>	5-1
<hr/>		
5.2	<i>GUIDE TO SETTING UP A VMS SYSTEM</i>	5-1
<hr/>		
5.3	<i>GUIDE TO VMS FILE APPLICATIONS</i>	5-3
<hr/>		
5.4	<i>GUIDE TO VMS PROGRAMMING RESOURCES</i>	5-3
<hr/>		
5.5	<i>INTRODUCTION TO VMS SYSTEM SERVICES</i>	5-3
<hr/>		
5.6	<i>VAX MACRO AND INSTRUCTION SET REFERENCE MANUAL</i>	5-8

Contents

5.7	VMS AUTHORIZE UTILITY MANUAL	5-8
5.8	VMS DECNET TEST SENDER/DECNET TEST RECEIVER UTILITY MANUAL	5-8
5.9	VMS DECWINDOWS DESKTOP APPLICATIONS GUIDE	5-9
5.10	VMS DECWINDOWS GUIDE TO APPLICATION PROGRAMMING	5-9
5.11	VMS DECWINDOWS PROGRAMMING DOCUMENTATION SUPPLEMENT	5-12
5.12	VMS DECWINDOWS USER INTERFACE LANGUAGE REFERENCE MANUAL	5-12
5.13	VMS DECWINDOWS XLIB PROGRAMMING VOLUME	5-13
5.14	VMS I/O USER'S REFERENCE MANUAL: PART I	5-14
5.15	VMS LICENSE MANAGEMENT UTILITY MANUAL	5-14
5.16	VMS LINKER UTILITY MANUAL	5-14
5.17	VMS MONITOR UTILITY MANUAL	5-14
5.18	VMS NETWORK CONTROL PROGRAM MANUAL	5-15
5.19	VMS NETWORKING MANUAL	5-15
5.20	VMS RECORD MANAGEMENT SERVICES MANUAL	5-16
5.21	VMS RTL LIBRARY (LIB\$) MANUAL	5-17
5.21.1	LIB\$ADAWI Routine _____	5-17
5.21.2	LIB\$CREATE_VM_ZONE Routine _____	5-17
5.21.3	LIB\$SPAWN Routine _____	5-17
5.21.4	LIB\$SYS_TRNLOG Routine _____	5-18
	LIB\$SYS_TRNLOG	5-19

5.22	VMS RTL SCREEN MANAGEMENT (SMG\$) MANUAL	5-22
5.23	VMS SYSTEM MANAGER'S MANUAL	5-22
5.24	VMS SYSTEM SERVICES REFERENCE MANUAL	5-22
5.24.1	\$ASCEF Service	5-22
5.24.2	\$ASCTIM Service	5-23
5.24.3	\$BINTIM Service	5-23
5.24.4	\$CANCEL Service	5-23
5.24.5	\$CHANGE_ACL Service	5-23
5.24.6	\$CHKPRO Service	5-24
5.24.7	\$CMKRNL Service	5-25
5.24.8	\$CRELNM Service	5-25
5.24.9	\$CREMBX Service	5-25
5.24.10	\$CREPRC Service	5-26
5.24.11	\$CRMPSC Service	5-26
5.24.12	\$DCLCMH Service	5-27
5.24.13	\$DEQ Service	5-27
5.24.14	\$DGBLSC Service	5-28
5.24.15	\$ENQ Service	5-29
5.24.16	\$FAO Service	5-29
5.24.17	\$FORMAT_ACL Service	5-30
5.24.18	\$GETDVI Service	5-30
5.24.19	\$GETQUI Service	5-30
5.24.20	\$GETSYI Service	5-31
5.24.21	\$GETUAI Service	5-35
5.24.22	\$MOUNT Service	5-35
5.24.23	\$NUMTIM Service	5-36
5.24.24	\$PARSE_ACL Service	5-36
5.24.25	\$PUTMSG Service	5-36
5.24.26	\$QIO Service	5-38
5.24.27	\$SETDDIR Service	5-38
5.24.28	\$SETEXV Service	5-38
5.24.29	\$SETIMR Service	5-39
5.24.30	\$SETPRV Service	5-39
5.24.31	\$SETSFM Service	5-39
5.24.32	\$SETSSF Service	5-39
5.24.33	\$SETUAI Service	5-39
5.24.34	\$SNDJBC Service	5-39
5.24.35	\$SNDOPR Service	5-40
5.24.36	\$UNWIND Service	5-40
5.24.37	\$UPDSEC Service	5-40
5.24.38	\$WFLOR Service	5-41
5.24.39	Error Messages Referring to Shared Memory	5-41

Contents

5.24.40	Self-Modifying Item Lists	5-41
5.25	<i>VMS UPGRADE AND INSTALLATION SUPPLEMENT: VAXSTATION 2000, MICROVAX 2000</i>	5-42
5.26	<i>VMS UPGRADE AND INSTALLATION SUPPLEMENT: VAX 8600, 8650</i>	5-42
5.27	VMS UPGRADE AND INSTALLATION SUPPLEMENTS	5-43
5.28	<i>VMS UTILITY ROUTINES MANUAL</i>	5-43
5.29	<i>VMS VERSION 5.3 NEW FEATURES MANUAL</i>	5-43
5.30	<i>VMS VERSION 5.4 NEW FEATURES MANUAL</i>	5-43
APPENDIX A NEW AND MODIFIED SYSTEM MESSAGES FOR VMS VERSION 5.4		A-1
A.1	VMS SYSTEM ERROR MESSAGES	A-1
APPENDIX B VMS DECWINDOWS PERFORMANCE CONSIDERATIONS		B-1
B.1	RECOMMENDED MINIMUM MEMORY CONFIGURATIONS FOR DECWINDOWS	B-1
B.2	RUNNING VMS DECWINDOWS APPLICATIONS REMOTELY	B-1
B.3	SUGGESTION FOR RUNNING APPLICATIONS REMOTELY	B-2
B.4	USING AUTOGEN	B-2
B.5	IMPROVING DECWINDOWS MEMORY SHARING	B-3

APPENDIX C BUILDING AND COPYING A VMS SYSTEM DISK		C-1
C.1	INTRODUCTION	C-1
C.2	BUILDING THE OPERATING SYSTEM ON ANOTHER DISK	C-1
C.3	COPYING THE OPERATING SYSTEM FILES TO ANOTHER DISK	C-3
C.4	ADDING AN ALTERNATE SYSTEM ROOT DIRECTORY	C-4

APPENDIX D FILE DESIGN ATTRIBUTES		D-1
D.0.1	File Design Attributes	D-1
D.0.1.1	Initial File Allocation • D-1	
D.0.1.2	Contiguity • D-1	
D.0.1.3	Extending a File • D-2	
	D.0.1.3.1 Auto-Extend Size Selection • D-2	
	D.0.1.3.2 Establishing Auto-Extend Default Quantities • D-4	
	D.0.1.3.3 Placement and Contiguity of Extends • D-6	
D.0.1.4	Truncating a File • D-7	
D.0.1.5	Units of I/O • D-7	
D.0.1.6	Multiple Areas for Indexed Files • D-8	
D.0.1.7	Bucket Fill Factor for Indexed Files • D-8	

INDEX

EXAMPLES

1-1	Sample Product Authorization Key (PAK)	1-2
2-1	Mounting a Foreign Tape	2-25
2-2	Mounting a Foreign Disk	2-25
2-3	Operator Request Message from MOUNT Command	2-25
2-4	Corrected Operator Request Message from MOUNT Command	2-26
3-1	Using the SET OBJECT OUTGOING CONNECT PRIVILEGES Command	3-21
3-2	Example of a Repeated Operations Error	3-23
3-3	Using Count Parameter in a Maintenance Operation	3-23
4-1	Sample Output of Routine LIB\$SHOW_VM_ZONE	4-45

Contents

5-1	Freeing Compound Strings in the Hello World! Sample Application _____	5-10
5-2	Freeing Memory Allocated by the VMS SET ARG Routine ____	5-11

TABLES

1-1	License Unit Requirement Table (LURT) _____	1-12
2-1	DECterm Gray Levels _____	2-4
2-2	Session Manager Logical Names _____	2-19
2-3	SET ACL Command Qualifiers — Corrected Problems and Enhancements _____	2-27
3-1	DECdtm Images _____	3-19
3-2	Patches for DECnet Compatibility Problems _____	3-25
3-3	KFQSA Controller Device Names _____	3-36
3-4	RECNXINTERVAL Values _____	3-71
3-5	Layered Products with Special Requirements for VMS Version 5.4 _____	3-77
3-6	VMS Versions and Associated VAX Computers _____	3-82
3-7	Information Provided by \$GETDVI and F\$GETDVI _____	3-83
4-1	Additional UIL Resources _____	4-22
4-2	New Flags Added to LIB\$CREATE_VM_ZONE _____	4-44
4-3	VAX C Run-Time Library Socket Routines _____	4-54
4-4	Valid XDELTA Values for the BOOT Command Qualifier _____	4-61
4-5	XDELTA BOOT Command Qualifiers for VAX 6000-Series Computers _____	4-64
4-6	BOOT Command Qualifiers for VAX 9000 Computers _____	4-67
5-1	Arguments Unsupported by the UIL Compiler _____	5-13

Preface

This manual describes software enhancements and corrections in Version 5.4 of the VMS operating system.

These release notes supersede all release note documentation for previous versions of the VMS operating system. They include or update any previous release notes that are still pertinent to the Version 5.4 release. The source for each note is indicated by a marginal label designating the version number of the original release; for example, "V5.2."

Note: The VMS Version 5.3-2 release provided hardware support for the VAX 4000 Model 300 and the TA90E tape drive. VMS Version 5.3-2 was supplied only to those customers who purchased this new hardware.

For information about the new features included in VMS Version 5.4, see the *VMS Version 5.4 New Features Manual*.

Intended Audience

This manual is intended for all system users. Read this manual before you install, upgrade, or use Version 5.4 of the VMS operating system.

Document Structure

This manual contains the following chapters and appendixes:

- Chapter 1 contains information about license management.
- Chapter 2 contains release notes intended for general users.
- Chapter 3 contains release notes intended for system managers.
- Chapter 4 contains release notes intended for programmers.
- Chapter 5 contains additions and corrections to manuals in the VMS documentation set.
- Appendix A contains new and modified system messages for VMS Version 5.4.
- Appendix B contains DECwindows performance considerations.
- Appendix C contains replacement text for the section entitled "Building and Copying a VMS System Disk" in the *Guide to Setting Up a VMS System*.
- Appendix D contains replacement text for Section 3.2.1 of the *Guide to VMS File Applications*.

Preface

Conventions

The following conventions are used in this manual:

mouse	The term <i>mouse</i> is used to refer to any pointing device, such as a mouse, a puck, or a stylus.
MB1, MB2, MB3	MB1 indicates the left mouse button, MB2 indicates the middle mouse button, and MB3 indicates the right mouse button. (The buttons can be redefined by the user.)
PB1, PB2, PB3, PB4	PB1, PB2, PB3, and PB4 indicate buttons on the puck.
SB1, SB2	SB1 and SB2 indicate buttons on the stylus.
Ctrl/x	A sequence such as Ctrl/x indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
PF1 x	A sequence such as PF1 x indicates that you must first press and release the key labeled PF1, then press and release another key or a pointing device button.
Return	In examples, a key name is shown enclosed in a box to indicate that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)
...	In examples, a horizontal ellipsis indicates one of the following possibilities: <ul style="list-style-type: none">• Additional optional arguments in a statement have been omitted.• The preceding item or items can be repeated one or more times.• Additional parameters, values, or other information can be entered.
.	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
()	In format descriptions, parentheses indicate that, if you choose more than one option, you must enclose the choices in parentheses.
[]	In format descriptions, brackets indicate that whatever is enclosed within the brackets is optional; you can select none, one, or all of the choices. (Brackets are not, however, optional in the syntax of a directory name in a file specification or in the syntax of a substring specification in an assignment statement.)
{ }	In format descriptions, braces surround a required choice of options; you must choose one of the options listed.

red ink	<p>Red ink indicates information that you must enter from the keyboard or a screen object that you must choose or click on.</p> <p>For online versions of the book, user input is shown in bold.</p>
boldface text	<p>Boldface text represents the introduction of a new term or the name of an argument, an attribute, or a reason.</p> <p>Boldface text is also used to show user input in online versions of the book.</p>
<i>italic text</i>	<p><i>Italic text</i> represents information that can vary in system messages (for example, Internal error <i>number</i>).</p>
UPPERCASE TEXT	<p>Uppercase letters indicate that you must enter a command (for example, enter OPEN/READ), or they indicate the name of a routine, the name of a file, the name of a file protection code, or the abbreviation for a system privilege.</p>
-	<p>Hyphens in coding examples indicate that additional arguments to the request are provided on the line that follows.</p>
numbers	<p>Unless otherwise noted, all numbers in the text are assumed to be decimal. Nondecimal radices—binary, octal, or hexadecimal—are explicitly indicated.</p>

1

License Management

V5.4

This chapter contains supplemental information to the *VMS License Management Utility Manual* provided with VMS Version 5.4. Although most of the information in this chapter pertains to managing VMS and System Integrated Product (SIP) licenses, some of the information pertains to managing layered product licenses.

Note: For VMS Version 5.4, you *must* register your VAXcluster license before you can use your VAXcluster software.

VMS VAXcluster users: See Section 1.8 for important information.

1.1

Registering Your Licenses

V5.4

This section provides an overview of the license registration procedure.

Before VMS Version 5.0, you were required to install keys for certain products. For example:

- If you had a MicroVAX computer, you installed the VMS multiuser key.
- If you had any System Integrated Products (SIPs) such as DECnet or VAX Volume Shadowing, you used keys that were shipped on separate distribution media.

For VMS Version 5.0 and subsequent versions, these products and many other products are enabled only when you register their license information in the LICENSE database and then activate the license. Products may limit use or produce error messages if you fail to register and activate a license.

After you install the VMS operating system, you must register a VMS license. A VMS license lets you use the VMS operating system. You must also register the licenses for any of the following SIPs you have purchased:

- VAXclusters
- DECnet-VAX
- VAX RMS Journaling
- VAX Volume Shadowing

In addition to VMS and the SIPs, many layered products that run on VMS Version 5.4 also require license registration. See the *VMS License Management Utility Manual* for a full explanation of license registration.

Note: If you upgrade to VMS Version 5.4, there is no need to reregister licenses for VMS and SIPs.

License Management

1.1 Registering Your Licenses

After installing the VMS operating system, register your licenses in the following order:

- 1 Register the VMS license for the VAX computer on which you have just installed the VMS operating system. If you have a VAXcluster environment, you then register a VMS license for each additional VAX computer in the cluster. Each registered VMS license is assigned to one of the nodes for the cluster.
- 2 Register the licenses for any SIPs that you purchased. (If you upgraded to VMS Version 5.4, there is no need to reregister licenses for VMS and SIPs.)

Section 1.2 describes how to respond to the prompts from the command procedure `SYS$UPDATE:VMSLICENSE.COM`. You can use this procedure to register a license for any Digital product that supplies a PAK.

You can also register licenses with the `LICENSE REGISTER` command. See the *VMS License Management Utility Manual* for examples of license registration using `VMSLICENSE.COM` and `LICENSE REGISTER` commands. The manual, which is a part of the VMS Version 5.4 Base Documentation Set, provides details about all `LICENSE` commands, error messages, and recovery procedures for licensing tasks.

1.2 Registering a License Using the Command Procedure VMSLICENSE.COM

V5.4

To register each license, use the following procedure:

- 1 If you have not already done so, log in to the `SYSTEM` account. At the dollar sign (\$) prompt, enter the following command:

```
$ @SYS$UPDATE:VMSLICENSE 
```

The procedure displays the following menu:

VMS License Management Utility Options:

1. Register a Product Authorization Key
2. Amend an existing Product Authorization Key
3. Cancel an existing Product Authorization Key
4. List Product Authorization Keys
5. Modify an existing Product Authorization Key
9. Exit this procedure

Type '?' at any prompt for a description of the information requested.

Enter one of the above choices [1]

- 2 Type 1 and press Return. The procedure displays the following message:

```
* Do you have your Product Authorization Key? [YES]
```

Make sure you have a PAK for the license you are registering, then type Y and press Return.

License Management

1.2 Registering a License Using the Command Procedure VMSLICENSE.COM

3 The procedure displays the following information and prompts:

The REGISTER option allows you to add a new license to a license database. A Product Authorization Key (PAK) provides the product name and information you need to register the license. You must enter all the information provided by your PAK exactly as specified.

PAK ID:

Issuer [DEC]

Authorization Number []

- a. If Digital is the issuer, press Return. Otherwise, enter the issuer name on the PAK.
- b. Enter the authorization number that appears on the PAK and press Return.

4 The procedure asks for the following information:

PRODUCT ID:

Product Name []

Producer [DEC]

- a. Enter the product name that appears on the PAK and press Return.
- b. Press Return to specify Digital (DEC) as the producer.

5 The procedure asks for the number of units:

NUMBER OF UNITS:

Number of Units []

If the PAK lists the number of units, enter the number and press Return.

If the PAK does not list the number of units, do not enter a value. Press Return and go to the next step.

6 The procedure asks for the key level:

KEY LEVEL:

Version []

- a. If the PAK lists a version number, enter the number. Press Return and go to step 7.
- b. If the PAK does not list a version number, do not enter a value. Press Return. If you do not enter a version number, the procedure asks for the product release date:

Product Release Date []

Enter the product release date that appears on the PAK and press Return.

7 The procedure asks for the key termination date:

KEY TERMINATION DATE:

Key Termination Date []

If the PAK lists a key termination date, enter the date and press Return.

License Management

1.2 Registering a License Using the Command Procedure VMSLICENSE.COM

If the PAK does not list a key termination date, do not enter a value. Press Return and go to the next step.

8 The procedure asks for the following information:

RATING:

Availability Table Code []
Activity Table Code []

Usually, a PAK lists information for either the availability table code or the activity table code.

- a. If the PAK lists an availability table code value or the word **CONSTANT** followed by an integer, enter the information and press Return. If the PAK does not list this information, do not enter a value. Press Return after the prompt.
- b. If the PAK lists an activity table code value or the word **CONSTANT** followed by an integer, enter the information and press Return. If the PAK does not list this information, do not enter a value. Press Return after the prompt.

9 The procedure asks for the following information:

MISCELLANEOUS:

Key Options []

- a. If the PAK does not give values for this item, press Return and go to the next step.

If the PAK gives values for this item, enter the values and press Return. For example, if you are running the VMS operating system in a VAXcluster environment, the PAK lists **MOD_UNITS** and **NO_SHARE** as key options. Enter these options and press Return:

Key Options [] MOD_UNITS,NO_SHARE

- b. If you are registering a VMS license, the procedure displays the following message:

```
This Product Authorization Key has been provided with the NO_SHARE
option. This requires that this key be restricted to a specific
node within a cluster.
```

```
Enter the node name of the VAXcluster member to which this license
is restricted.
```

```
Is this PAK restricted to a cluster member node? [YES]
```

If you press Return, the procedure asks for the name of the node to which the license is restricted. The display includes the current node name as the default:

```
Node this PAK is restricted to (SCS node name) [NNAME]
```

If you are registering this PAK for the node you are currently logged in to, press Return. Otherwise, enter the node name of the VAX computer for which you are registering this PAK and press Return. Use the same name that you intend to use for the **SYSGEN** parameter **SCSNODE**.

License Management

1.2 Registering a License Using the Command Procedure VMSLICENSE.COM

10 The procedure asks for the following information:

```
Product Token []  
Hardware-Id []
```

If the PAK lists values for product token and hardware ID, enter the values and press Return. Otherwise, press Return.

11 The procedure prompts for the checksum:

```
Checksum []
```

Enter the checksum given on the PAK and press Return.

Note: Currently, the checksum string always begins with the number 1, which is the only number in the string. The other 16 positions are always alphabetic characters from A through P.

12 The procedure displays the information that you entered. For example:

```
License Database File: SYS$COMMON:[SYSEXE]LMF$LICENSE.LDB  
    Issuer: DEC  
Authorization: USA126087  
    Producer: DEC  
Product Name: VAX-VMS  
    Units: 460  
    Date:  
    Version: 5.4  
Termination Date: 28-SEP-1990  
Availability: E  
Activity:  
Options: MOD_UNITS,NO_SHARE  
Token:  
Hardware ID:  
Checksum: 1-ADEB-DOCJ-NENC-KDBM
```

This authorization key is restricted to: NNAME

Is this information correct? [YES]

Carefully compare the information on the screen with the information on the PAK. If the information is correct, type Y and press Return.

If the information is incorrect, type N and press Return.

Note: If you enter the information incorrectly, an error message is displayed and your license is not registered. Keep in mind that a CHECKSUM error can result when you enter incorrect information for the other items on the PAK. If an error message is displayed, carefully check all the data that you entered.

When you indicate that the information is incorrect, the procedure displays the following question:

```
Do you wish to make corrections? [YES]
```

To correct the data, type Y and press Return.

13 If you choose to make corrections, the procedure takes you through the questions again but supplies the data you entered previously as defaults for each data field. Each time the procedure displays correct information, press Return. When the procedure displays incorrect information, enter the correct data and press Return. To cancel

License Management

1.2 Registering a License Using the Command Procedure VMSLICENSE.COM

the current data without entering new data, type the backslash (\) character and then press Return.

If you entered the information correctly, the procedure displays a confirmation:

```
Registering VAX-VMS license in SYS$COMMON:[SYSEXE]LMF$LICENSE.LDB...
```

If you entered some information incorrectly but did not type Y to indicate that you wanted to make corrections, the procedure displays:

```
Registering VAX-VMS license in SYS$COMMON:[SYSEXE]LMF$LICENSE.LDB...
%LICENSE-F-BADCHK, checksum does not validate
```

```
Do you wish to make corrections? [YES]
```

To correct the data, type Y and press Return.

If you enter an incorrect checksum string, the procedure responds as follows:

```
1-ADEB-DOCJ-NENC-KDEX is not a valid license checksum string.
```

The license checksum is a 17-character verification string created by the PAK issuer for each PAK. The checksum string is presented in the format n-cccc-cccc-cccc-cccc, where n is an integer and c is a character from A through P. A PAK presents the checksum string with hyphen (-) characters for readability. Because the LMF does not count them for authorization, you can leave them out. Otherwise, you must enter the checksum string exactly as specified as your PAK or Product Authorization Amendment (PAAM).

If a default value is displayed and you wish to use it just press the RETURN key. The license checksum is a required field for the REGISTER and AMEND options.

```
Checksum []
```

Enter the correct checksum at the prompt and press Return.

- 14 After the license is successfully registered, the procedure asks whether you want to activate the license on the current node:

```
Do you want to LOAD this license on this system? [YES]
```

If you registered the PAK on a standalone system and you want to make the software available (active) immediately, type Y and press Return.

If you registered the license in a VAXcluster environment but do not want to make it available (active) on the current node, type N and press Return. After you exit this procedure, you can enter a LICENSE LOAD command to activate the license on a different node.

License activation is a process that makes a registered license known to a system. For details about license activation in a VAXcluster environment, see the *VMS License Management Utility Manual*.

If you type Y and the license is activated successfully, the procedure displays an informational message and returns you to the first menu and prompt as follows:

License Management

1.2 Registering a License Using the Command Procedure VMSLICENSE.COM

%LICENSE-I-LOADED, DEC VAX-VMS was successfully loaded with 460 units

VMS License Management Utility Options:

1. Register a Product Authorization Key
2. Amend an existing Product Authorization Key
3. Cancel an existing Product Authorization Key
4. List Product Authorization Keys
5. Modify an existing Product Authorization Key

9. Exit this procedure

Type '?' at any prompt for a description of the information requested.

Enter one of the above choices [1]

15 To register another PAK, type 1 or press Return. Then respond to the questions, again entering information from a PAK. Note that this time the procedure provided a default option [1] with the first menu. The procedure automatically supplies the data you previously entered when each question is asked a second time. When you register a second PAK, you need only enter data that is different from the last data entered. Some information, such as table codes and options, might be common to multiple PAKs. You can save typing time by registering all your PAKs during one VMSLICENSE session.

16 To exit the procedure, type 9 and press Return. The procedure finishes and returns you to the dollar sign (\$) prompt.

If you received an error message when the procedure attempted to load a license, this error does not affect the license registration, only license activation. Read the sections of the *VMS License Management Utility Manual* that describe activating a license. For example, read the LICENSE LOAD command description.

Note: If you are registering your licenses as part of a VMS operating system installation, refer to the installation guide that came with your VAX computer. This guide contains information on additional tasks you need to complete (such as running UETP) before using the system.

1.3 Information on Licensing for Service Customers

V5.4

If you are a VMS Service Customer and you have a VMS Service Update Kit, read this section for information on licensing. A VMS Service Update Kit includes a Warranty Mandatory Update. If you do not have a VMS Service Update Kit, the information in this section does not apply to you.

During both the VMS Version 5.4 installation procedure and the VMS Version 5.4 upgrade procedure, you install the Version 5.4 Warranty Mandatory Update (WMUP). The WMUP automatically runs a command procedure named LMF\$CONFIG.COM that performs the following functions for each VAX computer in your configuration that is supported by versions of VMS prior to VMS Version 5.0:

- Creates a system-specific LICENSE database named SYS\$SPECIFIC:[SYSEXE]LMF\$SYSTEM.LDB. If you are upgrading,

License Management

1.3 Information on Licensing for Service Customers

any previous versions of this database are deleted (if you accidentally modified this database, the modifications are lost).

- Determines the appropriate VMS Service Update PAK (SUP) for your system, registers it in LMF\$SYSTEM.LDB, then displays a confirmation message upon the successful completion of the registration process.
- During a cluster upgrade, LMF\$CONFIG.COM runs on the other members of the cluster as they reboot after the upgrade. If you add a new member to the cluster, CLUSTER_CONFIG automatically runs LMF\$CONFIG.COM.

The procedure LMF\$CONFIG might ask you to supply information. For example:

- When you run LMF\$CONFIG.COM on a MicroVAX processor, the procedure asks you to specify the number of users for which you are licensed. Be sure to specify the choice that represents the numbers of users on your VMS license.
- If you have not assigned a system communications services (SCS) node name to your computer, LMF\$CONFIG.COM asks you to supply it. Because all VMS SUPs must be associated with a particular VAX computer, LMF\$CONFIG.COM attempts to determine the computer's SCS node name. The SCS node name is defined by the SYSGEN parameter SCSNODE. If the procedure cannot determine the SCS node name, you are asked to supply it.

You do not need an SCS node name for a standalone VAX computer. However, if this computer becomes a part of a VAXcluster environment, you must enter the correct SCS node name. Otherwise the License Management Facility (LMF) cannot activate the VMS SUP on the VAX computer in the cluster.

If you change the SCS node name after LMF\$CONFIG.COM runs, you must modify the license. From the node on which you changed the name, enter a command in the following format:

```
LICENSE MODIFY VAX-VMS /INCLUDE=new_node_name-  
/DATABASE - SYS$SPECIFIC:[SYSEXE]LMF$SYSTEM.LDB
```

Note that LMF\$CONFIG.COM generates these licenses automatically. You do not need to manually register licenses for computers released prior to VMS Version 5.0.

However, for any VAX computer released since VMS Version 5.0, you must manually register the VMS PAK provided with your VMS kit by using VMSLICENSE.COM or the LICENSE REGISTER command. VMSLICENSE.COM and the LICENSE REGISTER command register licenses in the default common LICENSE database SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB. For a complete list of the versions of the VMS operating system and the VAX computers these versions first supported, see Table 3-6.

License Management

1.3 Information on Licensing for Service Customers

For example, assume that your VAXcluster environment contains a VAX 6320 computer. The VAX 6320 computer was released after VMS Version 5.0. If you have not manually registered your VMS PAK for the VAX 6320 computer, when LMF\$CONFIG.COM runs, it displays the following message:

The LMF\$CONFIG.COM procedure does not support the VAX 6320.

This procedure was developed to transparently register VAX-VMS Product Authorization Keys (PAKs), for systems released prior to VMS Version V5.0.

Please use the SYS\$UPDATE:VMSLICENSE.COM or LICENSE commands to register your VAX-VMS PAK. These procedures are documented in the VMS License Management Utility Manual.

You must register the VMS PAK provided with your newer VAX computer. When you register a VMS PAK manually, be sure to assign it to the proper SCS node name as documented with VMSLICENSE.COM or the LICENSE MODIFY command. You do not need to reregister a VMS PAK when you upgrade to future versions of VMS.

You must also use VMSLICENSE.COM or the LICENSE REGISTER command to register your VAXcluster SUP and the PAKs for any other products you have. These licenses are also stored in LMF\$LICENSE.LDB. (Note that LMF\$CONFIG.COM never modifies LMF\$LICENSE.LDB.)

Each time a system starts up, LMF activates all licenses from LMF\$LICENSE.LDB and LMF\$SYSTEM.LDB. License activation is a process that makes a license known to the current computer by loading information into the computer's memory.

1.4 Modifying License Units with the License Management Utility

V5.4

The following information is provided as a supplement to other License Management Facility (LMF) documentation. Before you use this information, you should read the *VMS License Management Utility Manual*, and you should understand the terms and conditions of your license agreement.

For VMS Version 5.4, many products require a PAK that includes license data to be registered in the LICENSE database. Some PAKs provide a MOD_UNITS option, which lets you modify the size of the registered licenses. If you have registered a license with the MOD_UNITS option, you can modify the size of the license to match the product to your VAX computer or VAXcluster environment. You can modify all licenses with the MOD_UNITS option, including those that specify the following:

- A size of 0 license units (unlimited availability)
- A predetermined size with a number of license units

To determine the options and size specified for your license, enter a command in the following format and press Return.

```
$ LICENSE LIST/FULL FORTRAN
```

License Management

1.4 Modifying License Units with the License Management Utility

The following display of a license list appears:

Use Ctrl/Z to exit, PF3-PF4 for Previous-Next Screen and Arrow keys to Scroll.

License Management Facility

License Database File: ART::SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB
Created on: 17-AUG-1989
Created by user: MONET
LMF Version: V1.0

Issuer: DEC
Authorization: USA-10
Product Name: FORTRAN
Producer: DEC
Units: 900
Version: 5.4
Date: (none)
Termination Date: 21-DEC-1991
Availability: F (Layered Products)
Activity: 0
Options: MOD_UNITS
Hardware ID:

Revision Level: 1
Status: Active
Command: REGISTER
Modified by user: DEGAS
Modified on: 21-AUG-1989 14:32:23.41

This display of a modifiable license includes MOD_UNITS next to the *Options* label. The size of the license is displayed next to the *Units* label. The license for this example provides the MOD_UNITS option and 900 license units.

You cannot activate licenses registered with fewer license units than a VAX computer requires. If your VAX computer or VAXcluster environment needs a different number of license units from the number registered with your license, find an appropriate license unit value for your VAX computer and change the size of your license as follows:

- 1 Enter the LICENSE LIST/FULL *product-name* command to display the license. Examine the display. Look for a code A through F, which is next to either the *Availability* label or the *Activity* label in the LICENSE LIST display. The codes, which designate license type, determine the appropriate license size for your VAX computer.
- 2 Find the name of your VAX computer in the first column of Table 1-1, the License Unit Requirement Table (LURT).

License Management

1.4 Modifying License Units with the License Management Utility

Table 1-1 License Unit Requirement Table (LURT)

System Marketing Model	License Types by Code					
	VMS				SIP	LP
	A	B	C	D	E	F
VAX 11/730	10	NA	NA	NA	230	50
VAX 11/750	12	NA	NA	NA	230	100
VAX 11/780,785	13	NA	NA	NA	230	100
VAX 6000-210,6000-310	58	NA	NA	NA	230	300
VAXft 3000	58	NA	NA	NA	230	300
VAX 6000-220, 6000-320,6000-410	69	NA	NA	NA	230	600
VAX 6000-230,6000-330	81	NA	NA	NA	400	900
VAX 6000-240, 6000-340,6000-350, 6000-420	93	NA	NA	NA	400	1200
VAX 8200,8250	20	NA	NA	NA	230	100
VAX 8300,8350	25	NA	NA	NA	230	200
VAX 8530	65	NA	NA	NA	230	400
VAX 8550,8700,8810	72	NA	NA	NA	400	600
VAX 8600,8650	28	NA	NA	NA	230	400
VAX 8800,8820	93	NA	NA	NA	400	1200
VAX 8830,6000-360, 6000-430	119	NA	NA	NA	600	1800
VAX 6000-440, 6000-450,6000-460	143	NA	NA	NA	600	2400
VAX 9000-210,9000-410	143	NA	NA	NA	600	2400
VAX 9000-220,9000-420	241	NA	NA	NA	800	4800
VAX 9000-430	330	NA	NA	NA	800	4800
VAX 9000-440	386	NA	NA	NA	800	4800
MicroVAX II	18	NA	100	NA	230	50
MicroVAX 2000,3100	18	NA	100	NA	230	20
MicroVAX 3500,3600,3800, 3900	60	NA	100	NA	230	200

Key to License Type Codes and Values

A—VMS capacity
 B—VMS server
 C—VMS concurrent user
 D—VMS workstations
 E—System integrated products
 F—Layered products
 NA—Not applicable

(continued on next page)

License Management

1.4 Modifying License Units with the License Management Utility

Table 1-1 (Cont.) License Unit Requirement Table (LURT)

System Marketing Model	License Types by Code					
	VMS				SIP	LP
	A	B	C	D	E	F
MicroVAX 3300,3400	60	NA	100	NA	230	100
VAX 4000-300	60	NA	100	NA	230	300
VAXstation II,II/GPX	NA	NA	NA	100	50	10
VAXstation 2000,2000/GPX	NA	NA	NA	100	50	10
VAXstation 3100,3200,3500, 3520,3540	NA	NA	NA	100	50	10
VAXserver 2000	NA	52	NA	NA	50	10
VAXserver 3300,3400,3500, 3600,3900,4000-300	NA	100	NA	NA	50	10
VAXserver 6000-210, 6000-310	NA	1443	NA	NA	230	200
VAXserver 6000-220, 6000-320,6000-410, 6000-420	NA	1737	NA	NA	230	400

Key to License Type Codes and Values

- A—VMS capacity
- B—VMS server
- C—VMS concurrent user
- D—VMS workstations
- E—System integrated products
- F—Layered products
- NA—Not applicable

- 3 Find the row with the appropriate name and the column with the code corresponding to the type of license you have. Note the value at the intersection of the row and column. For example, find the intersection of VAX 8800 with column F (the value shown is 1200). Unless NA (meaning a value is not applicable for this license type) appears, the value that you find is the number of units required for your VAX computer and type of license.
- 4 Modify your license by entering the value found in Table 1-1 as the parameter in a LICENSE MODIFY/UNITS=*number* command. For example, to modify a FORTRAN license running on a VAX 8800 computer, enter the following:

```
$ LICENSE MODIFY/UNITS=1200 FORTRAN
```

If you have entered the correct value for your license and VAX computer, the license should activate with the next LICENSE LOAD command. To activate the modifications immediately on a previously activated license, enter the following commands as well:

License Management

1.4 Modifying License Units with the License Management Utility

```
$ LICENSE UNLOAD FORTRAN
$ LICENSE LOAD FORTRAN
LMF-I-LOADED, DEC FORTRAN was successfully loaded with 1200 units
```

For details, see the *VMS License Management Utility Manual*.

Note: The license unit requirements provided by the License Unit Requirement Table are subject to change.

1.5 License Management Facility Notes

V5.4

The following list is offered to help new users with some common concerns and questions regarding the License Management Facility (LMF). For full explanations of these issues, see the *VMS License Management Utility Manual*.

- If you do not have a valid VMS license that is registered and activated, the system displays a warning message as part of system startup and restricts system use to the operator's console, OPA0.
- If a checksum error is displayed when you register a license, check *all* the fields of data that you entered.
- After your PAKs are registered, they are activated (loaded) automatically as part of each system startup.
- If a VMS availability license is registered with insufficient license units for the specified VAX computer, the system displays a warning message at system startup but allows normal system use.
- If a VMS activity license is registered with insufficient license units, the system displays the following message when the user (process) attempts to log in:

```
%LICENSE-F-EXCEEDED, licensed product has exceeded current license limits
```

Users can always log in to the operator's console, OPA0, however.

- The default LICENSE database is located in the file `SYS$COMMON:[SYSEXE]LMF$LICENSE.LDB`. You can move the database, although Digital does not recommend doing so. If you move the database, you must either define the logical name `LMF$LICENSE` at the system level to point to the new database, or use the `/DATABASE=filespec` qualifier with all LICENSE commands. To redirect LMF to another database location on a more permanent basis, add the following line to the command procedure `SYS$MANAGER:SYLOGICALS.COM`:

```
$ DEFINE/SYSTEM LMF$LICENSE device:[directory]LMF$LICENSE.LDB
```

If you specify a device other than `SYS$SYSDEVICE`, you must also mount the specified disk from the `SYLOGICALS.COM` command procedure.

- If you have a service contract with Digital for VMS Version 5.4, you might have VMS licenses registered in a separate LICENSE database located in the file `SYS$SPECIFIC:[SYSEXE]LMF$SYSTEM.LDB`. Each node of a VAXcluster environment might have a separate database containing only a VMS license. To access these VMS

License Management

1.5 License Management Facility Notes

licenses, you must use the /DATABASE qualifier with your LICENSE commands. You should not need to modify these licenses unless you change the SCS node name of the VAX computer. For information about these licenses and the LMF\$CONFIG command procedure that created them, see the *VMS License Management Utility Manual*.

- If you have multiple system disks in a VAXcluster environment, where all the systems can access one of the system disks, put your common LICENSE database on the readable disk. For any systems that boot from a separate system disk, you must redirect LMF to the LICENSE database. Define the logical name LMF\$LICENSE as the disk containing the database.

If you have multiple system disks in a VAXcluster environment, where some systems cannot access one of the system disks and where you must keep separate LICENSE databases, keep the databases identical. Whenever one database is modified, you must copy it to update the other databases.

- Each VMS license is restricted to a single node. You must assign a System Communications Services (SCS) name to the license when you register with the VMSLICENSE.COM command procedure, or you must enter a LICENSE MODIFY/INCLUDE=*node-name* command after you register the license. Although you can successfully activate an unassigned VMS license on a standalone system, you cannot activate one in a VAXcluster environment.

Note: The SCS node name is not necessarily the DECnet node name. SCSNODE is a System Generation Utility (SYSGEN) parameter.

1.6 VMS License Types

V5.4

The VMS operating system uses one of the following types of license depending on the hardware and software configuration used and currently supported:

- **VMS Availability License**

Provides unlimited access to the users on a VAX computer or VAXcluster environment and is sometimes referred to as a capacity license or clusterwide license. VMS availability licenses are sized according to License Unit Requirement Table entries for each VAX computer.

- **VMS Multiuser License**

Provides use according to a specified number of concurrent users. This is an activity-based license.

- **VMS Workstation License**

Provides use for a single user on a VAX workstation. Workstation licenses actually use the same licensing formulas as the VMS multiuser license. This is an activity-based license.

License Management

1.6 VMS License Types

- **VMS Server License**

Provides use for a single user on a VAXserver system. Server licenses actually use the same licensing formulas as the VMS multiuser license. This is an activity-based license.

1.7

License Activity Use Definition for VMS Licenses

V5.4

The *VMS License Management Utility Manual* describes a type of license called an activity license, which is based on the number of concurrent users. Every product has the option to define an activity as related to the License Management Facility. VMS defines activities, sometimes referred to as VMS users, as follows:

- Each remote terminal connection is considered an activity. This is true even if you set host to your local node (SET HOST 0).
- Each connection from a terminal server is considered an activity.
- A multiple-window session on a workstation is considered one activity, regardless of the number of windows.
- A batch job is not considered an activity.
- A remote network connection (other than a remote terminal connection) is not considered an activity.

VMS determines the number of activities through the LOGINOUT image as part of initiating a new interactive process.

1.8

VAXcluster License

V5.4

After you *install* VMS Version 5.4, you *must* register your VAXcluster license before you can use your VAXcluster software. If you are a service customer and you *upgrade* to VMS Version 5.4, you do not need to reregister this license.

During system startup, each VAXcluster node checks for a valid VAXcluster license. If you have not registered a VAXcluster license for a VAXcluster environment, each node displays the following message at system startup time:

```
&LICENSE-E-NOAUTH, DEC VAXCLUSTER use is not authorized on this node  
-LICENSE-F-NOLICENSE, no license is active for this software product
```

You cannot log in to any terminal other than the operator's console, OPA0. If you attempt to log in to a VAXcluster node before you register and activate a VAXcluster license, the system displays the following message:

```
DEC VAXCLUSTER license is not active
```

Note: Although interactive logins are disabled, your VAXcluster software performs normally, as configured by startup procedures and system parameters. Systems not running VAXcluster software are unaffected.

1.9 DECnet-VAX License

V5.4

After you *install* VMS Version 5.4, you *must* register your DECnet-VAX license before you can use DECnet-VAX software. If you are a service customer and you *upgrade* to VMS Version 5.4, you do not need to reregister this license.

There are two DECnet-VAX licenses:

- The end node license named DVNETEND
- The routing node license named DVNETRTG

All routing nodes must have a routing license. Each end node can have either an end node license or a routing license. If neither license is registered and activated, DECnet does not start and your use is limited to local DECnet only (SET HOST 0). If DECnet-VAX is running when you register your license, you must stop and then restart DECnet-VAX.

You can control which VAXcluster nodes have access to each kind of license. Using the LICENSE MODIFY/INCLUDE=(*node-name[,node-name,...]*) command, you can assign licenses to nodes and limit access as needed. For example, you can assign a routing node license to only one VAXcluster node and assign the end node licenses to the remaining VAXcluster nodes. If you choose this approach, make sure you assign each end node license to the same list of nodes. That is, specify identical *include lists* for each license of the same type. For details, see the *VMS License Management Utility Manual*.

1.10 VAX RMS Journaling License

V5.4

After you *install* VMS Version 5.4, you *must* register your VAX RMS Journaling license before you can use VAX RMS Journaling. If you are a service customer and you *upgrade* to VMS Version 5.4, you do not need to reregister this license.

On systems that do not have registered and activated journaling licenses, users cannot write to any files marked for journaling.

1.11 VMS Volume Shadowing License

V5.4

After you *install* VMS Version 5.4, you *must* register your VMS Volume Shadowing license before you can use VMS Volume Shadowing. If you are a Service Customer and you *upgrade* to VMS Version 5.4, you do not need to reregister this license.

Please note that, with VMS Version 5.4, the name of the product has been changed from VAX Volume Shadowing to VMS Volume Shadowing. VAX Volume Shadowing is now referred to as VMS Volume Shadowing Phase I. VMS Version 5.4 Volume Shadowing is referred to as VMS Volume Shadowing Phase II. The same VMS Volume Shadowing license applies for both Phase I and Phase II.

License Management

1.11 VMS Volume Shadowing License

To use VMS Volume Shadowing Phase I, all computers connected to the CI require the VMS Volume Shadowing license. If the license is not installed, a MOUNT/SHADOW command generates the following error message:

```
%MOUNT-F-SHADOWFAIL, failed to create (or add to) the shadow set  
-LICENSE-F-NOLICENSE, no license is active for this software product
```

To use VMS Volume Shadowing Phase II, all computers in the VAXcluster, including satellites, must have a VMS Volume Shadowing license. If you are a Service Customer and you upgrade to VMS Version 5.4, you do not need to reinstall your VMS Volume Shadowing license, but you must license any satellite that is not currently licensed for VMS Volume Shadowing.

2 General User Release Notes

This chapter discusses information about the VMS Version 5.4 operating system that is of interest to the general user.

For information about the new features included in VMS Version 5.4, see the *VMS Version 5.4 New Features Manual*.

2.1 \$ENTRY — New Symbol

V5.2

The successful execution of a PRINT or SUBMIT command creates or updates the global symbol \$ENTRY. The value of this symbol is a string containing the entry number of the print or batch job just entered.

To retain a job's entry number for later reference, immediately store its value in another symbol to avoid losing the information when \$ENTRY is updated to reflect the most recent job queued. The following example illustrates the use of \$ENTRY:

```
$ SUBMIT TEST.COM
Job TEST (queued CLUSTER_BATCH, entry 493) pending
$ BATCH_JOB = $ENTRY
.
.
.
$ DELETE/ENTRY=' BATCH_JOB'
```

2.2 Batch/Print Facility — Performance Improvements

V5.3

VMS Version 5.3 contains several performance-related improvements for the Batch/Print facility. These changes are designed to improve response time of selected DCL commands and to increase overall throughput of the Batch/Print subsystem during periods of peak utilization. However, the amount of performance gain experienced by a particular site varies greatly depending on the size and characteristics of the work load placed on the job controllers. Users on systems where the queuing system use is light will probably not perceive any improvement, whereas users on systems that generate considerable batch and print activity may notice small-to-moderate gains in service responsiveness.

Sites that should benefit most from these performance improvements are ones that heavily utilize the queuing system and that use the system in the following ways:

- Issue many single-file SUBMIT and PRINT command requests.
- Use the SPAWN command to create subprocesses that require job controller action on termination.

General User Release Notes

2.2 Batch/Print Facility — Performance Improvements

- Generate a high number of interactive or detached process terminations, or both.
- Use the /NOTIFY qualifier to SUBMIT and PRINT commands (or through \$SNDJBC system service calls).

For these functions processed by the job controller, performance is enhanced by reducing I/O operations to the queue file. Also, efficiencies are achieved by decreasing the number of \$SNDJBC system service calls used to enter a simple job and by eliminating unnecessary communication between job controllers to perform job notification.

V5.4

VMS Version 5.4 contains additional performance improvements for the Batch/Print facility. The same goals and cautions expressed previously for the Version 5.3 enhancements also apply to the Version 5.4 enhancements.

Sites that should benefit most from these performance improvements are ones that heavily utilize the queuing system and that use the system in the following ways:

- Enable image accounting or generate considerable accounting file activity because of process terminations or batch job completions, or both
- Use the standard VMS Print Symbiont or the LAT Symbiont to handle print execution queues
- Enter timed release jobs by using the /AFTER qualifier to PRINT and SUBMIT commands (or through \$SNDJBC system service calls)
- Delete files automatically on job completion by using the /DELETE qualifier to PRINT and SUBMIT commands (or through \$SNDJBC system service calls)

For these functions, sites can enhance performance by optimizing I/O operations to the accounting file, eliminating redundant symbionts to job controller communication, and improving internal job controller algorithms. In addition, by changing to the job controller's algorithm for processing queue file free-list records, sites can reduce I/O operations to the queue file during the execution of various functions.

2.3

Command Procedures — Restriction

V5.0

The VMS operating system now requires that all commands, full-line comments, and labels in command procedures be preceded by the dollar sign (\$) character. Although users have always been instructed to place a dollar sign before commands and labels, command procedures that omitted dollar signs before labels did not necessarily stop executing.

With VMS Version 5.0, labels without dollar signs are treated as data lines. Any reference to a label without a dollar sign will not execute as expected.

2.4 DECwindows Notes

The release notes in this section pertain to DECwindows software.

2.4.1 DECterm Terminal Emulator

The release notes in this section pertain to the DECterm terminal emulator.

-
- 2.4.1.1 Conformance Level Change for Support of Terminal State Reports**
- V5.4** In previous versions of VMS, DECterm had to operate at conformance level 3 (VT300 mode) to support terminal state reports. This conformance level requirement caused a problem because the DCL command SET TERMINAL/INQUIRE set the conformance level to 2 (VT200-series mode).
- With VMS Version 5.4, DECterm requires only a level 2 conformance level in order to support terminal state reports.

-
- 2.4.1.2 Graphics**
- V5.1-1** The following information is specific to DECterm graphics:
- In some cases, DECterm creates a private color map in which the color map changes for the entire workstation when the DECterm window has the input focus. DECterm creates this private color map when ReGIS or sixel graphics are displayed in the window and DECterm is unable to allocate a sufficient number of colors (by default, 4 colors on a 4-plane system or monochrome system and 16 colors on color systems with more than 4 planes) from the default color map. To restore a DECterm window to the default color map, first clear the window by clicking on Clear Display in the Commands menu, then reset the terminal by clicking on Reset Terminal in the Commands menu.
 - Any dialog boxes that are created while DECterm is using a private color map are displayed black. To avoid this problem, create the dialog boxes (for example, bring them up for the first time) when DECterm is using the default color map.
 - Only graphics, not text, are written to the graphics backing store. When part of a window has to be redrawn, DECterm draws the graphics portion of the window first, then overlays it with text. Therefore, the window might not look the same when redrawn as it did when the picture was first drawn.
 - Because ReGIS addresses the entire window, not just 24 rows and 80 columns, the relationship between text and graphics may not always be the same as on the VT330 or VT340 terminal.
 - The following ReGIS features are not implemented: command display mode, scrolling, hard copy, and output cursors. The workstation mouse is used as the input cursor, but its shape cannot be changed with the S(C(I)) command.

General User Release Notes

2.4 DECwindows Notes

V5.4

- In previous VMS versions, the DECterm gray levels used for the default color map on monochrome systems were too dark. This darkness problem occurred because the values were chosen to be the same as on the VT340, which has different video hardware.

Beginning with VMS Version 5.4, DECterm uses a monochrome color map that uses higher intensities. Table 2-1 lists the previous and new gray levels according to color index.

Table 2-1 DECterm Gray Levels

Color Index	Previous Gray Level	New Gray Level
0	0	0
1	13	33
2	26	66
3	40	100
4	6	25
5	20	50
6	33	75
7	46	85
8	0	0
9	13	33
10	26	66
11	40	100
12	6	25
13	20	50
14	33	75
15	46	85

As in previous VMS versions, when DECterm is emulating 4-bit planes, colors 0 and 7 are replaced with the text foreground and background colors (normally set through the Session Manager). Color 0 is set to the darker of the two colors.

2.4.1.3 Initializing DECterm

V5.3

The following information is specific to DECterm initialization:

- To avoid having your DECterm windows shrink to the default size of 80 characters by 24 lines unexpectedly, Digital suggests that system-wide and user login command procedures (SYLOGIN.COM and LOGIN.COM) not execute the following:
 - The SET TERMINAL/INQUIRE command on DECterm windows
 - The SET TERMINAL/INQUIRE command on mailbox devices created from the Session Manager because this prevents the Session Manager from starting applications such as DECterm

General User Release Notes

2.4 DECwindows Notes

Use of the SET TERMINAL/INQUIRE command is unnecessary because the DECterm controller provides VMS with the proper characteristics and size of DECterm windows.

To make login procedures work correctly both on DECterm windows and in other environments, such as on terminals, use the following commands in a systemwide or user login command procedure:

```
$ !
$ ! SYS$MANAGER:SYLOGIN.COM and users' LOGIN.COM may contain the
$ ! following command line:
$ ! $ IF (F$MODE() .EQS. "INTERACTIVE") THEN SET TERMINAL/INQUIRE
$ ! To avoid resizing of a terminal window on a workstation, you may
$ ! substitute the following command sequence:
$ !
$ IF f$getdvi( "sys$output:", "trm" )
$ THEN
$     devnam = f$getdvi( "sys$output:", "devnam" ) - "_" - "_"
$     devnam = f$extract(0, 2, devnam)
$     if devnam .eqs. "WT" then goto skip_inquire
$     if devnam .eqs. "TW" then goto skip_inquire
$     if devnam .eqs. "FT" then goto skip_inquire
$     if devnam .eqs. "RT" then goto skip_inquire
$     set terminal sys$output:/inquire
$ skip_inquire:
$ ENDF
```

This routine bypasses the SET TERMINAL/INQUIRE command on DECterm, SET HOST, and VWS, and on nonterminal devices such as the mailboxes created by the Session Manager.

- If you attempt to resize a DECterm window before you see a prompt in the window, the window may disappear.

2.4.1.4 Keyboards and Languages

The following information is specific to DECterm keyboards and languages:

V5.1-1

- To create a compose character in DECwindows, you must press the COMPOSE and space key at the same time, not the COMPOSE key alone as on character-cell keyboards. The COMPOSE key is a modifier and is used like a SHIFT key.
- You should make changes to keyclick, auto-repeat, keyboard dialects (national keyboards), keyboard usage mode (data processing/typewriter mode) and caps lock/shift lock in the Session Manager across all workstations, rather than through DECterm.
- You select National Replacement Character Sets (NRCSs) in the Customize/7-bit NRCS Selection menu in DECterm. Their selection is independent of both the keyboard dialect and the keyboard usage mode. For example, to use the French NRCS with the French keyboard, you must change both the Session Manager and DECterm.
- When the keyboard becomes locked (for example, because the input silo is full), DECterm rings the bell for each character that comes in until the lock condition is cleared, rather than disabling keyclick as on the VT300-series terminals.

General User Release Notes

2.4 DECwindows Notes

- V5.2
 - In some situations, DECterm can continue to auto-repeat a key even after the key has been released; this is especially a problem with WPS-PLUS under ALL-IN-1. To avoid this problem, activate the Lock User Features toggle button in the Customize General dialog box.
- V5.4
 - The Dutch NRCS is no longer supported; selecting Dutch NRCS is the same as selecting North American (ASCII).

2.4.1.5 Memory — How to Reduce DECterm Use

- V5.4

In earlier versions of VMS, when the user disabled the Record Lines Off Top setting in the Customize Display dialog box, DECterm allocated memory for recorded lines but did not use this recorded memory. For VMS Version 5.4, DECterm does not allocate this memory, so you can reduce the memory that DECterm uses by disabling this setting for those windows in which you do not need the recorded lines.

2.4.1.6 Memory — Logout Problem

- V5.3

When you log out of a DECterm window, about 64 pages (32 kilobytes) of virtual memory are not freed. Thereafter, if you create and delete many DECterm windows, the DECterm controller process might run out of virtual memory. If the controller runs out of virtual memory, it fails and you lose all of your DECterm windows.

To prevent this problem, you should periodically exit from all your DECterm windows so that a new controller is created the next time you create a DECterm window. For example, choose Quit from the Session Manager's Session menu instead of choosing Pause.

2.4.1.7 Memory — Resize Request Problem

- V5.4

If DECterm is unable to allocate enough memory to resize its display list (when the number of rows or columns or the number of lines saved off the top changes), it reverts to the previous size of the display list. DECterm will not display an error message but will ignore the resize request.

2.4.1.8 PC Interoperability Restrictions

- V5.3

The following interoperability restrictions apply when you use DECterm on a personal computer (PC):

 - The DECterm window is not sized properly on the PC screen. Initially, the DECterm window is located partially off the screen, and you must center it manually. You must also resize it manually because it is too big for the PC screen.

To make the window fit on the screen, use the condensed font in a window that is 80 columns by 24 rows. To do this, choose Window... from the Customize menu, click on the Condensed font (132 columns) button, then click on OK. To save the change, choose Save Current Settings from the Customize menu. DECterm automatically moves the window so that it is on the screen.

 - When the Backspace toggle button is enabled, pressing the backspace key produces no response on the PC. To work around this problem, choose Keyboard... from the Customize menu and click on the Delete option button.

General User Release Notes

2.4 DECwindows Notes

- The Comma Key Sends ,< option does not work on the PC. Pressing the comma key produces a comma, but pressing Shift/, produces the number 9. To work around this problem, choose Keyboard... from the Customize menu and click on the Comma Key Sends ,, option button.
- The Tilde Key Sends ESC option in the Customize Keyboard dialog box does not work on the PC server. When this option is enabled, pressing the tilde (~) key gives no response; it should act as an escape key (and does on the native platform). To work around this problem, click on the Tilde Key Sends '~ option button.
- Because of a problem with PC-based and Macintosh-based X servers, DECterm supports only the following keys on LK201 keyboards: function, editing, and keypad. These servers do not generate all the KEYSYMs the LK201 supports; therefore, not all DECterm functions are accessible. For example, on a keyboard that is not an LK201, you cannot run EDT in screen mode from DECterm.

Users with keyboards that do not include some of the LK201 keys will not easily be able to run programs that depend on those keys. However, functional equivalents for the keys are often available. For example, EDT has a line-editing mode in which you enter commands in alphanumeric form rather than through function keys. If a program requires function keys, you can manually enter the escape sequences for the keys. For example, the keypad 7 key is the 3-character sequence <ESC>Ow, in which <ESC> is an escape, which you can enter as Ctrl/3 or Ctrl/[. The escape sequences sent for each key are listed in the *DECterm Text Programming Manual*.

2.4.1.9 Text

V5.1-1

The following information is specific to DECterm text:

- User-loadable characters (DRCS), print screen, local mode, and control representation mode are not implemented.
- The checkerboard character (character 97 in the DEC Special Graphics character set) is used instead of the reverse question mark as an error character.

2.4.1.10 Window Scrolling Problem

V5.4

Sometimes a DECterm window freezes during scrolling, especially if the window has been left logged in over an extended period of time, such as overnight. You can recover from this condition by choosing the Clear Communications command from the Commands menu.

2.4.2 DECW\$COLOR Guidelines for Changes in Achromatic Color (GrayScale) Rendering

V5.4

To improve image quality on a monochrome monitor, the procedure for achromatic color (GrayScale) rendering has been changed for VMS Version 5.4.

General User Release Notes

2.4 DECwindows Notes

In previous versions of VMS, achromatic color was rendered by converting the red, green, and blue values to a single intensity and then storing this value in all three components. With VMS Version 5.4, the server performs the same red, green, and blue intensity conversion but stores the resulting value in the green colormap entry only; null values are stored in the red and blue entries. This procedure produces a clearer image on a monochrome monitor, but in the process it changes the rendition on a color monitor from shades of gray to shades of green.

These rendering changes apply to achromatic color only. You indicate achromatic color by setting DECW\$COLOR to false in the private server setup file for a particular system. If you set DECW\$COLOR to true or if there is no private server setup file, the color rendering procedure is the same as with previous versions of VMS.

If you require the previous achromatic color rendition, you should define the systemwide logical name DECW\$COLOR_USE_V2_GRAYSCALE_SEMANTICS to be true. Conversely, if you do not define DECW\$COLOR_USE_V2_GRAYSCALE_SEMANTICS, or if you define it to be false, achromatic color is rendered using the green component only.

2.4.3 DECwindows Startup Problem

V5.4 You must wait until DECnet is running before you create remote DECterm windows. If you start DECwindows before starting DECnet, and then start DECnet, you will not be able to create remote DECterm windows.

2.4.4 Desktop Applications

The following sections describe information pertaining to DECwindows Desktop Applications.

2.4.4.1 Calendar — Corrected Problems

V5.4 The following Calendar problems have been corrected in VMS Version 5.4:

- You can now see the Auto Click on Clock toggle button in the Day View dialog box (available by choosing Day View... from the Customize menu on 100-dpi monitors).
- The missing menu label Past Days has been added to the General dialog box (available by choosing General... from the Customize menu).

2.4.4.2 Calendar — Restrictions and Limitations

V5.3 With VMS Version 5.3, DECwindows Calendar uses a new format for the Calendar database file. The first time the Calendar is invoked, it prompts you to confirm that you want to upgrade your existing Calendar datafile to the new format. Once converted, your database can no longer be accessed by previous versions of DECwindows Calendar.

General User Release Notes

2.4 DECwindows Notes

If your VAXcluster contains workstations running mixed versions of VMS (for example, both VMS Version 5.3 and Version 5.2-1), and you want to use Calendar from any workstation, you must choose when you want to convert the Calendar database. The conversion can be done at your convenience, but you must run Calendar remotely if the version on your workstation does not match your Calendar database version. Instructions for running DECwindows applications remotely can be found in the *VMS DECwindows User's Guide*.

V5.4 If the only entries on a particular day are repeat entries, Calendar loses the first alarm of a repeat entry on that particular day. If your first alarm for an entry is normally 5 minutes before the entry, add an additional alarm before the entry to ensure that the 5-minute alarm goes off.

2.4.4.3 Cardfiler — Scroll Bar

V5.4 By default, Cardfiler now places the scroll bar for the list box on the right.

2.4.4.4 Clock — Changing the System Time

V5.1 Clock does not notice if you change the system time backward while the clock is running. If you change the system time, you must exit from Clock and then rerun Clock so that the change takes effect.

2.4.4.5 CDA Viewer — Corrections and New Functionality

V5.4 The following corrections have been made to the Compound Document Architecture (CDA) Viewer for VMS Version 5.4:

- In previous versions, the origin of the image was at the origin of the frame. This problem has been corrected; the origin of the image is now at the origin of the bounding box.
- In previous versions, text sometimes was placed in the wrong galley, and a document ended abruptly after a small galley (or text block), such as a caption. After a nesting galley was encountered, the remainder of the document appeared to be blank or the remaining text appeared in the wrong galley.

This problem has been corrected; support has been added for nested galleys.

- In previous versions, the behavior of the new-page directive was unpredictable and occasionally produced access violations. For example, an access violation occurred when you attempted to go to the end of the document.

This problem has been corrected; it is now possible to generate a blank page.

2.4.4.6 CDA Viewer — Problems and Unexpected Behavior

V5.4 The following problems and unexpected behavior exist:

General User Release Notes

2.4 DECwindows Notes

Problems

- If you cancel the PostScript viewing process while skipping over several pages, the page number indicator shows which page was being interpreted when the cancellation happened, but the screen may not show the results of that page. This situation occurs because when skipping pages, the interpreter's drawing is deactivated; it does not actually write the skipped pages anywhere.

You skip pages if both the following conditions occur:

- The document is unstructured or you disable the Use Comments option
- You ask to view a page behind the current page or more than one page in front of the current page, or if the current page is being reinterpreted

Unexpected Behavior

- When you first use Display PostScript text with the CDA Viewer or with any other application using Display PostScript, the display takes much longer than subsequent uses of the same fonts and sizes. This occurs because Display PostScript must first generate the characters from outline definitions and then use a cached version of the character. Subsequent releases of VMS will attempt to speed up the initial display of some sizes and orientations of characters.
- When you first use Display PostScript, the workstation display freezes for several seconds while the Display PostScript extension loads and initializes. This condition does not happen again until you restart (not reset) the server.
- When you tell the CDA Viewer to process several pages in a row, it attempts to optimize by interpreting all the pages but not displaying some of the nonfinal pages. This can cause periods of apparent inactivity while the nonfinal pages are interpreted.
- When displaying PostScript files, the CDA Viewer attempts to optimize the display of documents by using the Document Structuring Conventions specified by Adobe Systems Incorporated. The CDA Viewer determines whether a document conforms to these conventions by looking for the structuring comment *!/PS-Adobe-* at the beginning of the document. If a document has this comment as its first line but does not obey the structuring conventions (for example, if pages are not independent), the results are unpredictable. Most likely, you will see PostScript error messages displayed.

PostScript output from Version 1.2 of VAX DOCUMENT obeys the conventions specified by Adobe Systems Incorporated. The following products are known to produce incorrect structuring comments or incorrectly follow the structuring conventions:

- VAX DOCUMENT Version 1.1
- DECpage Version 3.1
- DECwrite Version 1.0

General User Release Notes

2.4 DECwindows Notes

If you view a PostScript file with no structure statements or when you disable the Use Comments option, the CDA Viewer operates correctly. However, if you depart from reading pages in a strictly sequential order, you encounter long delays while the PostScript code, starting at the beginning of the document, is reinterpreted up to the page you asked for.

To view PostScript files with incorrect structure statements from the File Selection menu, disable the Use Comments option in the Page Size dialog box.

2.4.4.7 CDA Viewer — Restrictions

V5.4

When you view PostScript files using Watch Progress mode with the CDA Viewer, error messages are regenerated upon exposes, resulting in error message boxes frequently popping up after you acknowledge the error message.

You can work around the error message problem by not using Watch Progress mode or by resizing and moving the Viewer:

- 1 When an error message is posted, resize and move the Viewer so that the error message window is no longer obscuring the Viewer window.
- 2 Acknowledge the message, and it will not be regenerated.

2.4.4.8 DECwindows Mail — Corrected Problems

The following DECwindows Mail problems have been corrected:

V5.2

- Increasing the vertical size of DECwindows Mail's main window in the pane interface resulted in commands operating on the wrong messages and, in some cases, caused access violations. This problem has been corrected; access violations no longer occur in this context.
- In some instances, DECwindows Mail aborted with an access violation if you did not select a message during the extract, print, reply, and forward operations. This problem has been corrected; a message box now asks the user to select a message in this case.
- Deselecting (with Shift/MB1) the last of a group of selected messages resulted in a subsequent access violation. This problem has been corrected; access violations no longer occur in this context.
- If an access control string was used in a distribution list file specification, the password was not replaced and all recipients saw the user name and password used to access the distribution list. This problem has been corrected; the word password is substituted for the real password.

V5.3

- Delete Drawers... did not delete the actual mail file. This problem has been corrected.
- When you used a dialog box to move or copy messages selected in the Main window, making a selection caused the Main window selection to be lost. This problem has been corrected; the Main window selection is no longer lost.

General User Release Notes

2.4 DECwindows Notes

- Using Pick From Selected Folder in the paned interface on the Deliver folder prevented any future delivering or reading of new mail. This problem has been corrected; future delivering or reading of mail is no longer affected.
 - Entering an illegal personal name in the Create-Send window no longer makes that window unusable, as was the case in previous versions.
 - The icon color for both small and large icons now changes when new mail arrives.
 - The Pane widget (used in the Read window and in the paned Main window) now complies with the *XUI Style Guide*. The knob that was previously grabbed to adjust the panes is gone, and the panes are now separated by a double line. To adjust a pane, move the cursor over the double line (it should change to the double-arrow pane cursor), press and hold MB1, and drag the mouse to the desired position.
- V5.4**
- In previous versions, the Options button in the Print dialog box caused an intermittent access violation. This problem has been corrected; access violations no longer occur in this context.
 - In VMS Version 5.3, reading large messages containing many large records sometimes caused DECwindows Mail to fail. This problem has been corrected; reading large messages no longer causes failures.
 - In VMS Version 5.3, DECwindows Mail failed when you attempted to read or send a DDIF message if the CDA Viewer could not be activated. This problem has been corrected; failures no longer occur in this context.
 - In VMS Version 5.3, moving messages by using the dialog box to a drawer with a display name that was not a valid file name sometimes caused DECwindows Mail to fail. This problem has been corrected; failures no longer occur in this context.
 - In VMS Version 5.3, DECwindows Mail failed if you attempted a paste operation when the clipboard was empty. This problem has been corrected; failures no longer occur in this context.

2.4.4.9 DECwindows Mail — Problems and Restrictions

DECwindows Mail has the following problems and restrictions:

- V5.1**
- SYS\$SCRATCH defines the directory where any temporary files are created. If DECwindows Mail is run in a detached process, SYS\$SCRATCH must be explicitly defined in that process.
 - When you read a mail message while the Auto Refile option is set, the message is immediately moved to the MAIL folder. If you delete or move the message, the action does not take place until the INBOX folder is closed. Therefore, if you read a message, delete it, and open the MAIL folder without first closing the INBOX or NEWMAIL folder, you see the deleted message listed in the MAIL folder. When you eventually close the INBOX or NEWMAIL folder, the deleted message is deleted from the MAIL folder.

General User Release Notes

2.4 DECwindows Notes

- Printing is performed through the standard DECwindows print function, which does not automatically use print defaults specified in your VMS Mail Utility user profile. Any modifications made to the print options using either the Print... menu item from the File pull-down menu or the Modify Print Defaults... menu item from the Customize pull-down menu are in memory as long as DECwindows Mail is active but are not stored from one DECwindows Mail session to another.
- System managers should be sure to set MAIL\$SYSTEM_FLAGS at system startup time as described in the *VMS Mail Utility Manual*.
- DECwindows Mail enables purging of deleted mail by default. This is true even if you have set NOAUTO_PURGE through the VMS Mail Utility.
- V5.3** • Input focus is not always assigned to the Create-Send window when it is mapped.
- On the In Window submenu of the main window Read menu, menu entries of the form "Window #" will be enabled improperly even if no messages are selected. Choosing one of these entries with no messages selected may cause an access violation.
- V5.4** • There is no way to avoid reading PostScript files through the CDA Viewer or to force files that Mail considers non-PostScript files to be displayed using the CDA Viewer. The only way to work around this problem is to forward the files to yourself and either add or remove %! from the beginning of the file.
- From the Mail Utility, there is no way to tell the CDA Viewer to change options like Use Comments or to orient or scale the image. The only way to work around this restriction is to extract the PostScript file, and view it with the CDA Viewer.

2.4.4.10 DECwindows Mail — Problem Sending PostScript Files

V5.4 If you send a PostScript file using the default editor, the text in the PostScript file may wrap. This wrapped condition can generate invalid PostScript, and the message will not be printed or viewed correctly on the receiving end. To prevent wrapping, you can increase the width of the Create-Send window sufficiently or send the file through one of the following methods:

- From the Main or Read window's Create-Send menu, choose the File (no editor)... command.
- From the Create-Send window's File menu, choose the Include File (no editor)... command.

General User Release Notes

2.4 DECwindows Notes

-
- 2.4.4.11 Paint — Corrected Problems**
- V5.4 The following Paint problems have been corrected:
- In previous versions, attempting to use the pencil or paintbucket tool on the right or bottom border of the Paint window caused Paint to fail. This problem has been corrected; failures no longer occur in this context.
 - In previous versions, if you resized the Paint window to be smaller than the zoom box, the zoom box did not shrink accordingly. If the picture was larger than the screen, a failure sometimes occurred if the zoom window extended past the edge of the pixmap and you attempted to draw there. This problem has been corrected; the zoom box now shrinks accordingly.
-
- 2.4.4.12 Print Screen — LJ250 Problem Corrected**
- V5.3-2 In VMS Versions 5.3 and 5.3-1, a problem existed with the Session Manager's Print Screen function on an LJ250 printer. The screen image was not properly captured when color output in sixel format was requested. VMS Version 5.3-2 corrects this problem.
-
- 2.4.4.13 Print Screen — Restrictions**
- V5.3 The following restrictions apply to Print Screen:
- Sixel output is larger than it appears on a 100-dpi screen.
 - When the 2:1 aspect ratio is chosen for sixels and the picture is rotated, it is scaled incorrectly. To work around this problem, choose the No Rotate option.
 - On small-memory systems, print screens of large areas may cause memory overflow errors.
-
- 2.4.4.14 Print Screen — VAXstation 3520 and 3540 Correction**
- V5.4 In previous versions of VMS, you could not print the screen (through the Print Screen menu) on a 24-plane system such as the VAXstation 3520 and 3540.
- This problem has been corrected; you can now capture screen output on these 24-plane systems. However, please note that the outline of the Print Screen box is not printed.
-
- 2.4.4.15 Print Screen — VAXstation 3520 and 3540 Restriction**
- V5.3 Print Screen on VAXstation 3520 and 3540 systems works only if one colormap is installed on the system. If any software that installs a second colormap is running, Print Screen returns an error.

2.4.5 FileView

FileView gives you access to DECwindows applications and provides commands for you to work with files. See the *VMS DECwindows User's Guide* for more information about FileView.

The release notes in this section pertain to FileView.

2.4.5.1 Changes to FileView

V5.3

The following FileView changes have been made:

- FileView's user interface definition has been converted to User Interface Language (UIL). Customization of FileView through XDefaults (DAT) files remains unsupported. (Previously, FileView used an XDefaults resource file for its user interface.) The name of FileView's application class was changed from VUE\$DEFAULTS.DAT to VUE\$MASTER.DAT. These changes may affect users who are doing advanced workstation customization.
- A new algorithm is used to process the Directory field in the main window. Other than improved error messages, the most noticeable change affects use of an unconcealed logical name defined as a search list of directories. DECW\$SYSTEM_DEFAULTS is an example of such a logical name. If you try to set your default directory to a search list of directories, the first directory in the search list is used and the other search list entries are ignored. In previous versions, this behavior was undefined. Use of logical names in the File Filter field remains fully supported.
- To support DECwindows displays on PC monitors, FileView's windows were changed, where necessary, to fit on screens as small as 512 pixels by 342 pixels.
- FileView now positions dialog boxes and menus (both pop-up and pull-down) to be completely visible when they first appear. (Previously, windows were sometimes clipped by the edges of the screen.) This change makes it more practical to use FileView near the bottom of the screen. Menu positions are adjusted without warping the pointer position. Menu visibility takes precedence over pop-up menu history.
- FileView was changed to make it easier to access files on a DECnet node other than the node on which FileView itself is running. (You must include a node name in the File Filter field.) Tasks are now passed the node name of remote files. Previously, the node name was passed to a task only if the node name was visible in the displayed file list.
- FileView now uses a single help widget for all help requests. Previously, if you requested help while a help window was already visible, a second help window appeared. Now, if the help window is displayed when you request help (either from the menu bar or by using context sensitive help), the subject changes in the current help window.
- The FileView code that processes task callback messages is now more tolerant about varying task message formats. FileView code has also been rewritten to report more meaningful error messages and to give better support for optional parameters on some messages.

General User Release Notes

2.4 DECwindows Notes

Saving a View

The initial settings in the Save View dialog box have been changed. When you choose Save View... from the Customize menu, the dialog box pops up with all its toggle buttons set and with Startup (the name of the startup view) contained in the Name of View field.

For your convenience, All and None buttons have been added to set and clear all the toggle buttons at once. Now, by default, Save View... behaves just like Save Startup View.

Previously, only the File Filter and Directory toggle buttons were set by default. The Name field was empty.

Customizing FileView

You can enter labels for the window and icon names, and you can indicate whether the DECnet node name and the current default directory should also be displayed. The window and icon names can be different from each other, and they can be stored as part of a saved view. By default, the DECnet node name and the default directory are included in both the window and icon names.

To start FileView as an icon instead of as a window, click on the Icon option button under the Initial State label. Then, choose Save Startup View from the Customize menu to save the change. The initial state setting can also be saved as part of a named view, but it only takes effect when you press and hold the Shift key and choose the saved view from the Views menu. (Rather than changing the current view, pressing and holding the Shift key creates a new view from the saved view's settings.)

Now that you can remove any or all of FileView's built-in verbs from the menu bar, and can save the menu bar (or lack thereof) as part of your startup view, you can create a constrained FileView environment that does not allow further customization. To restore FileView's default menu bar, follow these steps:

- 1 Open a window that contains the DCL prompt (\$).
- 2 If necessary, direct your display to your workstation with the DCL command SET DISPLAY.
- 3 Enter the DCL command DEFINE/JOB VUE\$PROFILE NL:.
- 4 Enter the DCL command RUN SYS\$SYSTEM:VUE\$MASTER.
A system-default FileView window will be displayed on your workstation.
- 5 Choose Logical Names... from the Utilities menu.
- 6 In the Logical Names dialog box, deassign VUE\$PROFILE.
- 7 Choose Save Startup View from the Customize menu.
- 8 Choose Quit from the Control menu.

The next time you run FileView, it will have the full (system default) menu bar.

2.4.5.2 Customizing FileView to View PostScript Files

V5.4

To view a PostScript (PS) file through FileView, the CDA Viewer must be told that the file is a PostScript file. Otherwise, the CDA Viewer claims that the file contains an unsupported format.

You can customize FileView to recognize PostScript files by adding a new verb that points to the following command file:

```
$!  
$! Command procedure to start the CDA Viewer from  
$! DECwindows FileView and to use the format qualifier if  
$! the file has a PS extension.  
$!  
$ vue$popup_progress_box 8  
$ vue$suppress_output_popup  
$ vue$get_next_selection  
$ vue$read_selection  
$ format = ""  
$ if (".PS" .eqs. f$parse(selection,,, "TYPE")) then format = "/format=ps"  
$ view/select=x_display 'selection' format  
$ exit
```

Through this command file, the new verb can view CDA files as well as PostScript files.

Note that you can also customize FileView such that a double-click on a file with the PS extension automatically runs the CDA Viewer.

2.4.5.3 FileView Restrictions

V5.1

FileView has the following restrictions:

- If you start FileView from the Session Manager window, you cannot have DCL INQUIRE statements in your LOGIN.COM or SYLOGIN.COM file. FileView executes both your LOGIN.COM and SYLOGIN.COM command procedures and is an interactive mode process. It cannot, however, handle input or output from your command files. A "No condition handler found" error message appears in the Session Manager control panel if FileView fails because of an INQUIRE statement in one of the login command procedures.
- FileView runs your tasks as subprocesses; therefore, your process quotas that are depleted by subprocess creation dictate how many FileView tasks you can run simultaneously.

Before creating a new process, FileView checks these quotas and displays a warning in a dialog box if any are too low. The quota name is included in the message and can be one of the following:

- ASTLM
- BIOLM
- BYTLM
- FILLM
- PGFLQUOTA
- PRCLM
- TQELM

General User Release Notes

2.4 DECwindows Notes

The most likely quotas to be consumed are your process limit (PRCLM) and buffered I/O byte count (BYTLM). To run a single task from FileView, your BYTLM quota should be a minimum of 10000. Add an additional 5000 for each task you want to be able to run simultaneously. So, to be able to run five simultaneous tasks, your PRCLM quota must be at least 5, and your BYTLM quota must be at least 30000. Process creation can reduce remaining ASTCNT and BIOCNT by 3, and FILCNT by 2. PGFLQUOTA usage is highly dependent on the task. If required, the system manager can adjust these user quotas by using the Authorize Utility.

FileView checks these quotas when creating its subprocesses. However, some quotas such as PGFLQUOTA are not consumed until the application is running. Therefore, if several applications are invoked at once, it is possible that PGFLQUOTA will be exhausted once the applications start up, without the error being detected by FileView. In this case, the applications can crash when the quota is exceeded.

When process creation fails due to quota exhaustion, FileView marks the task as Pending in the Work in Progress box until one of the running tasks has completed. The Pending task then becomes Active. If you try to start an additional task after the quota message has been displayed, the task is marked Pending, and the Work in Progress box pops up without a further warning message.

- If you paste a large amount of text into a FileView Task Output box while a text editor is running, the text might appear incorrectly. Press Ctrl/W to display the text correctly.
- If the total length of all the file names selected in the FileView window (including the device and directory name on each file) exceeds 65,535 characters, only a subset of the files is operated on when a verb is selected from a menu.

2.4.6 Help Widget Enhancement

V5.3

The default behavior for the help widget is now noncaching mode instead of caching mode. As a result, the help library starts up slightly faster, but requests for additional topics take longer.

2.4.7 Session Manager

The following sections contain information pertinent to the Session Manager.

2.4.7.1 Corrected Problems

The following Session Manager problems have been corrected:

General User Release Notes

2.4 DECwindows Notes

V5.2

- The Session Manager occasionally created one fewer than the selected number of terminal emulator windows during startup. This problem has been corrected; the correct number of terminal emulator windows are now created.
- The Session Manager could abort with an access violation on the second and subsequent attempts to use the Print Screen function. This problem has been corrected; the Session Manager no longer aborts in this context.

V5.4

- If the Print Screen output setting was set to “color,” applications could not be started on a monochrome head of a dual-head configuration. This problem has been corrected; applications can now be started in this context.
- When you start an application from the Applications menu of the Session Manager, the resources are now freed when the application terminates.
- The memory leak that occurred when an error box was displayed has been corrected.
- A user name entry longer than 100 characters no longer results in an access violation error.
- The application definition menu item resource processing has been changed to allow Kanji and MCS characters.
- You can now use the logical names listed in Table 2–2 to modify the quotas and flags used when you create the process to run the Window Manager. If the logical names are not defined (the default case), either a hard-coded value or the value of the SYSGEN parameter PQL is used.

Table 2–2 Session Manager Logical Names

Logical	Default Value
decw\$winmgr_pgflquota	5000
decw\$winmgr_bytlim	10,000
decw\$winmgr_diolm	100
decw\$winmgr_biolm	60
decw\$winmgr_astlm	100
decw\$winmgr_wsextent	4000
decw\$winmgr_fillm	92
decw\$winmgr_enqlm	600
decw\$winmgr_itquota	PQL value
decw\$winmgr_tqelm	PQL value
decw\$winmgr_wsquota	650
decw\$winmgr_wsdefault	512

(continued on next page)

General User Release Notes

2.4 DECwindows Notes

Table 2-2 (Cont.) Session Manager Logical Names

Logical	Default Value
decw\$winmgr_pswapm	PSWAPM flag is used ¹

¹To turn off the PSWAPM flag, define decw\$winmgr_pswapm to be any nonzero value.

- By default, the DECwindows logout code performs a security check to note the presence of other clients during the initial connection to the server. If it detects other clients, the DECwindows logout process exits.

Beginning with VMS Version 5.4, if the logical name DECW\$LOGIN_MULTIPLE is defined in executive mode in the system logical name table, the security check is bypassed. The translation of the logical name is not important; only the existence of the logical name is tested.

2.4.7.2 User-Defined Logo Support Added

V5.4

To substitute your own logo for the DIGITAL Logo displayed during login, you need to define a logical name and create a DCL command file that contains the commands to display your logo. The logical name translation must point to the command file. For example:

- 1 Enter the following command line, which requires SYSNAM privilege:

```
$ DEFINE/SYS/EXEC DECW$LOGIN_BACKGROUND SYS$MANAGER:MYLOGO.COM
```

- 2 In the SYS\$MANAGER directory, create the file MYLOGO.COM that contains the following command line:

```
$ RUN DECW$EXAMPLES:ICO.EXE
```

When you start DECwindows, logout creates a detached process in which SYS\$INPUT points to the command file that is defined by the translation of the logical name DECW\$LOGIN_BACKGROUND. The command file must contain commands that display the desired logo.

In VMS Version 5.3, the mode of the logo process was interactive. In VMS Version 5.4, the logo process is detached.

2.4.8 SYSGEN Parameter PQL_MPRCLM and Captive Accounts

V5.1

When you run AUTOGEN on a workstation that is running DECwindows, PQL_MPRCLM (the process quota Minimum Process Limit) is set to 8 to allow FileView to function with its subprocesses. Note that this parameter affects only workstations running DECwindows, even in a mixed cluster of workstations and other computers.

The *Guide to VMS System Security* recommends that you set the process limit to 0 for a captive account. This prevents a user from accessing DCL when running an application that allows a spawned process in a captive account. However, DECwindows Mail no longer allows a spawned process if the CAPTIVE flag is set in the account record; you do not need to follow this recommendation for captive accounts running Mail only.

General User Release Notes

2.4 DECwindows Notes

If you are setting up a captive account with access to other applications, you should check them to see whether spawned processes are allowed.

To override the DECwindows setting, add the following line to your SYS\$SYSTEM:MODPARAMS.DAT file:

```
PQL_MPRCLM = 0
```

Edit the SYS\$MANAGER:DECW\$CHECK_PARAMS.COM file so that it does not check for the setting of this SYSGEN parameter.

2.4.9 **ULTRIX — Running Applications from a Workstation**

V5.1

When a VMS server receives a DECnet connection initiated by an ULTRIX client, the user name associated with the connection is not usually the ASCII name of the user (for example, jqpublic). Rather, the user name is usually the user identification (UID) number in ASCII form (for example, 517).

ULTRIX applications differ; some require you to specify the user name and some require the UID. You should enter the following user name and UID authorization strings to the Security Customization menu to allow sample user jqpublic access to the VMS server (assume the ULTRIX node name is ULTRIX):

```
DECNET ULTRIX JQPUBLIC  
DECNET ULTRIX 517
```

Instead of using the preceding authorization strings, you could specify a wildcard to the Security Customization menu, as in the following example. However, the use of the wildcard character could allow other users to connect to the server and is therefore dangerous to security.

```
DECNET ULTRIX *
```

2.4.10 **ULTRIX — Authorization Guidelines for DECwindows Applications**

V5.4

When you use DECwindows to display remote applications running on an ULTRIX system to a VMS workstation by way of DECnet, you should authorize both the user name and the user ID of the user you want to authorize. The user ID is a number less than 32,000.

This authorization enables both applications that identify the user through the user name and applications that identify the user through the user ID to access the DECwindows server.

2.4.11 **VT1000 DECwindows Terminal Support**

V5.3-1

VMS Version 5.3-1 provides support for the VT1000 DECwindows terminal. For further information about using the VT1000 terminal with VMS, see the user's guide provided with the VT1000 terminal.

General User Release Notes

2.4 DECwindows Notes

2.4.12 Window Manager (Icon Box) — Corrected Problems

V5.3

The following corrections have been made to the Icon Box:

- The Icon Box is now centered on the screen according to the screen width.
- The grid has been disabled inside the Icon Box to support monitor independence. Windows remain visible and the icons move more smoothly. Note that the Window Manager prevents windows from being accidentally covered by icons by placing the icon outline next to the window, not on top of it.

2.5 DIGITAL Command Language (DCL) Notes

The release notes in this section pertain to the DIGITAL Command Language (DCL).

2.5.1 ANALYZE/ERROR_LOG Command — New Error-Logging Format

V5.2

The format of the system error messages logged to SYS\$ERRORLOG:ERRLOG.SYS has changed with VMS Version 5.2.

Now error-log messages cannot be analyzed (using the command ANALYZE /ERROR_LOG) by systems running versions of the VMS operating system prior to Version 5.1-1. Please analyze these messages on systems that are running VMS Version 5.2 and subsequent versions.

2.5.2 DCL Command Verb and Qualifier Length

V5.2

DCL currently checks only the first four characters of command verbs and qualifiers. Because of the continuing growth in the number of VMS products that use DCL command syntax, VMS is considering a change in which four characters may not be enough to identify all verbs or qualifiers. A sufficiently long transition period would precede any such change. Further details would be made available when the transition period begins.

When writing or modifying command procedures, or creating symbols for shorthand interactive use, it is important that you spell out the command syntax correctly. It is also recommended that you spell out the command in its entirety. This can prevent any problems or confusion if the four-character restriction is relaxed. This is also a good practice to follow in general because it helps to comment the command procedure and prevents ambiguity as new or updated products are installed on your system.

2.5.3 DCL Lexical Function F\$CONTEXT — Restriction Removed

V5.4

With VMS Version 5.2, the F\$CONTEXT lexical function was introduced with a restriction concerning the creation of a process context using the USERNAME and ACCOUNT selection items. For values that do not contain wildcard characters or that contain the single substitution character (%), the restriction required that the USERNAME selection value be a blank padded string of 12 characters, and that the ACCOUNT selection value be a blank padded string of 8 characters.

This restriction has now been removed.

2.5.4 DEFINE/FORM Command /SHEET_FEED Qualifier — Restriction

V5.0

Do not use the /PAGES qualifier to the PRINT command when submitting jobs to queues on which the DEFINE /FORM/SHEET_FEED command has been issued. When used with the /SHEET_FEED qualifier, the /PAGES qualifier causes the print symbiont to enter an infinite loop. The last page of the document prints repeatedly; the symbiont pauses after each page prints. If you encounter this problem, enter the following commands to stop and restart the queue:

```
$ STOP/QUEUE/RESET queue-name
$ START/QUEUE queue-name
```

2.5.5 DELETE and PURGE Commands — Corrected Problems

V5.4

With VMS Version 5.4, the following problems with the DCL commands DELETE and PURGE have been corrected:

- In previous versions, redundant translations in search lists caused PURGE to delete all versions of the specified file. The most notable case of this was in the system root directories. For example, if you entered the following commands, all versions of FILE.EXT would be deleted:

```
$ SET DEFAULT SYSS$MANAGER
$ PURGE [SYSMGR]FILE.EXT
```

This problem has now been corrected; redundant translations in search files no longer exist.

- The message codes generated by various errors detected by the DELETE and PURGE commands have been corrected, as follows:
 - In previous versions, purging to a foreign mounted volume generated a zero message code. Now, the following message code is generated:

```
%PURGE-W-FILNOTPUR, error deleting file-name
-PURGE-F-DEVFOREIGN, device is mounted foreign
```

Further, issuing a PURGE command to non-directory-structured devices (for example, terminals) now generates the following message rather than returning no error indication:

General User Release Notes

2.5 DIGITAL Command Language (DCL) Notes

`%PURGE-W-FILNOTPUR`, error deleting *file-name*
`-PURGE-F-IVDEVNAM`, invalid device name

- In previous versions, `DELETE` generated the following message for syntactically invalid file specifications, such as `FILE.EXT.GARBAGE`:

`%DELETE-E-DELVER`, explicit or wildcard version number required

Such errors now generate the following message code, followed by a more specific message:

`%DELETE-W-SEARCHFAIL`, error searching for *file-name*

- In previous versions, certain errors in purging a DECnet remote file, where the file specification included an access control string, generated error messages with the password displayed. This problem has been corrected; the password is now suppressed in the error messages.

2.5.6 **ENDSUBROUTINE Command — Correct Usage**

V5.4

The DCL commands `SUBROUTINE` and `ENDSUBROUTINE` define the beginning and end of a subroutine in a command procedure.

Beginning with VMS Version 5.4, every subroutine must end with the `ENDSUBROUTINE` command. Otherwise, a `CALL` command might not be able to locate label destinations. In this case, the following error message is displayed:

`%DCL-I-MSGNGENDS`, missing or misspelled `ENDSUBROUTINE` statement detected while scanning for label

For information about subroutine entry points, see Section 4.6.

2.5.7 **MOUNT Command — Correction to OPCOM Message**

V5.3-1

VMS Version 5.3-1 corrects an error in operator request messages issued by `MOUNT`. This error, introduced in VMS Version 5.2, attempted to correct an inconsistency in label processing between foreign mounted disks and tapes when a user specified a label. The following examples show this inconsistent behavior when processing labels. Example 2-1 illustrates mounting a foreign tape. Example 2-2 illustrates mounting a foreign disk. In both examples, the media has been initialized with the label `TEST`.

If `MOUNT` finds a valid ANSI label on the media when mounting a foreign tape, then that label is always used as shown in Example 2-1.

General User Release Notes

2.5 DIGITAL Command Language (DCL) Notes

Example 2-1 Mounting a Foreign Tape

```
$ MOUNT/FOREIGN MUA0: VAXVMS
%MOUNT-I-MOUNTED, TEST mounted on MUA0:
```

However, when mounting a foreign disk, the media label is not always used. When mounting a foreign disk, if MOUNT finds:

- A valid Files-11 label on the media, then the media label is used.
- A user-specified label in the command line, then that label is used in place of the media label. Example 2-2 illustrates this case. Because the user has specified the label VAXVMS in the command line, the VAXVMS label replaces the media label TEST.

Example 2-2 Mounting a Foreign Disk

```
$ MOUNT/FOREIGN DUA0: VAXVMS
%MOUNT-I-MOUNTED, VAXVMS mounted on DUA0:
```

Changes to the MOUNT command in Version 5.2 succeeded in making the behavior for disk and tapes consistent; however, it introduced an error in operator request messages by omitting the volume label from the operator request. Example 2-3 illustrates this error.

Example 2-3 Operator Request Message from MOUNT Command

```
$ MOUNT/FOREIGN $42$MUA0: VAXVMS
%MOUNT-I-OPRQST, Please mount device $42$MUA0: (HSC0)
%%%%%%%%%% OPCOM 8-NOV-1989 12:16:30.88 %%%%%%%%%%%
Request 16, from user SMITH on NODE1
Please mount device $42$MUA0: (HSC0)
```

In Example 2-3, MOUNT is unable to obtain the label and, therefore, displays the following default message:

```
%MOUNT-I-OPRQST, Please mount device $42$MUA0: (HSC0)
```

VMS Version 5.3-1 corrects this error in the operator request message. The correct message is illustrated in Example 2-4.

General User Release Notes

2.5 DIGITAL Command Language (DCL) Notes

Example 2-4 Corrected Operator Request Message from MOUNT Command

```
$ MOUNT/FOREIGN $42$MUA0: VAXVMS
%MOUNT-I-OPRQST, Please mount volume VAXVMS in
  device $42$MUA0:(HSC0)
%%%%%%%%%% OPCOM 8-NOV-1989 12:17:37.31 %%%%%%%%%%%
Request 17, from user SMITH on NODE1
Please mount volume VAXVMS in device $42$MUA0: (HSC0)
```

Note that the disk/tape inconsistency described in Example 2-1 and Example 2-2 has also been restored. Digital chose to restore the pre-Version 5.2 behavior in its entirety, rather than risk further destabilization in this area. Digital will reevaluate the disk/tape inconsistency for a future version of VMS.

2.5.8 OPEN Command — Negating Problem Corrected

V5.4

In previous versions of VMS, if you negated a qualifier using the DCL command OPEN, DCL appeared to negate the qualifier when, in fact, it did not.

For example, if you entered the following two commands, they were erroneously processed as the same command; consequently, both commands allowed shared access to FOO.TMP:

```
$ OPEN/SHARE FOO FOO.TMP
$ OPEN/NOSHARE FOO FOO.TMP
```

This situation has been corrected; the qualifiers to the OPEN command are now nonnegatable.

2.5.9 PRINT and SUBMIT Commands — Changes in Error Conditions

V5.3

With VMS Version 5.3, when qualifiers indicating file search criteria (for example, /SINCE, /EXCLUDE, or /BY_OWNER) are specified with either the PRINT or the SUBMIT command, and no files are found that meet the criteria specified by the qualifiers accompanying the two commands, the symbol \$STATUS indicates a fatal error, and one of the following pairs of messages is returned to the user:

```
%PRINT-F-CREJOB, error creating job
-JBC-E-EMPTYJOB, no files specified in job request
```

```
%SUBMIT-F-CREJOB, error creating job
-JBC-E-EMPTYJOB, no files specified in job request.
```

In previous versions of VMS, if no files were found that met the criteria specified by the qualifiers accompanying either the PRINT or SUBMIT command, VMS failed to display an error message to the user, and the symbol \$STATUS erroneously indicated a successful operation.

General User Release Notes

2.5 DIGITAL Command Language (DCL) Notes

2.5.10 SET ACL Command Qualifiers — Clarifications and Enhancements

V5.4

Previous releases of VMS contained discrepancies among the documented, intended, and observed behaviors of certain SET ACL commands.

VMS Version 5.4 corrects past discrepancies in SET ACL behavior and enhances the SET ACL command with new qualifiers.

Table 2-3 lists the corrected problems with and enhancements to the SET ACL command; see the *VMS DCL Dictionary* for Version 5.4 detailed descriptions.

Table 2-3 SET ACL Command Qualifiers — Corrected Problems and Enhancements

Qualifier	Description/Enhancement
/LIKE	Now deletes all protected entries. The behavior for SET ACL used with the /LIKE qualifier was documented incorrectly in previous versions of the <i>VMS DCL Dictionary</i> . The documentation now reflects the intended behavior: Protected entries that were originally set on the destination object are not retained but are deleted to remove any possible ambiguity from the operation. In previous VMS versions, the observed behavior sometimes matched the incorrect SET/ACL documentation because of a problem with the SET /ACL command. That problem has been corrected.
/NEW	As in the preceding description of the /LIKE qualifier, the observed behavior of the /NEW qualifier sometimes matched the incorrect SET/ACL documentation because of a problem with the SET/ACL command. The /NEW qualifier now deletes all protected entries.
/DELETE	When used for ACL deletion, continues to behave as documented in previous VMS versions. However, /DELETE=ALL now deletes the entire ACL.
/BACKUP	New for VMS Version 5.4. Modifies the time value from /SINCE or /BEFORE to select files according to their most recent backup.
/EXPIRED	New for VMS Version 5.4. Modifies the time value from /SINCE or /BEFORE to select files according to their expiration date.
/MODIFIED	New for VMS Version 5.4. Modifies the time value from /SINCE or /BEFORE to select files according to their last modification date.

(continued on next page)

General User Release Notes

2.5 DIGITAL Command Language (DCL) Notes

Table 2-3 (Cont.) SET ACL Command Qualifiers — Corrected Problems and Enhancements

Qualifier	Description/Enhancement
Qualifier combinations	<p>In previous versions, meaningless qualifier combinations were accepted, sometimes with unpredictable results.</p> <p>The legal combinations are now more strictly enforced, which could result in command line errors in existing command procedures.</p>

2.5.11 SET HOST/DTE Command — Modifications

V5.4

The SET HOST/DTE command uses the VMS terminal driver to provide flow control to other systems. This corrects a resource contention problem in VMS Version 5.0 that occasionally caused data overrun errors.

Once SET HOST/DTE receives 100 buffers of data, it stops reading from the specified terminal. As soon as the type-ahead buffer is full, the VMS terminal driver sends an XOFF flow control message. Once SET HOST/DTE has displayed most of the data, it starts reading from the terminal again.

The SET HOST/DTE command has been modified for VMS Version 5.4 and includes the following enhancements:

- You can now control the configuration of a connection to a remote system.
- New qualifiers enable you to select all configuration characteristics, such as XON/XOFF flow control, maximum number of buffers, read delay, and parity.
- A new interactive command mode enables you to configure the SET HOST/DTE session while the session is in progress.

For more information about the SET HOST/DTE command, see the *VMS DCL Dictionary*.

2.5.12 SHOW LOGICAL/FULL Command — Correction

V5.4

Prior to VMS Version 5.4, the SHOW LOGICAL/FULL command did not display the access control list (ACL) for a logical name table. This problem has been corrected; the SHOW LOGICAL/FULL command now displays ACLs for logical name tables.

2.5.13 SHOW MAGTAPE Command Now Obsolete

V5.4

With VMS Version 5.4, the DCL command SHOW MAGTAPE is obsolete. SHOW MAGTAPE has been superseded by the command SHOW DEVICES /FULL.

2.5.14 **SUBMIT/DELETE Command — Modifications**

V5.3

With VMS Version 5.3, if you issue a SUBMIT/DELETE command and include in that command's parameter list the name of a file to which you do not have delete (D) access, the SUBMIT command processing stops and no batch job is created. This is also true if you issue the SUBMIT command and include the /DELETE qualifier after the name of a file to which you do not have delete (D) access.

In previous versions of VMS, if you were denied delete (D) access to one or more files that you included as parameters to the SUBMIT/DELETE command, processing specified in the SUBMIT command continued. If any files that you specified in the SUBMIT command were successfully added to the job request (the file or files that you had unsuccessfully designated for deletion were not included in the job request), a batch job was created and entered in the appropriate queue.

This same behavior happens if you issue a SUBMIT command, and an error occurs because VMS cannot find or open the file you specified. Previous versions of VMS, as well as Version 5.3, stop processing the SUBMIT command and do not create a batch job if any file in its parameter list cannot be opened as input.

2.5.15 **Symbol Names — Use Caution When Making Symbol Name Assignments**

V5.0

Digital recommends that you use caution when assigning a symbol name that is already a DCL command name. Digital especially discourages the assignment of symbols such as IF, THEN, ELSE, and GOTO, which can affect the interpretation of command procedures.

V5.4

In VMS Version 5.4, a command procedure that contains the symbol or label named DECK can prevent a GOTO, GOSUB, or CALL command from locating the label destination. Also, a command procedure might fail to skip over a subroutine definition during execution.

2.6 **DIGITAL Standard Runoff (DSR) — Variant Name Change**

V5.2

In DIGITAL Standard Runoff, the maximum length of a variant name (/VARIANT=*string-qualifier*) has been changed from 15 characters to 31.

2.7 **Directory Entries — Primary as Opposed to Alias**

V5.0

VMS Version 5.0 distinguishes between primary directory entries (the directory entries created when files are created) and alias directory entries (entries created with the DCL command SET FILE/ENTER or similar operations). Every file has a back link that identifies its primary directory entry.

Following is a summary of the directory entry changes in VMS Version 5.0:

- ANALYZE/DISK_STRUCTURE no longer reports invalid back link errors on files with alias directory entries.

General User Release Notes

2.7 Directory Entries — Primary as Opposed to Alias

- Only primary directory entries are subject to the special directory entry protection rules described in the *Guide to VMS System Security*.
- When a primary directory entry is deleted, the file contents are deleted. Any remaining alias directory entries point to a nonexistent file.
- When an alias directory entry is deleted, either by the DELETE or the SET FILE/REMOVE command, only the directory entry is removed; the file contents remain.
- When a primary directory entry is removed with a SET FILE /REMOVE command, the file remains but no longer has a back link. From then on, alias directory entries are treated as if they were primary entries.
- If a new directory entry is made for a file with no back link, for example by SET FILE/ENTER or RENAME, the new directory entry becomes the primary entry.

A problem exists in VMS Version 5.0 in the treatment of multiple directory entries for the same file when they are located in the same directory file. When a file or directory entry is being deleted, the DELETE command always locates the directory entry of the file that occurs first in the directory, not necessarily the one named in the command. Consider the following sequence of commands:

```
$ CREATE [A]X.X  
$ SET FILE [A]X.X/ENTER=[A]Y.Y  
$ DELETE [A]Y.Y;
```

In this case, rather than removing the alias entry Y.Y, the DELETE command removes the primary entry X.X and deletes the file contents. This problem will be corrected in a future release of the VMS operating system.

2.8 EDT Editor — TMP Files Renaming Problem

V5.0

Because VMS Version 5.0 requires the owner of a file to have delete access to that file in order to allow renaming, EDT does not function properly if the owner does not have delete access to the original file.

The problem occurs when you edit a file and EDT creates a TMP file with the same protection mask as your source file. When you try to exit, EDT tries to rename the TMP file to the next higher version of the source file and fails.

This problem will be corrected in a future release of the VMS operating system. You can avoid the problem by making sure you have delete access to the file before you begin your editing session. Also, you can use the EDT command WRITE followed by the EDT command QUIT. WRITE creates a new file without renaming, and QUIT exits without saving (by renaming).

2.9 Ethernet/802 Controllers — New Support

V5.4 The VMS Version 5.4 operating system supports the following new Ethernet/802 controllers:

- DEMNA (DECLANcontroller 400)
- Second-Generation Ethernet Controller (SGEC)

For more information, see Sections 3.23.2 and 3.23.6.

2.10 Process Identification (PID) — All Significant Digits Must Be Specified

V5.2 All system services that control processes or obtain information about processes use **pidadr** as the first argument and **prcnam** as the second argument. These two arguments are also used to reference remote processes.

The process identification (PID) is unique for each process in the cluster. To reference a process on another node in the cluster, specify its PID as the **pidadr** argument. Prior to VMS Version 5.2, you could abbreviate a PID by omitting the high-order field. However, because the high-order field is now used to identify the node, you must specify all the significant digits of a PID; you can omit leading zeros.

For example, prior to VMS Version 5.2, you could specify the following to stop the process with the PID 48400136:

```
$ STOP/IDENTIFICATION=136
```

With VMS Version 5.2, you must specify the following, or you will receive a status code message warning you that this is a nonexistent process (SS\$_NONEXPR):

```
$ STOP/IDENTIFICATION=48400136
```

The process name has also been extended to reflect clusterwide accessibility. To access information about a remote process, the node name must be prefixed to the process name. For example, to reference the process BATCH_69 on node NNAME, use the name NNAME::BATCH_69.

This change in process naming has the following implications:

- Process name strings can be up to 23 characters long.
 - 6 characters for the node name
 - 2 characters for the double colon (::) that follows the node name
 - 15 characters for the process name
- A local process name can look like a remote process name. Therefore, if you specify ATHENS::SMITH, the system checks for a process named ATHENS::SMITH on the local node before checking node ATHENS for a process named SMITH.

General User Release Notes

2.11 RA92 DSA Disk Support

2.11 RA92 DSA Disk Support

V5.3-2

VMS Version 5.3-2 provides support for the RA92 DSA disk. The RA92 DSA disk is used with the SA800 and SA850 storage arrays. DSA is an evolving architecture that establishes interface standards and protocol requirements, and assures compatibility across product families. For information about programming the RA92 DSA disk, refer to Section 4.25.

2.12 Sort/Merge Utility — Changes and Enhancements

V5.2

The following are changes and enhancements to the Sort/Merge Utility:

- The specification file keyword `/FIELD` has been enhanced to allow you to define constants. In the following example, the value of the constant `zip_code` is defined as 03060.

```
/FIELD=(NAME=zip_code,SIZE:5,VALUE="03060",CHARACTER)
```

- The performance of the Sort/Merge Utility has been improved by 26 percent. A major factor in this improvement is that the Sort/Merge Utility now allows asynchronous reads on scratch files and input files.
- SORT scratch files are entries in the `SYS$SCRATCH` directory, rather than temporary files as defined by the VMS Record Management Services (RMS).
- Specification file fields that are defined as floating now allow the use of a decimal point. For example:

```
/COND=(NAME=A,TEST=(B EQ 150.40))
```

2.13 System Messages — New and Modified for VMS Version 5.4

V5.4

The following VMS facilities have new or modified system message information for VMS Version 5.4:

- BUGCHECK (System Bugcheck)
- DISMOUNT (DISMOUNT command)
- INIT (INITIALIZE command)
- LMCP (Log Manager Control Program)
- MOUNT (Mount Utility)
- OPCOM (Operator Communication Process)

For a detailed description about the new or modified system messages, see Appendix A.

2.14 TA90E Tape Drive Support

V5.3-2

Beginning with VMS Version 5.3-2, VMS supports the TA90E tape drive, which is an enhanced version of the TA90 tape drive. For information about using the TA90E tape drive, refer to Section 3.48.

2.15 Tape Support — Acceptance of ANSI Initialized Magnetic Tapes

2.15 Tape Support — Acceptance of ANSI Initialized Magnetic Tapes

V5.4

In previous versions of VMS, the only initialized empty magnetic tape volumes accepted as standard label tapes were those that included an empty file following the volume labels, which is the tape format produced by the DCL command INITIALIZE.

Beginning with VMS Version 5.4, magnetic tape volumes that have been initialized according to the recommended format in the appendix to the ANSI X3.27-87 standard are also accepted as standard output tapes by MOUNT, BACKUP, and the magnetic tape ACP. This new initialized volume format consists of a beginning-of-volume label group followed by two tape marks.

2.16 TLZ04 Tape Drive Notes

The release notes in this section pertain to the TLZ04 tape drive.

2.16.1 TLZ04 Tape Drive Now Supported

V5.4

VMS Version 5.3-2 provides support for the TLZ04 tape drive. The TLZ04 tape drive is for data storage only and is intended for use with the Backup Utility only. It is not intended to replace existing bootable media, such as the TZ30 or TK50 tape drives. See Section 4.31 for information about programming the TLZ04.

2.16.2 TLZ04 Performance Issues

V5.4

Important performance issues should be considered when using the TLZ04 tape drive. The TLZ04 tape drive uses helical scan technology. Because of this technology, the TLZ04 tape drive performs certain operations more slowly than other tape drives, such as the TK50 tape drive.

For example, the TLZ04 tape drive is approximately 5 to 10 times slower than the TK50 tape drive when performing copy operations. For backup operations, however, the TLZ04 tape drive is approximately 3 to 4 times faster than the TK50 tape drive.

2.17 VAX 6000-Series Computers

V5.4

The VMS Version 5.4 operating system supports the vector processing option available on existing VAX 6000-400 series computers. For complete information about using VAX 6000-400 series computers in a vector processing environment, see the *VMS Version 5.4 New Features Manual*.

2.18 VAX 9000-Series Computers

V5.4

The VAX 9000-series computers are new, multiuser mainframe systems that are fully compatible with other VAX computer software, including VMS software, optional software, and applications software. These high-performance computers utilize vector processing and can be used as symmetric multiprocessors.

General User Release Notes

2.18 VAX 9000-Series Computers

For more information on the VAX 9000-series computers, see the *VMS Version 5.4 New Features Manual*.

2.19 VAXft 3000 System Notes

The release notes in this section pertain to VAXft 3000 systems.

The VAXft 3000 computer is a new fault-tolerant system that uses VAXft System Services. This manual and the *VMS Version 5.4 New Features Manual* contain supplementary information.

For detailed information on the VAXft 3000 computer, see the VAXft System Services documentation and the *VAXft System Services Software Product Description*.

2.19.1 VAXft 3000 System Messages Displayed

V5.4

The VAXport and VAXport Driver Facility displays messages from the PW driver of a VAXft 3000 system. These messages contain information that describes the state of the PW port.

For example, the following message originates from the PW driver and explains that the PW driver is reinitializing the port and restarting port operations:

```
% PWx0, port is reinitializing
```

2.19.2 VAXft System Services — Required HELP Command Line

V5.4

To access VMS HELP for the DCL commands relating to VAXft system services, VMS Version 5.4 requires that you enter the following command line rather than the usual HELP command syntax:

```
$ HELP FTSS DCL_COMMANDS command Return
```

All DCL commands relating to VAXft system services are then displayed as subtopic options. These DCL commands are:

```
START/ZONE  
STOP/ZONE  
SHOW ZONE
```

2.20 VAX Text Processing Utility (VAXTPU) Notes

The release notes in this section pertain to the VAX Text Processing Utility (VAXTPU).

2.20.1 /NOWORK Qualifier Problem

V5.4

For VMS Version 5.4, the function of the VAXTPU qualifier /NOWORK is documented as preventing VAXTPU from using a work file.

General User Release Notes

2.20 VAX Text Processing Utility (VAXTPU) Notes

In fact, VAXTPU opens a work file with the extension TPU\$WORK_FILE. There is no way to prevent VAXTPU from attempting to open a work file. This problem will be corrected in a future release of VMS.

2.20.2 /WORK Qualifier Problem

V5.4 With VMS Version 5.4, there is a problem with the VAXTPU qualifier /WORK. If you attempt to supply an invalid file name to the /WORK qualifier, VAXTPU outputs an error message and then exits.

This problem will be corrected in a future release of VMS.

2.20.3 Display Manager Definition Restriction

V5.3 Do not define the logical name TPU\$DISPLAY_MANAGER to be DECWINDOWS. When this is done, VAXTPU experiences an access violation the second time it is called during a session by another application, such as the Mail Utility (MAIL). There is no way to work around this problem except to avoid this definition of TPU\$DISPLAY_MANAGER. This problem will be corrected in a future release of the VMS operating system.

2.21 VMS Mail Utility — Folder Name Parameter Now Supports Mixed Case

V5.1 The VMS Mail Utility folder name parameter now supports mixed case when you enclose the name within double quotation marks. Starting with Version 5.0, when you specified a folder name, it was changed to all capitals even if you used quotation marks. Before Version 5.0, if you quoted the folder name, it was left in mixed case. Version 5.1 restores the Version 4.x capability and supports mixed case in quoted folder names.

A folder name can be 1 to 39 characters in length. Valid characters for folder names are A through Z, a through z, dollar sign (\$), underscore (_), hyphen (-), and 0 through 9. To retain mixed case, enclose the folder name within quotation marks.

3 System Manager Release Notes

This chapter contains information about VMS Version 5.4 that is of interest to the system manager.

For information about the new features included in VMS Version 5.4, see the *VMS Version 5.4 New Features Manual*.

3.1 \$CREATE_RDB System Service — New Location for Rights Database

V5.4

You can use the \$CREATE_RDB system service to create a rights database from a user program.

In previous versions of VMS, the rights database file was created in the SYS\$SPECIFIC:[SYSEXE] directory. The SYS\$SPECIFIC:[SYSEXE] directory was transparent to the user unless the directory was located in a cluster environment with a common system disk, where each node would see a different rights database.

With VMS Version 5.4, a new rights database file is created in the SYS\$COMMON:[SYSEXE] directory. To minimize confusion, users who find the rights database file in the SYS\$SPECIFIC:[SYSEXE] directory (as a result of previous VMS versions) should move it to the SYS\$COMMON:[SYSEXE] directory. (Note that this change to the SYS\$COMMON:[SYSEXE] directory is not necessary if the RIGHTSLIST logical name is correctly defined.)

3.2 Analyze/Disk_Structure Utility Notes

The release notes in this section pertain to the Analyze/Disk_Structure Utility.

3.2.1 ANALYZE/DISK_STRUCTURE — Corrected Problems

V5.4

The following ANALYZE/DISK_STRUCTURE problems have been corrected in VMS Version 5.4:

- ANALYZE/DISK_STRUCTURE issues a report of multiple allocated blocks sorted by logical block number (LBN). In previous versions, the utility did an on-disk sort using SYS\$SCRATCH. Since the utility locks the volume being analyzed, this meant:
 - a. A user who was analyzing the volume pointed to by the logical name SYS\$SCRATCH had to redefine SYS\$SCRATCH.
 - b. Single-disk systems could not analyze their disks.

These restrictions have been removed. Users can now analyze the volume pointed to by SYS\$SCRATCH because the utility uses an in-memory sort.

System Manager Release Notes

3.2 Analyze/Disk_Structure Utility Notes

- In previous versions, running the utility produced the following message in certain circumstances:

`%VERIFY-W-ALLOCSET, blocks incorrectly marked free`

Running the utility a second time produced the following message for the same circumstance:

`%VERIFY-W-ALLOCCLR, blocks incorrectly marked allocated`

Only on the third run of the utility did the message disappear. These messages were false and resulted from incorrect processing of marked-for-delete files.

This problem has been corrected; the incorrect messages no longer appear.

- In previous versions, if you ran the utility and did not have privilege to access a directory, that directory's contents was treated as lost and was moved to the SYSLOST directory.

The utility now detects this condition and does not move any files to [SYSLOST]. Instead, the following error message is issued:

`%ANALDISK-W-NOPRIVDIRSUM, some directories protected against access,
cannot recover lost files`

- In previous versions, the utility could not repair a corrupted boot block. This problem has been corrected; the utility can now repair a corrupted boot block.
- In previous versions, if you used the /CONFIRM qualifier, you might be asked the following question even when no error was reported:

Repair this error? (Y or N):

This problem has been corrected; the question is no longer asked.

- When a bad file header is detected, the utility now gives more information in the header about the particular error.

3.2.2 Facility Code Change

V5.4

For VMS Version 5.4, the facility code for ANALYZE/DISK_STRUCTURE error messages has been changed from VERIFY to ANALDISK, as follows:

- **Previous code:**

`%VERIFY-I-OPENQUOTA, error opening QUOTA.SYS`

- **New code:**

`%ANALDISK-I-OPENQUOTA, error opening QUOTA.SYS`

3.3 Audit Analysis Utility — /SELECT=STATUS=FAILURE Qualifier Problem

V5.4 For VMS Version 5.2 and subsequent versions, the /SELECT=STATUS=FAILURE qualifier to the ANALYZE/AUDIT command does not function correctly. To work around this problem, specify an explicit failure value using the qualifier /SELECT=STATUS=CODE=*value*.

3.4 Authorize Utility Notes

The release notes in this section pertain to the Authorize Utility.

3.4.1 Adding Proxy Accounts

V5.4 When you add user entries to the network proxy authorization file using the ADD/PROXY command or when you update user entries using the MODIFY/PROXY command, the following occurs:

- The Authorize Utility signals DECnet to update its volatile database.
- Proxy additions or modifications take effect immediately on all nodes in a cluster that share the proxy database.

For more information on the ADD/PROXY and MODIFY/PROXY commands, see the *VMS Authorize Utility Manual*.

3.4.2 Modifications to the RESTRICTED Flag

V5.4 In VMS Version 5.2, the VMS operating system placed restrictions on the CAPTIVE flag in the user authorization file (UAF) to make it easier for site security administrators to write secure command procedures. To maintain compatibility with existing command procedures that depend on the old behavior of the CAPTIVE flag, VMS Version 5.2 introduced the RESTRICTED flag.

In a future release of VMS, system software components will be modified so they do not use the RESTRICTED flag to disable SPAWN commands. In particular, MAIL and TPU will not disable a SPAWN command or built-in if the account has been marked RESTRICTED. However, these facilities will disable SPAWN commands for captive accounts.

Third-party software designers are encouraged to remove any SPAWN restrictions from the RESTRICTED flag and to use the CAPTIVE flag instead. The future purpose of the RESTRICTED flag will be to ensure that an account completely executes all the commands contained in the system login procedure (SYS\$MANAGER:SYLOGIN.COM), the user's login procedure (LOGIN.COM), and any command procedures invoked from these two procedures. However, once the login sequence has been completed, a restricted account will be indistinguishable from an unrestricted account.

System Manager Release Notes

3.4 Authorize Utility Notes

3.4.3 Setting the DISCTRLY Flag

V5.3-2

Due to an error in VMS Version 5.2, the login procedure did not set the DISCTRLY flag properly when you logged into an account with either the CAPTIVE or RESTRICTED flags set.

This problem was corrected in VMS Version 5.3-1. Digital recommends that site-security administrators of sites running VMS versions prior to Version 5.3-1 manually set the DISCTRLY flag for all accounts that are marked either CAPTIVE or RESTRICTED.

3.4.4 Submitting Batch Jobs

V5.3-2

Prior to VMS Version 5.3-2, batch jobs submitted in a captive account would fail with a CAPTINT error. Likewise, network connections to captive accounts would fail with a "network partner exited" message on the initiating node. These batch restrictions have been removed in VMS Version 5.3-2.

Digital now recommends that you use the MODIFY/NOBATCH and MODIFY/NETWORK commands in AUTHORIZE to disable BATCH and NETWORK access in captive (or restricted) accounts. With the restrictions removed, network server objects now function properly with the CAPTIVE flag set.

For more information about captive and restricted accounts, see the *Guide to VMS System Security*.

3.5 AUTOGEN Command Procedure Notes

The release notes in this section pertain to the AUTOGEN command procedure.

For information about recent changes to AUTOGEN, see the *VMS Version 5.4 New Features Manual*.

3.5.1 Feedback Mechanism — Additional Data Files Used

V5.0

AUTOGEN now uses two new data files, SYS\$SYSTEM:AGEN\$ADDHISTORY.TMP and AGEN\$ADDHISTORY.DAT. These files are used in conjunction with the feedback mechanism to avoid accumulating the values of ADD_ symbols found in MODPARAMS.DAT and VMSPARAMS.DAT from one invocation of AUTOGEN to the next. These files should not be deleted or modified.

3.5.2 Hexadecimal Values Processed Correctly in MODPARAMS.DAT

V5.0

Parameters set to hexadecimal values in MODPARAMS.DAT and that correspond to a negative decimal value are now processed correctly. For example, the MODPARAMS.DAT record "SMP_CPUS=%XFFFFFFFF" results in the parameter SMP_CPUS being set to -1.

3.5.3 Mechanism to Control MSCP Server Buffer Size is Obsolete

V5.0

Prior to Version 5.0, a system manager could define internal AUTOGEN symbols of the form MSCP_* (for example, MSCP_BUFFER) in MODPARAMS.DAT to control the amount of nonpaged pool allocated to the MSCP server for buffer space. AUTOGEN recognized these symbols and set up the corresponding bit fields in the parameters VMS5 and VMS6. This was necessary because the MSCP server must be loaded early in the boot sequence on local area VAXcluster boot nodes.

Beginning with Version 5.0, the method used to load the MSCP server has changed. The early loading of the server on local area VAXcluster boot nodes is still required; it is accomplished using the new SYSGEN parameter MSCP_LOAD, which can have either of the following values:

0	Do not load the server (default)
1	Load the server

If MSCP_LOAD is 1, the amount of nonpaged pool allocated to the server's buffer is controlled by another new SYSGEN parameter, MSCP_BUFFER, which is expressed in pages (128 pages is the default).

If you have defined any of the old AUTOGEN internal symbols in MODPARAMS.DAT, they should be removed because they may conflict with the new SYSGEN parameters.

For more information, refer to the *VMS System Generation Utility Manual*.

3.5.4 OLDSITE Mechanism is Obsolete

V5.0

The OLDSITE mechanism for propagating parameter values is now obsolete. Make sure that SYS\$SYSTEM:MODPARAMS.DAT contains a record for any SYSGEN parameter that AUTOGEN does not calculate that requires a value different from its default.

3.5.5 Prefix ADD_ Can No Longer Subtract Values for Parameters Generated by FEEDBACK

V5.2

For FEEDBACK-generated parameters *only*, AUTOGEN does not let you subtract parameter values by using the prefix ADD_ in MODPARAMS.DAT. For example, the following command has no effect on the parameter GBLPAGES:

```
ADD_GBLPAGES = -20
```

Likewise, if you use the ADD_ prefix to replace the value of a parameter with a smaller value, the net subtraction is also not permitted. For example, replacing: ADD_GBLPAGES=200 with ADD_GBLPAGES=100 has no effect on the parameter GBLPAGES.

Note: AUTOGEN FEEDBACK has a mechanism to decrease parameter values based on the system load. Users who want to decrease FEEDBACK-generated parameters further can do so by entering a MAX_ prefix for that parameter in MODPARAMS.DAT, or by

System Manager Release Notes

3.5 AUTOGEN Command Procedure Notes

explicitly setting the parameter value (*parameter = parameter-value*). Digital strongly recommends that these entries be removed from MODPARAMS.DAT after a single execution of AUTOGEN, so that FEEDBACK can resume calculating the parameter.

3.5.6 System Files Not Marked for NOBACKUP

V5.0

AUTOGEN warns you if it finds an existing page, swap, or dump file that is *not* marked for NOBACKUP; these items are probably unnecessarily increasing your disk backup time. If you receive this warning, perform the following steps:

- 1 Make sure that at least one viable page file remains.
- 2 Rename the file in question so that it will not be installed during the next boot.
- 3 Shut down and reboot the system.
- 4 Use the SET FILE/NOBACKUP command to mark the file NOBACKUP.
- 5 Rename the file to its original name.
- 6 Shut down and reboot the system.

3.5.7 Swap File Size Changes

V5.0

The VMS memory management swap file allocation algorithm has been changed significantly. A swap slot is allocated only when a process is selected as an outswap candidate. The swap slot need not be virtually contiguous or contained in one file. This means that swap file requirements will be significantly less than for previous versions of the VMS operating system.

As a result, modifications have been made to the sizing algorithm for AUTOGEN's swap file. This allows AUTOGEN to produce much smaller swap files, typically only one quarter of what would have been produced under Version 4.n. If you have overridden AUTOGEN's swap file calculation by defining the symbol SWAPFILE=0 in MODPARAMS.DAT, remove this symbol definition in order to let AUTOGEN create a smaller swap file.

3.5.8 Switching Window Systems

V5.2

With VMS Version 5.2, if you want to switch from VWS to DECwindows (or vice versa), you must reboot your system twice in order for AUTOGEN to set system parameters appropriately.

Note: When switching window systems, first delete all AUTOGEN FEEDBACK data files. This is necessary because DECwindows and VWS FEEDBACK parameters are not compatible. Prior to executing AUTOGEN NOFEEDBACK, enter all layered product and third-party software parameter requirements into the file MODPARAMS.DAT.

System Manager Release Notes

3.5 AUTOGEN Command Procedure Notes

The command sequence is as follows:

1 First delete all AUTOGEN FEEDBACK files:

```
$ DELETE SYS$$SYSTEM:AGEN$*.DAT
$ RUN SYS$$SYSTEM:SYSGEN
SYSGEN> SET WINDOW SYSTEM 1      ! or 2 for VWS
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
```

2 Reboot the system using the following command:

```
$ @SYS$$SYSTEM:SHUTDOWN
```

3 After the reboot, execute AUTOGEN using the following command:

```
$ @SYS$$UPDATE:AUTOGEN GETDATA REBOOT NOFEEDBACK
```

4 The system will automatically reboot using the newly generated DECwindows or VWS parameters.

Regular execution of AUTOGEN FEEDBACK will ensure that system parameters reflect user load. After the system has been running a typical load for at least 24 hours, invoke AUTOGEN FEEDBACK as follows:

```
$ @SYS$$UPDATE:AUTOGEN SAVPARAMS your-favorite-end-phrase
```

Note: If you want to change window system more than once, save copies of your system parameter files (SYS\$\$SYSTEM:VAXVMSSYS.PAR) for both DECwindows and VWS. You can subsequently change window system by a conversational boot using the appropriate parameter file. These saved versions should be kept up to date (that is, after running AUTOGEN through SETPARAMS, save the newly generated parameter file SYS\$\$SYSTEM:VAXVMSSYS.PAR).

3.6 Backup Utility Notes

The release notes in this section pertain to the VMS Backup Utility (BACKUP).

3.6.1 Backing Up Files Marked for Recovery Unit Journaling

V5.4

The Backup Utility cannot back up a file marked for recovery unit journaling if the file has active transactions. If you encounter this problem during a backup procedure, you should attempt to access the file using another utility.

For example, you can access the file with the DCL command OPEN. The attempt to open the file causes detached recovery to restore records modified during the transaction to their states before the transaction began. If detached recovery succeeds, the OPEN command succeeds and you can proceed with the backup procedure. If detached recovery fails, the OPEN command fails and detached recovery returns error messages to the terminal and to the operator communication process (OPCOM). For more information, see the *VAX RMS Journaling Manual*.

If you use the OPEN command to trigger detached recovery on a file, be sure to close the file afterward with the CLOSE command.

System Manager Release Notes

3.6 Backup Utility Notes

3.6.2 Backup Utility Problems

V5.4

The following BACKUP problems exist in VMS Version 5.4. These problems will be corrected in a future release of VMS.

- When you specify dates for the year 2000 or greater using the /TAPE_EXPIRATION qualifier, the tape is shown as expired. To work around this problem, do not set expiration dates greater than December 31, 1999.
- A problem exists with Backup disk journal file names larger than 31 characters. If the journal file name is larger than 31 characters but less than 35 characters, the integrity of the journal file may be questionable. A Backup journal file name larger than 35 characters can cause an access violation in BACKUP. To work around this problem, use journal file names of less than 31 characters.
- Any file within a directory file that has a file ID larger than 65,535 may generate an error during either the /RECORD or the /DELETE pass. If the BACKUP operation is requesting a DELETE pass, the way to work around this problem is to manually delete the files that were not deleted.
- A data-integrity problem may exist when you perform a BACKUP save of a disk that has high-water marking enabled. To detect a problem during the Backup save operation, use the /VERIFY qualifier. The following error message is displayed:

`%BACKUP-E-VERIFYERR, verification error for block n of file`

To work around this problem, disable high-water marking on the volume before you back up the disk, then enable it again when the backup operation is complete.

3.6.3 Corrected Backup Utility Problems

V5.4

The following BACKUP problems have been corrected for VMS Version 5.4:

- In previous versions of VMS, save-set names that exceeded 17 characters and were written to tape failed at the verify pass and an error message was displayed. This problem occurred because the save-set names were truncated and therefore could not be located. This truncation problem has been corrected; save-set names exceeding 17 characters no longer fail at the verify pass.
- In previous versions of VMS, significant performance degradation occurred on computers that did not emulate the cyclic redundancy check (CRC) instructions. This performance problem has been corrected through a longword alignment of the BACKUP context structure; performance degradation no longer occurs.
- In previous versions of VMS, answering Yes to the question posed in the following message caused BACKUP to terminate with a reserve operand fault:

System Manager Release Notes

3.6 Backup Utility Notes

```
%BACKUP-I-RESUME, resuming operation on volume 2
%BACKUP-I-READYWRITE, mount volume 2 on _CSID$MUA0: for writing)
Enter "YES" when ready: Yes
```

This termination problem has been corrected; a reserve operand fault no longer occurs.

- In previous versions of VMS, in some cases standalone BACKUP became suspended when you used the /LOG qualifier. This qualifier problem has been corrected; standalone BACKUP no longer becomes suspended when you use the /LOG qualifier.
- In previous version of VMS, a data-integrity problem existed with files that have extension headers. To detect the data-integrity problem during the save operation, use the /VERIFY qualifier. BACKUP then issues a VBNMISSING error message on files that exhibit the problem. To determine if a file contains extension headers, issue the following command:

```
$ DUMP/HEADER/BLOCK=C:0 disk:[directory]filename.extension
```

If the value of the extension file identification field is nonzero, the file contains extension headers.

3.6.4 **Forced Error Handling**

V5.0

Most VMS utilities and DCL commands treat a forced error flag as a fatal error. For example, if you use the DCL command COPY to move a file that contains a block with the forced error flag, the resulting error causes the operation to terminate.

The Backup Utility, however, is designed to continue in the presence of almost all errors, including forced errors; BACKUP continues to process the file, creating a new copy of the file in the output save set. An error message indicating the forced error is displayed, but the forced error is not present in the new copy of the file that is being created. Subsequent use of the new file (for example, in a restore operation) will indicate no errors. Thus, data that was formerly marked as bad with the forced error flag may be accidentally propagated and seem correct.

System managers (and other users of BACKUP) should assume that forced errors reported by BACKUP signal degradation of the data in question and should act accordingly. The safest procedure is to replace the file containing the forced error with a good copy of the file from a previous backup operation.

For more information on DIGITAL Storage Architecture (DSA) and forced errors, see the *VMS I/O User's Reference Manual: Part I*.

3.6.5 **Image Save Operation Restriction**

V5.4

With VMS Version 5.4, BACKUP does not support the verify operation with disk-to-disk image backups or with BACKUP copy. (Backups from disk-to-tape still support the verify operation.)

System Manager Release Notes

3.6 Backup Utility Notes

This restriction will be removed in a future release of VMS. To ensure data integrity, use the BACKUP/COMPARE command after the BACKUP save is complete.

3.6.6 ODS-1 Disk Structures — Corrected Backup Problems

V5.4

Prior to VMS Version 5.4, the following problems existed when backing up files from an ODS-1 disk structure. Both of the problems are more apparent on fragmented disk structures.

- The block count for each header extent was calculated incorrectly. This incorrect calculation caused the file's size to be truncated and compromised data integrity.
- The backup process looped endlessly. This infinite loop usually occurred on large files with multiple header extents.

These problems have been corrected; block count is now calculated correctly and the infinite loops no longer occur.

3.6.7 Postprocessing Backup Problem Corrected

V5.3-2

In VMS Versions 5.3 and 5.3-1, potential problems existed when you used BACKUP with the /DELETE and /RECORD qualifiers.

When you use the /DELETE qualifier, BACKUP creates an output file and then deletes the input file. However, in Versions 5.3 and 5.3-1, if BACKUP encountered an error while it created the output file, BACKUP incorrectly deleted the input file. This problem occurred during disk-to-disk image backups or backup copy operations. In the case of the /RECORD qualifier, when BACKUP was unable to open a file for input, BACKUP incorrectly recorded the date and time in the file's header record.

These problems have been corrected for VMS Version 5.3-2; the /DELETE and /RECORD qualifiers now perform as documented.

3.6.8 Reading Multiple Save Sets from a TU81-PLUS Tape Drive

V5.2

Under some circumstances, when reading multiple backup save sets from a TU81-PLUS tape drive, the drive can become misaligned and return a "position lost" status, thus aborting the backup. This can occur while attempting to read a save set after having just read a preceding save set on the same volume. This is a timing problem that will be addressed in a future fix for the TU81-PLUS controller microcode. In the meantime, you can use the following temporary solutions:

- Before entering your BACKUP command, reposition the tape with the following DCL command:

```
$ SET MAGTAPE/SKIP=RECORDS:-1 tape_drive
```
- Use \$BACKUP/REWIND to read the second or subsequent save set on the tape. Note that this rewinds the tape to the beginning (BOT) and then searches forward for the specified save set.

3.6.9 Standalone Backup — Saving a Bound Volume Set Not Supported

V5.4

In Standalone Backup for VMS Version 5.4, the saving of a bound volume set is not supported. This operation will be supported in a future release of VMS. Digital recommends that you use online BACKUP as an alternative to saving bound volume sets.

3.6.10 Using BACKUP with Compound Document Files

V5.1

Normal use of BACKUP correctly preserves all file attribute information for compound document (for example, DDIF) files. However, the BACKUP/INTERCHANGE command fails to preserve the semantics attribute. As a workaround, DDIF files restored from BACKUP/INTERCHANGE save sets can be relabeled as DDIF files using the following command line:

```
$ SET FILE/SEMANTICS=DDIF file-spec[...]
```

3.7 Batch/Print Facility Notes

The release notes in this section pertain to the VMS Batch/Print Facility.

3.7.1 Generic Queue Restriction Now Enforced

V5.0-2

The VMS Batch/Print Facility is designed to allow one execution queue to be the target of no more than eight generic queues. In Versions 5.0 and 5.0-1 of the VMS operating system, this restriction was not enforced, resulting in unpredictable behavior if more than eight generic queues directed jobs to one execution queue.

Beginning with Version 5.0-2 of the VMS operating system, this restriction is enforced.

3.7.2 Print Symbiont Working Set Purge Less Frequent

V5.2

In previous versions of VMS, when the print symbiont finished processing a file, it purged its working set if no other queue it supported contained a job that was printing. For example, the command PRINT FILE.TEXT /COPIES=5 caused a print symbiont supporting only one queue to purge its working set five times.

With VMS Version 5.2, the print symbiont process purges its working set only if none of the queues supported by that symbiont contains a non-pending job within a purge-delay time interval following completion of any file it processes. The purge-delay time interval is currently defined as approximately 5 minutes.

System Manager Release Notes

3.7 Batch/Print Facility Notes

3.7.3 **START/QUEUE/MANAGER/BUFFER_COUNT Command Enhancement**

V5.2

With VMS Version 5.2, the /BUFFER_COUNT qualifier of the START/QUEUE/MANAGER command has been enhanced to accept a value greater than the previous limit of 127 local buffers. Increased use of local buffers for the queue file by each job controller in a cluster may improve overall batch/print performance by reducing the disk I/O at the expense of memory consumption, locking activity, and CPU cycles. This can be important for a system that supports many queues and whose workload is batch or print job intensive.

The enqueue limit and working set quotas for the job controller process can now support up to 1500 local buffers, provided that memory is available for the system to extend the job controller's working set size dynamically. If the job controller's working set cannot be extended sufficiently, excessive page faulting will occur. If free memory is limited, the local buffer value should not exceed 300.

V5.4

The enqueue limit and working set quotas for the job controller process have been increased for VMS Version 5.4. This increase enables the job controller to support up to 5000 local buffers for the queue file, provided that memory is available for the system to extend the job controller's working set size. If the job controller's working set cannot be extended sufficiently, excessive page faulting will occur. If free memory is limited, the local buffer value should not exceed 800.

Beginning with VMS Version 5.4, the job controller provides automatic adjustment of the size of the local buffer cache if the /BUFFER_COUNT qualifier on the START/QUEUE/MANAGER command is not specified, or the buffer count value is 0. By default, 100 local buffers are allocated when the queue file is opened. Each job controller, however, can dynamically increase the size of its local buffer cache based on the number of queues defined along with other factors. The algorithm is conservative; therefore, you may need to set the local buffer cache size manually for optimal performance.

For a complete description of the /BUFFER_COUNT qualifier, see the *VMS DCL Dictionary*.

3.7.4 **START/QUEUE/TOP_OF_FILE Command — Line-Feed Improvement**

V5.2

In previous versions of VMS, an unwanted line feed was printed at the top of the first page of output following a START/QUEUE/TOP_OF_FILE command. This problem has now been corrected.

3.7.5 **Unsynchronized Cluster Time Affects SUBMIT/AFTER Command**

V5.0

In a VAXcluster environment, a batch job submitted to execute at a specified time may begin execution a little before or after the requested time. This occurs when the clocks of the member systems in the VAXcluster environment are not synchronized. For example, a job submitted using the DCL command SUBMIT/AFTER=TOMORROW may execute at 23:58 relative to the host system's clock.

System Manager Release Notes

3.7 Batch/Print Facility Notes

This problem can occur in a cluster even if a job is run on the same machine from which it was submitted. The redundancy built into the batch/print system allows more than one job controller in the cluster to receive a timer asynchronous system trap (AST) for the job and, thus, to schedule it for execution. Moreover, this behavior is exacerbated if the batch job immediately resubmits itself to run the next day using the same SUBMIT command. This can result in multiple instances of the job executing simultaneously because TOMORROW (after midnight) may be only a minute or two in the future.

A solution to this problem is to place the SUBMIT command in a command procedure that begins with a WAIT command; the delta time specified in the WAIT command should be greater than the maximum difference in time between any two systems in the cluster. Use the SHOW TIME command on each system to determine this difference in time.

The cluster time can be kept in synchronization by periodic execution of the DCL command SET TIME/CLUSTER. This recalibrates the individual system times.

3.8 Bootstrapping — Special Files Handling

V5.0

With VMS Version 5.0, a set of special files is opened during the system bootstrap to prevent files from being accidentally deleted or being accidentally shared among member nodes in a cluster. The special files include the page file, the swap file, the dump file, the cluster incarnation data file, and the executive loaded image file.

The implications of this change are as follows:

- The following SET FILE commands require exclusive access to a file. Because the page file, swap file, dump file, and executive loaded images are open, the following SET FILE commands fail and issue the error message "ACCONFLICT, file access conflict" when you attempt to modify any of the files.

```
SET FILE/[NO]BACKUP
SET FILE/DATACHECK
SET FILE/END_OF_FILE
SET FILE/ERASE
SET FILE/[NO]EXPIRATION_DATE
SET FILE/EXTENSION
SET FILE/GLOBAL_BUFFER
SET FILE/OWNER
SET FILE/TRUNCATE
SET FILE/VERSION_LIMIT
```

- With proper privileges, it is possible to mark a special file for deletion. That is, when the DCL command DELETE is entered to delete a special file, that file is removed from the directory and marked for deletion. However, the file body is not deleted. The only way to reclaim the disk blocks occupied by the special file is to reboot the system and enter the command ANALYZE/DISK_STRUCTURE/REPAIR on the system disk.

System Manager Release Notes

3.8 Bootstrapping — Special Files Handling

- The primary page file (PAGEFILE.SYS), the primary swap file (SWAPFILE.SYS), the dump file (SYSDUMP.DMP), and the cluster incarnation data file (SYS\$INCARNATION.DAT) must reside in the system-specific rooted directory, SYS\$SPECIFIC:[SYSEXE]. If any of these files reside in the common rooted directory, SYS\$COMMON:[SYSEXE], the system bootstrap will not be able to find or use the files.

If you use shared dump files, you may be affected by this change. A **shared dump file** is a system dump file that is used by two or more nodes in the VAXcluster environment. To allow dump files to be shared among nodes in a VAXcluster environment, perform the following steps:

- 1 Create the shared dump file SYS\$COMMON:[SYSEXE]SYSDUMP-COMMON.DMP by entering the following command lines, where *n* is the size of the system dump file:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> CREATE SYS$COMMON:[SYSEXE]SYSDUMP.DMP/SIZE=n
SYSGEN> EXIT
```

- 2 For each VAXcluster node sharing the file, enter the following DCL command line (*n* is the system-specific root for the node):

```
$ SET FILE SYS$COMMON:[SYSEXE]SYSDUMP-COMMON.DMP -
_ $ /ENTER=SYS$SYSDEVICE:[SYSn.SYSEXE]SYSDUMP.DMP
```

Note: You can use the file name SYSDUMP.DMP for a shared dump file. However, when you upgrade your cluster, you must change the name. The VMS upgrade procedure does not allow a shared dump file to be named SYSDUMP.DMP.

3.9 CLUSTER_CONFIG.COM Changes

The release notes in this section describe changes to the cluster configuration command procedure CLUSTER_CONFIG.COM.

3.9.1 CLUSTER_CONFIG.COM Now Requires VOLPRO Privilege

V5.4

Executing the command procedure (CLUSTER_CONFIG.COM) now requires the VOLPRO privilege in addition to the SYSPRV, OPER, CMKRNL, BYPASS and NETPRV privileges. If you run CLUSTER_CONFIG.COM from an account that does not have these privileges, an error message such as the following is displayed:

```
Insufficient privileges. SYSPRV, OPER, CMKRNL, BYPASS, NETMBX, VOLPRO required.
To ensure that you have the required privileges, please execute
this procedure from the system manager's account.
```

To make sure the required privileges are available, you should run CLUSTER_CONFIG from the system manager's account (SYSTEM).

3.9.2 PAGEFILE=0 and SWAPFILE=0 No Longer Written to MODPARAMS.DAT

V5.4

Prior to VMS Version 5.4, CLUSTER_CONFIG.COM wrote lines containing PAGEFILE=0 and SWAPFILE=0 to the MODPARAMS.DAT file of every newly added satellite node. The values prevented AUTOGEN from changing the size of the satellite node's page and swap files.

The PAGEFILE=0 and SWAPFILE=0 lines are no longer included in the MODPARAMS.DAT file. Therefore, AUTOGEN can change the size of the satellite node's page and swap files over time, depending on the amount of activity on the node. To prevent AUTOGEN from changing the size of a satellite node's page and swap files, add PAGEFILE=0 and SWAPFILE=0 to the MODPARAMS.DAT file manually.

For more information on AUTOGEN operations, see the *Guide to Setting Up a VMS System*.

3.10 Cluster I/O — Subsystem Changes

The release notes in this section pertain to the VAXcluster disk-I/O subsystem.

3.10.1 Disk Failover

V5.2

With VMS Version 5.2, the disk class drivers trigger failover in response to certain attention messages generated by DIGITAL Storage Architecture (DSA) controllers. For most configurations, this will not result in any operational changes other than a faster recovery from some classes of error.

This change eliminates the requirement that disks that are dual pathed between local controllers be mounted on both nodes.

3.10.2 MSCP Serving Third-Party Disks

V5.2

Problems that prevented the mass storage control protocol (MSCP) from serving third-party disks to other members of a VAXcluster configuration have been corrected in VMS Version 5.2.

For the disk to be served correctly, its device driver must use a device type in the range DT\$_FD1 through DT\$_FD7, the device class must be DC\$_DISK, and the device driver must correctly initialize the parameter UCB\$_MEDIA_ID.

The parameter UCB\$_MEDIA_ID contains a bit-encoded media identification that is used by the class drivers to form a local device name for a disk served by a remote VAXcluster member. The parameter consists of five 5-bit fields and one 7-bit field. The fields are defined as follows:

System Manager Release Notes

3.10 Cluster I/O — Subsystem Changes

31	26	21	16	11	6	0
+-----+-----+-----+-----+-----+-----+						
D0	D1	A0	A1	A2	N	
+-----+-----+-----+-----+-----+-----+						

Fields D0 and D1 contain the device-type name and are the only fields required by the disk class drivers to form a device name. They are encoded so that A is a 1, B is a 2, and so forth. The remaining fields are optionally used to contain a media name, such as ZZ01 with ZZ encoded in the field A0-A2 in the same fashion as D0 and D1, and the 01 encoded in N as a binary number.

The parameter `UCB$L_MEDIA_ID` can be initialized by `DPT_STORE` macros or by driver initialization routines. Refer to the *VMS Device Support Manual* for more details.

Note that this device name *must* match the local device name provided to `SYSGEN` on the serving node. For example, if the disk name is ZBA0, `UCB$L_MEDIA_ID` should contain `%XD0800000`. The `SYSGEN` command to connect the device on the serving node is:

```
SYSGEN> CONNECT ZBA/DRIVER=[dir]filename/UNITS=numunits
```

To serve this device to the other VAXcluster members, use the following DCL command:

```
$ SET DEVICE ZBA0:/SERVED
```

Note that the MSCP server must have been loaded by setting the `SYSGEN` parameter `MSCP_LOAD` (refer to the *VMS VAXcluster Manual*).

If `UCB$L_MEDIA_ID` is not correctly initialized, no error message is generated. If `UCB$L_MEDIA_ID` is blank, the disk is not visible to the other VAXcluster members. If `UCB$L_MEDIA_ID` contains a different name from that used locally, the disk appears to the other VAXcluster members as a different device, which may result in uncoordinated access to the disk by multiple members.

3.11 Debugger Notes

The release notes in this section pertain to the VMS debugger.

3.11.1 System Management Considerations

V5.2

In previous versions of VMS, the debugger and the program being debugged ran in the same process.

Starting with VMS Version 5.2, the debugger consists of two parts (main and kernel) to accommodate the debugging of multiprocess programs.

The following changes affect system management:

- For a program that runs in one process, a debugging session now requires two processes instead of one.
- For a multiprocess program, a debugging session requires as many processes as are used by the program, plus an additional process for the main debugger.

Under these conditions, multiple users who are simultaneously debugging programs can place an additional load on a system. This section describes the resources used by the debugger, so that you can tune your system for this activity.

Note that the following discussion covers only the resources used by the debugger. In the case of multiprocess programs, you may also need to tune the system to support the programs themselves.

3.11.2 System Resources

V5.2

The kernel and main debugger communicate through global sections. The main debugger communicates with up to eight kernel debuggers through a 65-page global section. Therefore, you may need to increase the SYSGEN global-page and global-section parameters (GBLPAGES and GBLSECTIONS, respectively). For example, if 10 users are using the debugger simultaneously, 10 global sections using a total of 650 global pages are required by the debugger.

3.11.3 User Quotas

V5.2

Each user needs a PRCLM quota sufficient to create an additional subprocess for the debugger, beyond the number of processes needed by the program.

BYTLM, ENQLM, FILLM, and PGFLQUOTA are pooled quotas. You may need to increase these quotas to account for the debugger subprocess as follows:

- You should increase each user's ENQLM quota by at least the number of processes being debugged.
- You may need to increase each user's PGFLQUOTA. If a user has an insufficient PGFLQUOTA, the debugger may fail to activate or may cause "virtual memory exceeded" errors during execution.
- You may need to increase each user's BYTLM and FILLM quotas. The debugger requires sufficient BYTLM and FILLM quotas to open each image file being debugged, the corresponding source files, and the debugger input, output, and log files. The debugger command SET MAX_SOURCE_FILES can be used to limit the number of source files kept open by the debugger at any one time.

3.12 DECdtm Services Notes

VMS Version 5.4 adds support for DECdtm services. The DECdtm services provide base operating system support for distributed transactions and support the two-phase commit protocol.

The release notes in this section pertain to DECdtm services.

System Manager Release Notes

3.12 DECdtm Services Notes

3.12.1 Adjusting SYSGEN Parameters Before Using DECdtm Services

V5.4

If you intend to use the DECdtm services on your system, you should plan for extra memory to be allocated for transaction log buffering.

One method of allocating extra memory is to run the AUTOGEN command procedure with the FEEDBACK option. AUTOGEN collects feedback data on the running system and calculates appropriate new values for SYSGEN parameters.

Another way to accommodate transaction log buffering is to increase the value of the SYSGEN parameter MIN_NPAGEVIR by 12,000 bytes. Digital recommends that you edit SYS\$SYSTEM:MODPARAMS.DAT and change the relevant line that sets the MIN_NPAGEVIR parameter.

3.12.2 DECdtm Services — Startup Processes and How to Inhibit Them

V5.4

DECdtm services start up only after a full boot of VMS Version 5.4. DECdtm services do not start up when you perform an upgrade or if you specify a minimum boot using the command STARTUP_P1=MIN.

DECdtm services start two processes: IPCACP and TP_SERVER. To prevent IPCACP and TP_SERVER from starting up as part of a full boot, you must define the systemwide logical name SYS\$DECDTM_INHIBIT in the SYS\$STARTUP:SYLOGICALS.COM command procedure. You can define SYS\$DECDTM_INHIBIT to be any equivalence string.

3.12.3 Defining Logical Names Before Starting DECdtm Services

V5.4

If you usually define one or both of the system logical names NETNODE_LOCAL and NETNODE_REMOTE, make sure you make the assignments within SYS\$MANAGER:SYLOGICALS.COM so that the assignments are in effect before DECdtm services are started. If the logical names are not defined, the DECdtm startup process could fail.

3.12.4 File Name Considerations for Transaction Log Files

V5.4

Digital recommends that transaction log files be named SYSTEM\$*node-name*.LM\$JOURNAL), where *node-name* is the name of the node that uses the log file. Because the transaction log file referenced is related to the node name, you could encounter problems if you give a new node name to your system.

In a VAXcluster environment, this situation does not cause any problems because a mechanism called VAXcluster failover ensures that references to log file names are resolved, even if the log file's node has been renamed.

For example, consider a VAXcluster configuration containing nodes RED and BLUE, on which a transaction application is distributed. The transaction log file on node BLUE is named SYSTEM\$BLUE.LM\$JOURNAL. This log file contains records of transaction state transitions. Participating transaction managers and resource managers on other nodes might need to access

SYSTEM\$BLUE.LM\$JOURNAL to resolve transactions. The system manager of BLUE decides to rename the node name BLUE to GREEN. However, the old log file, SYSTEM\$BLUE.LM\$JOURNAL, still exists on the renamed system.

If node RED needs to resolve transactions, it would try to access node BLUE. However, since BLUE had been renamed, RED would not find BLUE and would assume that BLUE had failed. Using the VAXcluster failover mechanism, RED can still access the old log file, SYSTEM\$BLUE.LM\$JOURNAL, and can continue the process of resolving transactions.

This failover mechanism ensures transaction resolution on VAXcluster nodes. However, in a wide area network, transactions remain unresolved if one node tries to reference the transaction log file on a remote node that did not match the name of the remote node. To resolve this problem, for each node that is likely to call the remote node, you must change the network node database to point from the location of the old node name to location of the new node name.

For more information about VAXcluster failover, refer to the the LMCP section in the *VMS Version 5.4 New Features Manual*.

3.12.5 File Size Considerations for Transaction Log Files

V5.4

The size of a log file is an important factor in the performance of distributed transactions. For recommendations about determining the initial size of a log file and where to place it, refer to the the LMCP section in the *VMS Version 5.4 New Features Manual*.

3.12.6 Image Files

V5.4

Table 3-1 lists the image files included with DECdtm services.

Table 3-1 DECdtm Images

Image Name	Image
SYS\$LOADABLE_IMAGES: SYS\$TRANSACTION_SERVICES.EXE	Loadable executive image
SYS\$SYSTEM:IPCACP.EXE	IPC image
SYS\$SYSTEM:TPSERV.EXE	Log and recovery server
SYS\$SYSTEM:LMCP.EXE	Log Manager Control Program (LMCP)
SYS\$HELP:LMCP\$HLB.HLB	LMCP help library
SYS\$MESSAGE:LMCP\$MESSG.EXE	LMCP message file

(continued on next page)

System Manager Release Notes

3.12 DECdtm Services Notes

Table 3-1 (Cont.) DECdtm Images

Image Name	Image
SYS\$LIBRARY:DTI\$SHARE.EXE	Privileged shareable image
SYS\$STARTUP:DECDTM\$STARTUP.COM	DECdtm startup command procedure

3.13 DECnet-VAX Notes

The release notes in this section pertain to DECnet-VAX software.

For additional DECnet-VAX information, see the following:

- DIGITAL Distributed Queuing Service (DECdqs) — Creating the Account DQS\$SERVER (Section 3.15)
- Ethernet Notes (Section 3.23)
- NETCONFIG.COM Security Enhancements (Section 3.36.2)
- Security Review Recommended (Section 3.37)
- SET HOST (CTDRIVER/REMACP/RTPAD) Notes (Section 3.38)
- DECnet — File Access Protocol Extensions (Section 4.9)

3.13.1 Constraints on Passive Maintenance Functions Relaxed

V5.0

In previous versions of VMS, the passive functions of upline dump and downline load (without software ID) were performed only when the node was present in the volatile node database and the SERVICE CIRCUIT parameter matched the circuit over which the function was requested. With VMS Version 5.0, the SERVICE CIRCUIT parameter is ignored for these functions.

3.13.2 DECnet-VAX Objects — New Characteristic Added

V5.3-2

With VMS Version 5.3-2, a new characteristic has been added to DECnet-VAX objects. The new characteristic, **outgoing connect privileges**, is used to specify the privileges a user must possess to make outbound connections to an object. You can use this new characteristic to prevent unauthorized access to objects.

Use the new Network Control Program (NCP) command SET/DEFINE OBJECT OUTGOING CONNECT PRIVILEGES to set the characteristic. The privileges are specified in the same format as in the SET/DEFINE OBJECT PRIVILEGES command. The *VMS Network Control Program Manual* contains the format and parameters used with the SET/DEFINE OBJECT PRIVILEGES command.

Example 3-1 illustrates the use of the NCP command SET OBJECT OUTGOING CONNECT PRIVILEGES. At the DCL command prompt, the user sets privileges for the process. After invoking NCP, the user specifies the privileges necessary to connect to the object TEST. Because the object TEST requires the READALL privilege to make an outbound connection, the connection fails, resulting in an error message. After the user resets the privileges to include READALL, the process is able to connect to the object.

Example 3-1 Using the SET OBJECT OUTGOING CONNECT PRIVILEGES Command

```
$ SET PROCESS/PRIVILEGES=(NOALL,TMPMBX,NETMBX,OPER)
$ RUN SYS$SYSTEM:NCP
NCP> SET OBJECT TEST NUMBER 0 -
OUTGOING CONNECT PRIVILEGES READALL OPER
NCP> SET OBJECT TEST FILE SMITH$DISK:[SMITH]TEST.COM
NCP> EXIT
$ OPEN/READ LINK 0"SMITH CEADGUTY"::"0=TEST"
%DCL-E-OPENOUT, error opening 0"smith password"::"0=test" as
output -RMS-E-PRV, insufficient privilege or file protection
violation

$ SET PROCESS/PRIVILEGES=(NOALL,TMPMBX,NETMBX,OPER,READALL)
$ OPEN/READ LINK 0"SMITH CEADGUTY"::"0=TEST"
$ READ LINK RECORD
```

3.13.3 Downline Loading Correction

V5.1-1

VMS Version 5.1-1 incorporates a correction that prevents problems in downline loading system images resulting from improper assignment of the MOM\$LOAD logical name. The problem occurred when loadable images were placed in directories other than SYS\$SYSROOT:[MOM\$SYSTEM]. The corrected STARTNET.COM checks the system logical name table for the logical name MOM\$LOAD and does not assign that logical name if it has been previously defined.

3.13.4 Executor Parameters

The following sections describe changes in DECnet executor parameters.

3.13.4.1 Executor Path Split Policy — INTERIM Policy Corrected

V5.3-2

Path split policy specifies the policy for equal cost splitting of network traffic. The policy is set with the NCP command, SET/DEFINE EXECUTOR PATH SPLIT POLICY. You can set the policy to either INTERIM or NORMAL. The INTERIM policy specifies that all packets for a specific logical link connection travel over the same path for the duration of the connection. The NORMAL policy specifies that all traffic is split equally over all equal cost paths to a destination node.

Prior to VMS Version 5.3-2, a problem existed with the INTERIM path split policy. This problem has been corrected in VMS Version 5.3-2.

For more information about the SET/DEFINE EXECUTOR command, see the *VMS Network Control Program Manual*.

System Manager Release Notes

3.13 DECnet-VAX Notes

-
- V5.2** **3.13.4.2 MAXIMUM LINKS — Maximum Value Increased**
The maximum value for the EXECUTOR parameter MAXIMUM LINKS has been increased to 3885. Also, the value for MAXIMUM LINKS is no longer reduced by the value set for the EXECUTOR parameter ALIAS MAXIMUM LINKS.
-
- V5.0** **3.13.4.3 MAXIMUM PATH SPLITS Default Value**
The EXECUTOR parameter MAXIMUM PATH SPLITS default value is 1. As a result, DECnet-VAX does not path split over equal cost paths by default.
-
- V5.2** **3.13.4.4 PIPELINE QUOTA — Default Value Changed**
Prior to VMS Version 5.2, the DECnet EXECUTOR parameter PIPELINE QUOTA had a default value of 3000 bytes. Increasing this value to 10,000 can result in a significant improvement in DECnet-VAX performance.

With VMS Version 5.2, the default value for PIPELINE QUOTA has been increased to 10,000. This is a change in the default value only; systems that have PIPELINE QUOTA explicitly set in the DECnet database are not affected, even if the parameter's value is less than 10,000.

3.13.5 Local DTE Address Used on Outgoing Call

- V5.3-2** Prior to VMS Version 5.3-2, a problem existed with using the local data terminal equipment (DTE) address on an outgoing data link mapping (DLM) circuit for switched virtual circuits (SVC).

This problem has been corrected with Version 5.3-2. The DTE characteristic is now passed to the X.25 Packetnet Switch Interface (PSI). PSI then uses this address as the local DTE to send the outgoing call. When an outgoing DLM circuit is turned on, the call is sent over the physical line associated with the DTE whose address is specified in the DLM circuit characteristics.

For more information about the SET/DEFINE CIRCUIT DTE command, see the *VMS Network Control Program Manual*.

3.13.6 Maintenance Operations Module — Repeated Operations Error

- V5.3-2** According to the specification for serial communications upon which Digital's wide area communications devices are based (DEC STD 052), device shutdown is allowed to take up to 5 seconds.

A problem has been observed when certain maintenance operations, performed by the maintenance operations module (MOM), are repeated immediately on some synchronous devices, such as the DSV11. It is possible that the second operation can execute before the shutdown sequence for the first operation has completed. The result is a spurious fatal device error. Example 3-2 illustrates this problem.

Example 3-2 Example of a Repeated Operations Error

```
$ NCP LOOP CIRCUIT DSV-0-0
$ NCP LOOP CIRCUIT DSV-0-0
%NCP-W-LINCOM, Line communication error
Messages not looped = 1
%SYSTEM-F-DEVINACT, device inactive
```

Digital recommends that you use the COUNT parameter to specify the number of times a maintenance operation is to be performed rather than repeat operations immediately. Example 3-3 illustrates using the COUNT parameter.

Example 3-3 Using Count Parameter in a Maintenance Operation

```
$ NCP LOOP CIRCUIT DSV-0-0 COUNT 100
```

For more information about maintenance operations, refer to the *VMS Networking Manual*.

3.13.7 NCP/NML Requires OPER Privilege to Obtain Service Passwords

V5.2-1

In previous versions of VMS, no privileges were required to obtain service passwords from the permanent and volatile network databases. OPER privilege is now required to obtain service passwords.

3.13.8 NETACP\$BUFFER_LIMIT Logical Name — To Override Default BYTLM Quota

V5.2-1

You can use the NETACP\$BUFFER_LIMIT logical name to override the default BYTLM quota given to the network ancillary control process (NETACP). Prior to invoking SYS\$MANAGER:STARTNET.COM, you should define NETACP\$BUFFER_LIMIT in SYSTARTUP_V5.COM.

3.13.9 Network Control Program (NCP) — SET/DEFINE CIRCUIT Command

V5.2

In VMS Version 5.0, the COST parameter for the SET/DEFINE CIRCUIT command of the Network Control Program (NCP) accepted decimal values in the range 1 to 25. In VMS Version 5.2, the maximum value has been increased from 25 to 63 to accommodate an increased diversity in DECnet routing specifications. The default value in VMS Version 5.2 is still 10.

3.13.10 NML Checks for Illegal Address Configurations

V5.2-1

The network management listener (NML) object has been changed to check that the executor node and alias node addresses do not exceed the value of the MAXIMUM ADDRESS executor node parameter. A check is also made to ensure that the executor and alias nodes do not have the same address. These checks help prevent illegal address configurations from being defined in the permanent network database.

System Manager Release Notes

3.13 DECnet-VAX Notes

3.13.11 Node-Level Access Control Problem Corrected

V5.4

In previous versions of VMS, if you set default access to NONE on the executor node and then set access to OUTGOING or BOTH for a specific node, outgoing connections from the executor node to that specific node failed and the system displayed the following error message:

```
%SYSTEM-F-SHUT, remote node no longer accepting connects
```

VMS Version 5.4 corrects this error in node-level access control for outgoing connections.

3.13.12 Proxy Access Parameters — Changes

V5.0

The proxy access parameters used by the executive node to determine what kind of access is allowed have changed. By default, incoming and outgoing access are both enabled. The NCP commands to modify the parameters are as follows:

```
NCP> DEFINE EXEC INCOMING PROXY ENABLED
NCP> DEFINE EXEC INCOMING PROXY DISABLED
NCP> DEFINE EXEC OUTGOING PROXY ENABLED
NCP> DEFINE EXEC OUTGOING PROXY DISABLED
```

The proxy database is now built into the network ancillary control process (NETACP) as a volatile database at network startup time. Use the following NCP command to accomplish this:

```
NCP> SET KNOWN PROXIES ALL
```

The STARTNET.COM command procedure supplied by Digital has been modified to issue this command to build the volatile proxy database. If a private network startup procedure is used, and proxy access is desired, then the appropriate commands should be added to load the volatile database.

Changes made to NETPROXY.DAT using the Authorize Utility after the volatile proxy database has been created are also automatically changed in the volatile database.

3.13.13 Session Incompatibility with Phase IV Implementations

V5.0-1

Incompatibilities have been found between the Phase IV and Phase IV+ Session Control architecture. Since DECnet-VAX Version 5.0 implements Phase IV+ of the DIGITAL Network Architecture (DNA), it is affected by the incompatibilities. One of the following problems can occur when you attempt a connection from a Phase IV+ node to a Phase IV node:

- The Phase IV Session Control architecture defines the invokeProxy bit in the connect initiate message as being reserved. It also states that any bit defined as reserved must be set to zero unless otherwise specified. Some Phase IV implementations expect the invokeProxy bit to be zero and will reject the connection with a protocol error if it is nonzero. Others do not check this field because it is not used in Phase IV. Because proxy is part of the Phase IV+ design, the invokeProxy bit is now nonzero. This causes connections initiated from Phase IV+

System Manager Release Notes

3.13 DECnet-VAX Notes

implementations to be rejected by Phase IV implementations with a protocol error.

- Phase IV implementations expect the session version number in the connect initiate message to be 0, because this is the value for Session V1.0. The session version is now 1 to reflect Session V2.0. The connection will be rejected because of version skew.

Table 3-2 lists the DECnet implementations and the version/update required to resolve the compatibility problems.

Table 3-2 Patches for DECnet Compatibility Problems

DECnet Implementation	Version/Update That Contains the Patch
DECnet-10	Corrected in Autopatch Tape 14. ¹
DECnet-20	Corrected in Autopatch Tape 14. ¹
DECnet-VAX	All versions work correctly.
DECnet/E	Corrected in Version 2.1.
PRO/DECnet	Corrected in Version 2.1, patches also available in POS V3.1. ¹
DECnet/RT	Patch supplied in RT-11 update January, 1987. ¹
DECnet-IAS	No plans to correct the problem.
DECnet-DOS	All versions work correctly.
DECnet/RSX	Patch supplied with M+ Update C or later. ¹ Patch supplied with M/S Update C or later. ¹
DECnet/MicroRSX	Corrected in Version 1.1.
DECnet-ULTRIX	All versions work correctly.
DECnet Router	Corrected in Version 1.2.

¹Installation of these patches was optional. Some customers may have elected not to install the patches provided and may be running the correct version without the patches.

3.13.14 Support for X.25 Virtual Circuits Requirement

V5.0

For DECnet-VAX to support 128 X.25 virtual circuits for data link mapping, you should change the parameter /FILE_LIMIT in the file SYS\$MANAGER:LOADNET.COM from 10 to 128.

3.14 DECwindows Notes

The release notes in this section pertain to the DECwindows interface.

System Manager Release Notes

3.14 DECwindows Notes

3.14.1 DECwindows Startup

V5.2

The following notes pertain to starting DECwindows software:

- If the DECwindows startup command procedure `DECW$STARTUP.COM` determines that it is necessary to run the command procedure `AUTOGEN.COM`, it is now run with feedback if valid feedback data exists.
- If your system does not have valid VMS or VAXcluster licenses installed, appropriate messages are displayed on your console terminal and DECwindows is not automatically started. If this occurs, log in to the console terminal, install the valid licenses, and reboot the system.

3.14.2 DECwindows Startup for Upgrade Only

V5.2

In VMS Version 5.1, system managers were instructed to execute the DECwindows startup file, `DECW$STARTUP.COM`, in the site-specific startup file, `SYSTARTUP_V5.COM`. With VMS Version 5.2, the command procedure `DECW$STARTUP.COM` is performed as part of VMS startup after `SYSTARTUP_V5.COM` has completed. If you are upgrading from a Version 5.1, Version 5.1-1 or Version 5.1-B system, you need to remove the `@SYS$MANAGER:DECW$STARTUP` command from your `SYSTARTUP_V5.COM` file.

If you did not place a call to `DECW$STARTUP.COM` in your system startup file, you need to take no additional action after upgrading.

If you invoked `DECW$STARTUP.COM` outside of `SYSTARTUP_V5.COM`, do one of the following:

- Remove the call to `DECW$STARTUP.COM` that you inserted in the VMS Version 5.1, Version 5.1-1, or Version 5.1-B system.
- Signal VMS not to execute `DECW$STARTUP.COM` by defining the logical name `DECW$IGNORE_DECWINDOWS` in `SYSTARTUP_V5.COM` as follows:

```
$ DEFINE DECW$IGNORE_DECWINDOWS TRUE      ! Delay DECwindows startup
```

Note: You should define this logical name only if your site-specific startup is going to invoke `DECW$STARTUP.COM` at a later time.

3.14.3 Tailoring DECwindows

V5.2

If you have a small system disk (RD53) and you tailor off the DECwindows files, you may find that you end up with less free space than is indicated by the tailoring-off process. This problem is most likely related to `AUTOGEN` creating larger page, swap, and dump files. `AUTOGEN` is run when you tailor off device support.

Before tailoring *on*, please check that you have proper free space. Tailoring does not check that there is sufficient space for the selected files.

3.14.4 **Template File — New DECW\$SYLOGIN.TEMPLATE**

V5.3

The file DECW\$SYLOGIN.COM is no longer shipped with the DECwindows kit. Instead, the file DECW\$SYLOGIN.TEMPLATE is shipped. If you are upgrading from a previous version of DECwindows, you must delete DECW\$SYLOGIN.COM from the SYS\$MANAGER directory if you have not made modifications to the file. Login files in the current DECwindows kit do not reference DECW\$SYLOGIN.COM.

3.14.5 **Template File — Support for Configuring Multihead Systems**

V5.3-1

VMS Version 5.3-1 and subsequent versions have support for workstations with multiple graphics controllers and monitors that are controlled from one keyboard and one pointing device.

Information and procedures to configure the software for multihead systems are included in the file DECW\$PRIVATE_SERVER_SETUP.TEMPLATE in the SYS\$MANAGER directory.

3.14.6 **ULTRIX Connection (UCX) and TCP/IP Transport**

The following sections contain release notes about the ULTRIX Connection (UCX) layered product and the DECwindows TCP/IP transport.

3.14.6.1 **Sequence Lost Errors**

V5.3

Under a heavy interactive load, applications using the DECwindows TCP/IP transport may exhibit "sequence lost" error messages. DECW\$PAINT is a typical application that can exhibit this behavior. This behavior is caused by a problem in UCX Version 1.0, and there is no way to work around the problem.

Version 1.2 of UCX corrects the problem.

3.14.6.2 **Starting UCX Before DECnet — Problem**

V5.3

The UCX startup command file must be executed only after the DECnet startup command file has completed. If UCX is started before DECnet, DECnet does not work properly.

In the file SYS\$MANAGER:SYSTARTUP_V5.TEMPLATE, there are two possible commands that can be used to start DECnet. One command submits a batch job to start DECnet, and the other starts DECnet immediately. The simplest solution is to select the command that starts DECnet immediately, and then to place the command @SYS\$MANAGER:UCX\$STARTUP somewhere later in the SYSTARTUP_V5.COM command file.

If you want to start DECnet from a batch job, you should submit a batch job that first starts DECnet and then starts UCX. For example, you could create the file STARTNETUCX.COM, which should contain the following:

```
IF F$SEARCH("SYS$SYSTEM:NETACP.EXE") .NES. "" -
THEN @SYS$MANAGER:STARTNET
$ @SYS$MANAGER:UCX$STARTUP
```

System Manager Release Notes

3.14 DECwindows Notes

You could then add the following line to SYSTARTUP_V5.COM:

```
$ SUBMIT SYS$MANAGER:STARTNETUCX.COM
```

3.14.7 X Servers — Interoperability with Other Vendors' X Servers

V5.4

Many Digital applications experience problems when they are used with other vendors' X servers. These problems usually occur when the application needs a font that the server does not have. You can prevent these problems by directing the server to use alternate fonts, which is accomplished on the X server's node through a font alias file.

There is an example font alias file that you can use with any MIT-based X server (nearly all X servers from other vendors are MIT-based) called DECW\$EXAMPLES:FONTS.ALIAS. Copy the fonts.alias file to one of the font directories that the X server uses. On UNIX systems, this font directory is usually /usr/lib/X11/fonts/75dpi or /usr/lib/X11/fonts/100dpi. If a fonts.alias file already exists, you should combine the old and the new files to make a single fonts.alias file.

Note: UNIX file names are case sensitive. For file names such as fonts.alias, use lowercase letters only.

The fonts.alias file works by mapping fonts used by Digital's applications to fonts supplied on the MIT X11 R4 tape. Therefore, the following font families must already be installed on the server's node in order for the fonts.alias file to work:

- Courier
- Helvetica
- New Century Schoolbook
- Symbol
- Terminal
- Times

In addition, the cursor, DECW\$CURSOR, DECW\$SESSION, and fixed fonts must be installed.

3.15 DIGITAL Distributed Queuing Service — Creating DQS\$SERVER

V5.2

With the more security-conscious VMS Version 5.2, creating a default DECNET account is not the default choice of the command procedure NETCONFIG.COM. The DIGITAL Distributed Queuing Service (DECdqs) Version 1.1 uses the default DECNET account on the DECdqs client to receive notification from the DECdqs server that a job has completed printing, when a user has used the PRINT/NOTIFY command. If your DECdqs client-only VAX computer has no default DECNET account, users on your system cannot receive notification of job completion.

A VAX computer configured as a DECdqs server (and also client) uses the DQS\$SERVER account to receive notification of job completion when printing to other DECdqs servers. To receive notification of job completion on a DECdqs Version 1.1 client-only machine, create a DQS\$SERVER

System Manager Release Notes

3.15 DIGITAL Distributed Queuing Service — Creating DQS\$SERVER

account. Use the Authorize Utility to specify a unique user identification code (UIC) and a password that cannot easily be guessed, as follows:

```
UAF> ADD DQS$SERVER/UIC=[300,311]/DIRECTORY=[DQS$SERVER] -  
_UAF> /DEVICE=SYS$COMMON/PASSWORD=somepassword -  
_UAF> /FLAGS=(NODISUSER,RESTRICTED)
```

Remember to create the directory SYS\$COMMON:[DQS\$SERVER]. You then must do the following using the Network Control Program (NCP), using the same password you specified in AUTHORIZE:

```
NCP> DEFINE OBJECT DQS NUMBER 66 FILE DQS$SERVER.EXE PROXY NONE -  
_NCP> USER DQS$SERVER PASSWORD somepassword  
NCP> SET KNOWN OBJECTS ALL
```

3.16 Disk Header Space Problem

V5.4

Large-capacity disks (such as RA90s) may run out of file header space before they run out of free blocks and before they reach the MAXFILES value. This problem will be corrected in a future release of VMS.

As new files are added to a large-capacity disk or as new extents (sets of contiguous clusters) are created for existing files, the INDEXF.SYS file must be extended. The size of extents for INDEXF.SYS is fixed at 1000 blocks. While this size is acceptable for smaller disks, it only allows the INDEXF.SYS file to be extended for approximately 50,000 file extents. When the file extent limit has been reached, the single file header for INDEXF.SYS does not have additional room for file-header map information. In this case, the following error message, preceded by RMS-related error messages, is displayed:

```
SYSTEM-W-HEADERFULL, file header is full
```

You cannot increase the size of the INDEXF.SYS file once it reaches its header limit. To work around this space problem, you must use the BACKUP/IMAGE command to compress the data on the disk when the INDEXF.SYS file reaches its header limit.

To determine whether a particular disk is close to exhausting its headers for INDEXF.SYS, count the number of existing file extents for INDEXF.SYS, as follows:

- 1 Enter the following command line:

```
$ DUMP/HEADER/BLOCKS=COUNT:0 device:[000000]INDEXF.SYS
```

A sample portion of the output is as follows:

System Manager Release Notes

3.16 Disk Header Space Problem

```
Dump of file device:[000000]INDEXF.SYS;1
.....
Header area
.....
Map area
  Retrieval pointers
    Count:      6          LBN:      0
    Count:      3          LBN:     966
    Count:      3          LBN:  1223883
    Count:    12000        LBN:  1188273
    Count:     1002        LBN:   408198
.....
    Count:     1002        LBN:  1183425
    Count:      918        LBN:  1078845
    Count:      256        LBN:   629643
```

- 2 Examine the Map area section of the output and count the retrieval pointers.

The header for INDEXF.SYS can contain slightly more than 50 retrieval pointers. The first four retrieval pointers are created when the volume is initialized, and subsequent retrieval pointers are added as INDEXF.SYS needs to be expanded (when new files are created or extents are added to existing files).

INDEXF.SYS is extended in blocks of 1000 (increased to a multiple of the cluster factor of the volume). However, if no contiguous extents are as large as 1000 blocks (because the volume is fragmented), INDEXF.SYS is extended in smaller extents. Note that at least 1 block is required in the INDEXF.SYS file for each extent of each file on the disk. (More than 1 block per extent may be required if the file has a very large ACL.)

If the number of retrieval pointers approaches 50 (and especially when the last retrieval pointers show extents of less than 1000 blocks, as in the preceding sample output), the disk is approaching its limit for INDEXF headers.

If a disk is approaching its limit for INDEXF headers, use the BACKUP/IMAGE command to compress the data on the disk. It is important that you use the /INIT qualifier to create a contiguous INDEXF.SYS that is at least as large as the INDEXF.SYS file on the original disk.

When you initialize a new disk that will have individual files created by users and applications (as opposed to files created by restoring an image backup of or from another disk), Digital recommends that you estimate the total number of files you expect on that disk and that you use the INITIALIZE/HEADERS=*n* command to preallocate that number of file headers in INDEXF.SYS.

Note: The BACKUP/IMAGE/NOINIT command does not preserve the size of INDEXF.SYS specified with a preceding INITIALIZE/HEADERS=*n* command. For more information about the BACKUP command, see the *VMS Backup Utility Manual*.

System Manager Release Notes

3.17 DISMOUNT Command — Changes Regarding Open Files

3.17 DISMOUNT Command — Changes Regarding Open Files

V5.2

In previous versions of VMS, the DCL command DISMOUNT performed a set of relatively simple tests before attempting to dismount a Files-11 volume. These tests verified that the volume was in fact mounted, that it was not the system disk, and that the user had the necessary privileges to dismount the volume. If these tests ran successfully, the volume was marked for dismount and the DISMOUNT command returned a success status.

The dismount of a Files-11 volume actually completed when the volume became idle, that is, when the file system determined that all files in the volume had been closed. In many instances the user might not have been aware of open files on the volume until it was discovered that the volume had remained in the marked-for-dismount state for an extended period of time. At this point, however, the volume was committed to being dismounted regardless of the consequences brought about by closing the open files, if in fact they could be closed (see Section 3.17.1).

With VMS Version 5.2 and subsequent versions, the DISMOUNT command checks for conditions that will prevent the dismount from completing. The conditions are categorized as follows:

- Installed swap and page files
- Installed images
- Devices spooled to the volume
- Open user files (any files not falling into one of the first three groups)

If none of these conditions are found, the volume is marked for dismount as usual, and the volume changes quickly from the marked-for-dismount state to the dismounted state. If any of these conditions exists, the DISMOUNT command does not mark the volume for dismount, but instead displays messages indicating that the volume cannot be dismounted, the conditions that exist, and the number of instances of each condition. For example:

```
$ DISMOUNT $10$DJA100:

%DISM-W-CANNOTDMT, $10$DJA100: cannot be dismounted
%DISM-W-INSWPGFIL, 4 swap or page files installed on volume
%DISM-W-SPOOLEDEV, 3 devices spooled to volume
%DISM-W-INSTIMAGE, 7 images installed on volume
%DISM-W-USERFILES, 6 user files open on volume
```

As shown in the example, the conditions are displayed in order of decreasing severity (severity refers to the level of difficulty you would have rectifying the conditions).

The return status from the DISMOUNT command reflects the most severe conditions. You can use this return status to construct a command procedure or image that calls routines to handle the individual conditions. Once one condition has been addressed, the procedure should loop back and attempt the DISMOUNT command again to determine if other conditions exist. The symbol names and values for the four conditions are:

System Manager Release Notes

3.17 DISMOUNT Command — Changes Regarding Open Files

```
DISM$_INSWPGFIL = %X739018
DISM$_SPOOLEDEV = %X739020
DISM$_INSTIMAGE = %X739028
DISM$_USERFILES = %X739030
```

3.17.1 Closing Files

V5.2

With VMS Version 5.2 and subsequent versions, you can address all the conditions that prevent a volume from being dismounted if you have the appropriate privileges. In previous versions of VMS, you could not dismount disks with installed secondary swap and page files. Disks with secondary swap and page files were considered an extension of the system disk, which cannot be dismounted. You can now cancel the installed status of these files, thereby allowing you to dismount the volume.

Some knowledge of the files specific to your environment may be required to eliminate the conditions preventing a volume from being dismounted.

First you must determine the names of the files open on the device and the process that owns each file. Each file can then be addressed as shown in the following sections. This information can be displayed using the following command (*ddcu* is the name of the device you are attempting to dismount):

```
$ SHOW DEVICE/FILES ddcu:
```

System-Owned Files (Process ID = 0) with the Extension SYS

The files INDEXF.SYS and QUOTA.SYS can remain open. INDEXF.SYS is normally open on any mounted volume. QUOTA.SYS is normally open if quotas are enabled on the volume. Neither of these open files prevents the volume from being dismounted.

Any remaining files with the extension SYS are most likely installed secondary swap and page files. You can verify this by examining the site-specific system startup file SYS\$MANAGER:SYPAGSWPFILES.COM and by using the DCL command SHOW MEMORY/FILES/FULL. To cancel the installed status of these files, use one of the following SYSGEN commands:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> DEINSTALL filespec[/PAGEFILE ]
SYSGEN> DEINSTALL filespec[/SWAPFILE ]
SYSGEN> DEINSTALL/INDEX=page-file-number
```

For further information, refer to SYSGEN's online help.

System-Owned Files (Process ID = 0) with the Extension EXE

System-owned files with the extension EXE are most likely installed images. You should verify this by examining the installed-image list using the VMSINSTAL command LIST. You can then cancel the installed status of the files, as described in the *VMS Install Utility Manual*.

System Manager Release Notes

3.17 DISMOUNT Command — Changes Regarding Open Files

Process-Owned Files

Process-owned files are normally closed when the processes accessing the files finish with them. Contact the users who own the processes and ask them to complete their work and close the files or log out. If this cannot be done, you can force the processes to exit using the DCL command `STOP PROCESS/ID=process-id`.

Spooled Devices

You can locate spooled devices using the DCL command `SHOW DEVICE`. The `SHOW DEVICE` command displays “spooled” in the device status field if the device is spooled. You can examine the system startup command procedure `SYS$MANAGER:SYSTARTUP_V5.COM` to determine whether the device is spooled to the volume that is being dismounted and to get the names of the queues used by the spooled device. Once you have done this, you should first prevent any queued files from being lost by setting the queue to retain jobs on error, as follows:

```
$ SET QUEUE/RETAIN=ERROR queue-name
```

Next, stop the queue while queuing the current job again but by placing it on hold as follows:

```
$ STOP/QUEUE/REQUEUE/HOLD queue-name
```

The device can then be set not to be spooled:

```
$ SET DEVICE/NOSPOOLED device
```

You can now restart the queue without losing any jobs in the queue or any files that have been spooled to the volume. If you do not want to wait until the volume is remounted to restart the queue, you can set the device to be spooled to a different volume and restart the queue immediately.

3.17.2 Clusterwide Support for DISMOUNT

V5.2

You can use the `DISMOUNT` command throughout the cluster if you specify `DISMOUNT/CLUSTER`. This command first checks for conditions that will prevent the volume from dismounting on the local node. If none is found, it then checks for such conditions on all of the other nodes in the cluster. If the command `DISMOUNT/CLUSTER` finds one of the conditions on any node, it sends an error message identifying the device and the node on which the error occurred, followed by an error message indicating that there are open files on the volume. For example:

```
$ DISMOUNT/CLUSTER $10$DJA100:
```

```
%DISM-W-RMTDMTFAIL, $10$DJA100: failed to dismount on node SALT  
%DISM-W-FILESOPEN, volume has files open on remote node  
%DISM-W-RMTDMTFAIL, $10$DJA100: failed to dismount on node PEPPER  
%DISM-W-FILESOPEN, volume has files open on remote node  
%DISM-W-CANNOTDMT, $10$DJA100: cannot be dismounted
```

In this example, the final return status is `DISM-W-CANNOTDMT`. Note that while this message is also displayed when one of the error conditions is found on the local node, it acts as a return status only if the conditions are found on a remote node. Thus, it can be used in a command procedure

System Manager Release Notes

3.17 DISMOUNT Command — Changes Regarding Open Files

or an image to distinguish the location of the error condition. The symbol and value for this status are:

```
DISM$_CANNOTDMT = %X739010
```

3.17.3 Restoring the Previous Behavior of the DISMOUNT Command

V5.2

In some cases you may want to mark a volume for dismount even though files are open on the volume. Marking the volume for dismount prevents users from opening any new files, thereby allowing activity to wind down. Also, file-system caches are flushed at the time the volume is marked for dismount, which is especially important when the system is shutting down and the file-system caches must be written to the disk. For these reasons, the qualifier `/OVERRIDE=CHECKS` has been provided for the DCL command DISMOUNT to override the new VMS Version 5.2 behavior and allow the volume to be marked for dismount despite the fact that there are files open.

If you specify the qualifier `/OVERRIDE=CHECKS`, the DISMOUNT command reverts to the earlier behavior with the following exception. Informational messages are displayed to inform you of conditions that will prevent the volume from dismounting, immediately followed by an informational message indicating that the volume has been marked for dismount. The final status is success with a severity of informational (`DISM$_MARKEDDMT`). For example:

```
$ DISMOUNT/OVERRIDE=CHECKS $10$DJA100:
%DISM-I-INSWPGFIL, 2 swap or page files installed on volume
%DISM-I-SPOOLEDEV, 1 device spooled to volume
%DISM-I-INSTIMAGE, 5 images installed on volume
%DISM-I-OPENFILES, 3 user files open on volume
%DISM-I-MARKEDDMT, $10$DJA100: has been marked for dismount
```

You can specify the equivalent of the qualifier `/OVERRIDE=CHECKS` when using the `$DISMOU` system service by using the new `DMT$_M_OVR_CHECKS` flag. You should specify this flag in the `flags` argument to the `$DISMOU` system service if you desire the behavior of previous versions of VMS.

The command procedure `SYS$SYSTEM:SHUTDOWN.COM` has been modified in VMS Version 5.2 to specify the `/OVERRIDE=CHECKS` qualifier when dismounting volumes.

You must dismount DIGITAL Distributed File Service (DECdfs) client pseudodevices (`DFSCn:`) using the command DISMOUNT `/OVERRIDE=CHECKS DFSCn:`. For example:

```
$ DISMOUNT/OVERRIDE=CHECKS DFSC1001:
```

The following informational message will be displayed, and the device will be dismounted:

```
%DISM-I-USERFILES, 1 user file open on volume
%DISM-I-MARKEDDMT, DFSC1001 has been marked for dismount
```


3.18 Documentation — Title Changes for Installation and Upgrade Manuals

3.18 Documentation — Title Changes for Installation and Upgrade Manuals

V5.4 Beginning with VMS Version 5.4, many of the VMS installation and operations manuals have new titles. For example, *VMS Installation and Operations: VAX 8600, 8650* is now named *VMS Upgrade and Installation Supplement: VAX 8600, 8650*. You should consult these installation supplements for information on updating, upgrading, or installing VMS on your specific processor.

For a complete list of title changes, see the *Overview of VMS Documentation*.

3.19 DSDRIVER/LOCKMANAGER Problem Corrected

V5.3-2 Since VMS Version 5.2-1, DSDRIVER did not properly account for the maximum size of a lock ID during its internal synchronization on a shadow set. This could result in an operation using an incorrect lock, given a sufficiently large number of locks in the system. If this problem occurred, the result was a DISKCLASS bugcheck.

This lock ID problem has been corrected in VMS Version 5.3-2.

3.20 DSSI Device Naming No Longer Dependent on SYSGEN Parameter VMS5

V5.4 With VMS Version 5.3, the device names assigned to DIGITAL Storage System Interconnect (DSSI) disks attached to a KFQSA controller changed.

In Version 5.3, the DSSI device-naming scheme depended on the SYSGEN parameter VMS5 (in order to alleviate some of the problems anticipated with the change). With VMS5 set to 1, the old (prior to Version 5.3) device-naming scheme continued to be used, whereas setting VMS5 to zero enabled the new device-naming scheme. Systems that installed Version 5.3 for the first time had VMS5 set to zero by default. Systems that upgraded from a previous version of VMS had VMS5 set to 1, but Digital recommended that VMS5 be set to zero as soon as practical after upgrading to Version 5.3.

Beginning with VMS Version 5.4, all systems use the new device-naming scheme for DSSI disks, regardless of the value of the SYSGEN parameter VMS5. VMS5 is no longer used in determining device names.

An explanation of the DSSI device-naming scheme follows.

DSSI Device-Naming Scheme

In versions of VMS prior to Version 5.3, the DSSI device name was in the form *Dicu*, where *c*, the controller letter, was A, B, C, and so forth. The controller letter was taken from the device name of the port (PUA0, PUB0, PUC0, and so forth) representing the DSSI disk. If the allocation class *n* of the DSSI disk was nonzero, then the device name was in the form *\$n\$Dicu*. This scheme was inconsistent with the naming used for DSSI disks attached to embedded adapters, such as those on MicroVAX 3300/3400 systems.

System Manager Release Notes

3.20 DSSI Device Naming No Longer Dependent on SYSGEN Parameter VMS5

With the new naming scheme, the device name of a DSSI disk no longer depends on the device name of the port that represents the disk. Instead, all DSSI disks use the controller letter A. Thus, device names are now in the form $n\$DIAu$, where n is the nonzero allocation class of the DSSI disk, or $node-name\$DIAu$ if the allocation class is zero. Note that *node-name* is the node name of the DSSI disk and is not the same as the VMS parameter SCSNODE.

For example, a single KFQSA controller with three DSSI disk drives attached would have the device names listed in Table 3-3 for ports or disks, or both.

Table 3-3 KFQSA Controller Device Names

	Allocation Class=0		Allocation Class=4	
	Port	Disk	Port	Disk
Old scheme	PUA0	DIA0	PUA0	\$4\$DIA0
	PUB0	DIB1	PUB0	\$4\$DIB1
	PUC0	DIC2	PUC0	\$4\$DIC2
New scheme	PUA0	FRED\$DIA0	PUA0	\$4\$DIA0
	PUB0	BARNEY\$DIA1	PUB0	\$4\$DIA1
	PUC0	WILMA\$DIA2	PUC0	\$4\$DIA2

A benefit of the new device-naming scheme is that two systems in a dual-host configuration will always use the same device name for a shared DSSI disk. With the old device-naming scheme, which included the port controller letter for KFQSA-connected devices, a dual-host configuration with multiple KFQSA controllers per system could result in inconsistent device names across the two systems if the common DSSI was incorrectly attached (for example, if KFQSA controller 1 on MicroVAX A were attached to KFQSA controller 2 on MicroVAX B). The old scheme also precluded dual-hosting with mixed adapter types (embedded adapters and KFQSA).

With the new scheme, all systems, regardless of adapter type, use device names $n\$DIAu$ or $node-name\$DIAu$, on which the only variables are the allocation class, node name, and unit number of the DSSI disk. Because each of these parameters is associated with the disk itself, all systems with access to the disk will use the same device name. As a result, the new naming scheme allows dual-host configurations with multiple KFQSA controllers per system and mixed adapter types.

However, all DSSI disks must have unique device names. Therefore, Digital recommends that, for configurations with multiple DSSIs and many DSSI disks, each disk be given a unique unit number. You can do this by first setting the disk parameter FORCEUNI to zero and then by setting UNITNUM to the desired value. FORCEUNI is set to 1 by default, which forces the unit number to equal the device's node ID on the DSSI, regardless of the value of UNITNUM.

System Manager Release Notes

3.20 DSSI Device Naming No Longer Dependent on SYSGEN Parameter VMS5

To set any of the disk parameters (ALLCLASS, NODENAME, FORCEUNI, or UNITNUM) for KFQSA-connected DSSI disks, use the following procedure for each device:

- 1 For MicroVAX and VAXserver 3400/3600/3900 series systems, enter the SHOW DEVICE command at the console-mode prompt (>>>) to display the UQSSP controller number.
- 2 Enter the command SET HOST/DUP/UQSSP/DISK *n* PARAMS, where *n* is the UQSSP controller number of the device.
- 3 At the PARAMS> prompt, you can use the SHOW/SET commands to examine and change the values of device parameters. Then enter the WRITE command to write any new parameter values to nonvolatile storage in the device. (Changing ALLCLASS or NODENAME requires that the controller be initialized.)

For more information on the console command SET HOST/DUP, see the section “Configuring RF30 and RF71 Devices in a VAXcluster” in the *VMS Installation and Operations: MicroVAX, VAXstation, and VAXserver 3400, 3600, 3900 Series*, or the hardware information for your system.

3.21 DUDRIVER and DSDRIVER — Change to Improve Failover

V5.1

Changes have been made to the disk class drivers DUDRIVER and DSDRIVER that improve failover on dual-pathed DSA disks (for example, RA60, RA81) that are connected to local controllers. A node with a local controller that is accessing a disk through the MSCP server on the other node now discovers its local, secondary path soon after boot and switches to that path if the disk becomes unreachable through the remote server.

3.22 Dump File Notes

The release notes in this section pertain to dump files.

3.22.1 Dump File Size Changes

V5.0

The system bugcheck mechanism has been rewritten to allow selective dumps, and the System Dump Analyzer (SDA) has been enhanced to analyze both complete and selective dumps. These changes have been included so that dump files for large memory systems will not consume large amounts of disk space. Selection of complete or selective dumps is accomplished by using the new SYSGEN parameter DUMPSTYLE. The value of 0 enables the traditional complete memory dump. Setting DUMPSTYLE to 1 enables the selective dump.

As a result of this enhancement, modifications were made to the sizing algorithm for AUTOGEN's dump file. This allows AUTOGEN to produce smaller dump files. To enable the smaller selective dumps and AUTOGEN's new dump file sizing algorithm, set the parameter DUMPSTYLE=1 in MODPARAMS.DAT. If you have overridden AUTOGEN's dump file calculation by defining a symbol DUMPFIL=0 in MODPARAMS.DAT, remove this symbol definition to let AUTOGEN create a smaller dump file.

System Manager Release Notes

3.22 Dump File Notes

3.22.2 **SAVEDUMP Parameter and PAGEFILE.SYS Size — Interaction Caution**

V5.0

When a system dump is saved in the page file, the SYSGEN parameter SAVEDUMP specifies whether the space used by the dump should be reserved until the dump is analyzed. If your system is configured to write a complete physical memory dump (DUMPSTYLE set to 0, the default) you can calculate the size of the dump and make your page file large enough to hold both the dump and the required additional paging space.

However, a more difficult situation arises when you are dumping to the page file and your system is configured to write a selective dump file (DUMPSTYLE set to 1). Selective dumps write out processes until they are all dumped or until dump file space is exhausted. If you reduce your paging file to a size smaller than what is required for a full dump in an attempt to save disk space by selective dumping, the selective dumps might use up the required additional paging space and the dump would be discarded on reboot.

Although Digital does not recommend enabling SAVEDUMP, it does recognize that this parameter allows some configurations with very restricted disk space to save crash dumps. Systems that enable SAVEDUMP need a substantially larger primary page file in order to preserve system dumps.

3.22.3 **Selective Crash Dump Files — Caution**

V5.0

One of AUTOGEN's functions is to recommend a dump file size based on your system's physical memory and other system parameters. By default, this procedure has not changed and continues to work correctly. However, if you enable selective dumps, and NETACP uses a large (over 1000 pages) address space on your system, AUTOGEN's algorithm may not recommend a page file large enough to hold as many processes as may make a particular crash dump useful for analysis.

Selective dumps represent a tradeoff between the usefulness of a crash dump and the disk space required to hold it. There is no formula about how many processes are useful to dump. However, if you notice that only a small fraction of memory-resident processes are dumped during a bugcheck, you probably should increase the size of your system dump file according to the approximate size of the NETACP working set. You can do this by noting NETACP's working set size as given by the DCL command SHOW SYSTEM and in SYS\$SYSTEM:MODPARAMS.DAT and then either increasing the value of DUMPFIL by that amount or adding a line to have DUMPFIL increased beyond AUTOGEN's calculated value. You then need to invoke AUTOGEN as follows:

```
$ @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT
```

3.22.4 Shared Dump Files

V5.0

A **shared dump file** is a system dump file that is used by two or more nodes in the VAXcluster environment. To allow dump files to be shared among nodes in a VAXcluster environment, perform the following steps:

- 1 Create the shared dump file
SYS\$COMMON:[SYSEXE]SYSDUMP-COMMON.DMP by entering the following command lines (*n* is the size of the system dump file):

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> CREATE SYS$COMMON:[SYSEXE]SYSDUMP-COMMON.DMP/SIZE=n
SYSGEN> EXIT
$
```

- 2 For each VAXcluster node sharing the file, enter the following command line (*n* is the system-specific root for the node):

```
$ SET FILE SYS$COMMON:[SYSEXE]SYSDUMP-COMMON.DMP -
_ $ /ENTER=SYS$SYSDEVICE:[SYSn.SYSEXE]SYSDUMP.DMP
```

Note: You can use the file name SYSDUMP.DMP for a shared dump file. However, when you upgrade your cluster, you must change the name. The VMS upgrade procedure does not allow a shared dump file to be named SYSDUMP.DMP.

3.23 Ethernet Notes

The release notes in this section pertain to the Ethernet.

3.23.1 DEBNI Ethernet/802 Controller — New Support for the VAXBI Bus

The notes in this section pertain to the DEBNI Ethernet controller.

3.23.1.1 I/O Interface

V5.2

VMS Version 5.2 and subsequent versions support the DEBNI controller, which is a new Ethernet/802 controller that connects to the VAXBI bus. The QIO interface to the DEBNI controller is the same as that described for the DEBNA device driver in the *VMS I/O User's Reference Manual: Part II*, except that the device type of the DEBNI controller is DT\$_ET_DEBNI.

The DEBNI controller is supported by ETDRIVER. The DEBNI device name is as follows, where *c* is the controller and *u* is the unit number:

ET*c**u*

For example, ETA0 is the device name for the first DEBNI controller in the system.

The NCP LINE and CIRCUIT name for the DEBNI controller is as follows:

BNA-<controller-number>

For example, BNA-0 and BNA-1 are the NCP LINE and CIRCUIT names for the first and second DEBNI controllers in the system, respectively.

System Manager Release Notes

3.23 Ethernet Notes

3.23.1.2 Node ID

V5.2

If you configure a DEBNI controller at a lower node ID than the BI disk adapters, the disk adapter controller letter will be incremented incorrectly. For example, instead of DJA0, the disk would be DJB0.

This problem will be corrected in a future release of the VMS operating system.

3.23.2 DEMNA Ethernet/802 Controller — New Support for the XMI Bus

V5.4

The DEMNA controller (DECLANcontroller 400) is a new Ethernet/802 controller that connects to the XMI bus. The QIO interface to the DEMNA controller is the same as that described for the DEBNA device driver in the *VMS I/O User's Reference Manual: Part II*, except that the device type for the DEMNA controller is DT\$_EX_DEMNA.

The DEMNA controller is supported by EXDRIVER. Its device name is as follows, where *c* is the controller and *u* is the unit number:

EX*cu*

For example, EXA0 is the device name for the first DEMNA controller in the system.

The NCP line and circuit name for the DEMNA controller is as follows:

MNA-<controller-number>

For example, MNA-0 and MNA-1 are the NCP LINE and CIRCUIT names for the first and second DEMNA controllers in the system, respectively.

3.23.3 DEQNA Ethernet Adapter May Receive Corrupt Data

V5.0

Under certain rare circumstances, the DEQNA Ethernet adapter in large and complex Ethernet configurations may receive corrupted data. The VMS operating system automatically enables a data integrity feature that reduces the risk to VAXcluster users. Digital recommends that this feature remain enabled on all VAXcluster members that use DEQNA devices.

The DECnet command COPY provides data integrity checking. User-written applications performing data transfers to systems using DEQNA adapters must provide their own data integrity checking.

3.23.4 DEQTA Ethernet/802 Controller — New Support for the Q-bus

V5.3

Beginning with VMS Version 5.3, VMS supports the DELQA-Plus controller, which is a new Ethernet/802 controller that connects to the Q-bus. VMS refers to the DELQA-Plus controller as the DEQTA controller. The QIO interface to the DEQTA is the same as that described for the DESVA in the *VMS I/O User's Reference Manual: Part II*, except that the device type of the DEQTA is DT\$_XQ_DEQTA.

The DEQTA controller is supported by XQDRIVER. The DEQTA device name is as follows, where *c* is the controller and *u* is the unit number:

`XQcu`

For example, XQA0 is the device name for the first DEQTA controller in the system.

The NCP LINE and CIRCUIT name for the DEQTA controller is as follows:

`QNA-<controller-number>`

For example, QNA-0 and QNA-1 are the NCP LINE and CIRCUIT names for the first and second DEQTA controllers in the system, respectively.

3.23.5 Ethernet Controllers — Tuning the VMS Operating System

For Ethernet controllers, tune the VMS operating system by adjusting the network parameters and SYSGEN parameters, as described in the following sections.

3.23.5.1 Network Parameters — DEBNA Controllers

V5.0

For the DEBNA controller, check and adjust the network parameters in the following list:

- LINE BUFFER size must be 1498

Check the LINE BUFFER size. Make sure that it is set to 1498.

- LINE RECEIVE BUFFERS must be at least 8

The LINE RECEIVE BUFFERS parameter should not be set to a value of less than 8. If it is set to less than 8, it may cause an excessive loss of packets in the controller (DEBNA).

- HELLO TIMER

Adjust this parameter if Adjacent Node Listener Receive timeouts occur by increasing the Hello Timer value on the adjacent node.

- BUFFER_LIMIT in LOADNET.COM

The BUFFER_LIMIT should be increased for each additional DECnet line. Increase it from the default of 65K. The typical value for four lines is 131K. Line Open errors occur if BUFFER_LIMIT is too small.

3.23.5.2 Network Parameters — DEBNA, DEBNI, and DEMNA Controllers

V5.4

For the DEBNA, DEBNI, and DEMNA controllers, check and adjust the network parameters for BUF_LIM in LOADNET.COM.

You should increase the default for BUF_LIM, which is 65K, for each additional DECnet line. If BUF_LIM is too small, "exceeded quota" messages will be displayed when DECnet is started or the circuits will stay in the synchronizing state when the circuits are turned off.

System Manager Release Notes

3.23 Ethernet Notes

3.23.6 Second-Generation Ethernet Controller (SGEC) Now Supported

V5.4

VMS Version 5.4 supports the Second-Generation Ethernet Controller (SGEC), which is a new Ethernet/802 controller. The QIO interface to the SGEC is the same as that described for the DESVA device driver in the *VMS I/O User's Reference Manual: Part II*, except that the device type for the SGEC is DT\$_EZ_SGEC.

The SGEC is supported by EXDRIVER. Its device name is as follows, where *c* is the controller and *u* is the unit number:

```
EZcu
```

For example, EZA0 is the device name for the first SGEC controller in the system.

The NCP line and circuit name for the SGEC is as follows:

```
IS-<controller number>
```

For example, ISA-0 and ISA-1 are the NCP LINE and CIRCUIT names for the first and second SGEC controllers in the system, respectively.

3.24 INITIALIZE Command — Defining Volume Serial Numbers

V5.0

Since the introduction of DIGITAL Storage Architecture (DSA) disks, the VMS operating system has not properly initialized the Files-11 On-Disk Structure Level for a DSA disk to include the hardware serial number stored in the DSA volume's factory formatting data. This causes the home block SERIALNUM field to be zero, which in turn causes the \$GETDVI system service and the F\$GETDVI lexical function to return zero for the SERIALNUM item, instead of a unique serial number.

With VMS Version 5.0, the Files-11 initialization process has been corrected. The correction has been made both in the INITIALIZE command and in the BACKUP/IMAGE command. This means that all output volumes processed by these two commands will have properly defined serial numbers and return something other than zero from the \$GETDVI item code SERIALNUM.

Volumes transported to Version 5.0 and not processed with one of these two commands will continue to report zero for the SERIALNUM item, because the home blocks on such volumes will continue to contain zero in the SERIALNUM field. The most convenient way to overcome this problem is to perform a BACKUP/IMAGE on volumes where a correct SERIALNUM value is deemed important.

In addition, there is a restriction on SERIALNUM initialization by BACKUP/IMAGE. The process executing BACKUP must have LOG_IO privilege, or BACKUP must be installed with LOG_IO privilege. This is because the I/O function used to obtain the SERIALNUM information for inclusion in the home block is a physical I/O function. Digital expects to remove this restriction in a future release of the VMS operating system.

System Manager Release Notes

3.24 INITIALIZE Command — Defining Volume Serial Numbers

Finally, all these services are restricted to directly accessed DSA disks. DSA disks accessed through the MSCP server will continue to initialize with the SERIALNUM field set to zero. This restriction results from limitations in the MSCP server with respect to serving DSA disks. Digital will remove this restriction in a future release of the VMS operating system.

3.25 MA780 (Multiport Shared Memory)

V5.1

All processors connected to the multiport shared memory (MA780) must be running the same version of VMS, either Version 4.x or Version 5.x. Running one processor at Version 4.x and another at Version 5.x does not work because of changes in the global section data structure for VMS Version 5.0.

3.26 Magnetic Tape ACP Notes

The release notes in this section pertain to the magnetic tape ancillary control process (ACP).

3.26.1 Correction to I/O Process

V5.4

In previous versions of VMS, a problem with the magnetic tape ancillary control process (ACP) prevented tape I/O from completing, leaving a process in the LEF state indefinitely. This I/O problem occurred because pending user I/O was not being processed by the ACP after tape errors occurred during ACP processing.

This problem has been corrected for VMS Version 5.4. Pending I/O is now returned to the process, with the error status reported to the ACP rather than being held by the ACP.

3.26.2 Undocumented Implementation Removed

V5.4

In previous versions of VMS, the magnetic tape ancillary control process (ACP) implemented an undocumented variation of mount verification. The ACP issued operator requests when a volume went off line, and then attempted to reposition the volume to the correct record when the volume came back on line.

With VMS Version 5.4, the undocumented implementation specific to the magnetic tape ACP has been removed.

Mount verification has been available for tape drives since VMS Version 5.0 through the /MOUNT_VERIFICATION qualifier to the MOUNT command. For a complete description of /MOUNT_VERIFICATION, see the *VMS Mount Utility Manual*.

System Manager Release Notes

3.27 Mail Utility — Changes to PRINT/QUEUE Command

3.27 Mail Utility — Changes to PRINT/QUEUE Command

- V5.4** Prior to VMS Version 5.4, the interactive Mail Utility (MAIL) command PRINT/QUEUE did not check on the target queue's attributes to determine whether or not it actually was a print queue.
- Beginning with VMS Version 5.4, MAIL requires that the queue specified with the /QUEUE qualifier (or taken from the SYS\$PRINT logical name) be an actual print queue. If you attempt to print to a batch queue, MAIL returns a CREPRIJOB error.

3.28 Mass Storage Control Protocol (MSCP) Server

The release notes in this section pertain to the mass storage control protocol (MSCP) server.

3.28.1 Controller Letters — Restriction

- V5.1-1** The MSCP server serves only disks with controller letters A through G and shadow set virtual units that have the controller letter S. This restriction will be lifted in a future release.

3.28.2 Diskette Devices — Some Not Allowed

- V5.0** The mass storage control protocol (MSCP) does not allow all the functions associated with certain diskette devices. Therefore, the MSCP server (which is based upon MSCP) does not automatically serve diskette devices such as the RX01, RX02, and RX33.

3.29 Modified-Page Writer — Flushing of Modified-Page List Eliminated

- V5.2** Prior to VMS Version 5.2, the modified-page list was completely written or "flushed" under the following circumstances:

- Deletion of a global section with file backing store
- Balance-slot cleanup for a deleted process
- A process dead-page-table scan
- An operator-induced crash using the OPCCRASH procedure

The frequency of flushing depended on workload and configuration, and on some systems, there was a significant negative effect on system performance.

With VMS Version 5.2, all flushing of the modified-page list has been eliminated and replaced with a selective modified-page writing mechanism. Assuming that a system is configured with adequate main memory for its workload, there are two major benefits of the new approach:

- The size of the average modified-page list is significantly higher, increasing its effectiveness as a page cache.

System Manager Release Notes

3.29 Modified-Page Writer — Flushing of Modified-Page List Eliminated

- The average free-page file space is significantly greater, decreasing the frequency of process or system hangups due to insufficient page file space.

3.30 Modular Executive Notes

The release notes in this section pertain to the Modular Executive.

3.30.1 Effects on Privileged Code

V5.0

The reorganization of the Executive for VMS Version 5.0 affects only privileged code. All privileged code must be relinked with the Version 5.0 linker against the new Version 5.0 SYS.STB, the system symbol table.

In previous versions of VMS, several data structures were statically declared in the SYS.EXE image. With VMS Version 5.0, some of these data structures have been moved into one of the loadable executive images. All other loadable executive images and privileged images must reference the structure through a pointer stored in the base image. When data was moved out of the base image, the name of the cell was changed so that any code referencing the cell would not link with undefined symbols. Any privileged image that fails to link in such a way requires source-code changes to reference these data structures through pointers to these structures.

When you are debugging privileged code or device drivers, it may be necessary to disable system paging. The special SYSGEN parameter SYSPAGING was provided for this purpose. For VMS Version 5.0 and subsequent versions, the SYSPAGING parameter has been replaced with another special parameter, S0_PAGING, which is a mask with a "1" bit to disable paging. If bit 0 (low-order bit) of S0_PAGING is set, then paging of the Modular Executive is disabled. If bit 1 of S0_PAGING is set, then paging of RMS is disabled. Note that the S0_PAGING parameter is a special SYSGEN parameter and should be used only by your Digital Customer Services representative.

3.30.2 Effects on SYSGEN

V5.0

The SYSGEN CONNECT/drivername command specifies the name of the driver as recorded in the prologue table. If the driver has not been loaded, the system assumes that the driver name is also the name of an executable image (file type of EXE) in the SYS\$LOADABLE_IMAGES or the SYS\$SYSTEM directory, and loads the driver. The default for the driver name is the first two characters of the device name plus DRIVER.

3.30.3 Effects on System Management

The following sections describe the effect of the modular executive upon system images.

System Manager Release Notes

3.30 Modular Executive Notes

3.30.3.1 SYSSLOADABLE_IMAGES Directory on the System Disk

V5.0

The SYSSLOADABLE_IMAGES logical name points to a special directory on the system disk. This directory contains the set of images that are loaded during the bootstrap of the system. The Executive loaded images, device drivers, and other images loaded into system space (for example, SYSLOA780.EXE) reside in this directory. Images in this directory are special in that they are not executable in the conventional sense; that is, they cannot be executed with RUN or other DCL commands.

3.30.3.2 System Failure

V5.0

If the system fails, or if the system is forced to fail in an emergency shutdown with CRASH, the list of Executive loaded images in the system is printed on the console terminal. The bugcheck message and information about the stack are printed, followed by the list of Executive loaded images with the starting and the ending address of each image. Then a dump of memory is written to the system dump file on disk.

3.30.4 Introduction to the Modular Executive

V5.0

All of the code contained previously in the image SYS.EXE has now been separated into approximately 20 images, called loadable executive images. All executable code in the Modular Executive is contained in these images.

The partitioning of SYS.EXE is meant to group together modules that logically belong together in terms of the functions they perform. For example, modules that deal with image activation and image rundown were moved to an image called IMAGE_MANAGEMENT.EXE, while modules related to system security were moved into an image called SECURITY.EXE.

In the Modular Executive, SYS.EXE remains as one of the many executive images. However, SYS.EXE, now called the base image, has some unique functions:

- It provides a transfer vector area in system (S0) space for routines of the loadable executive images.
- It includes an area for universal data cells, which are cells that all code in both the Executive and other privileged images can access.

All transfer vectors and global data cells within the base image are permanently fixed. The base image is the unchanging pathway by which routines and data in loadable executive images can be accessed.

3.30.5 Version Numbers and Version Checking

V5.0

Prior to VMS Version 5.0, a system version number was used to control several different ways that the system could change. These included the following:

- Location of routines or data in SYS.EXE
- Layout of data structures

System Manager Release Notes

3.30 Modular Executive Notes

- Details of the interface to a routine

Even though the modular executive guarantees that all transfer vectors and global data cells within the base image are fixed, privileged code still needs relinking when the system changes in one of the three ways previously mentioned.

In addition to the overall system version number, the modular executive has several version numbers, one for each functional component of the Executive. Each symbol in the base image has a small set of version numbers associated with it. When a privileged image is linked against the system symbol table (SYS.STB), version numbers associated with all routines referenced by the image are recorded in the image header. Version numbers associated with routines not referenced by this image are not recorded. Thus, the version numbers recorded in the image header provide a complete description of dependencies of this image on the set of routines in the base image.

For major releases of the VMS operating system after Version 5.0, privileged images need not be relinked for every major release of the VMS operating system. A privileged image needs to be relinked against the new system symbol table only if a functional component on which the image is dependent contains an incompatible change (in data structures or routine interfaces) from the previous release.

For example, a user-written device driver contains references to various I/O routines; therefore, the version number of the I/O component in the system symbol table against which this driver is linked is recorded in the image header of the driver. Changes to other functional components of the modular executive for example, memory management will not likely affect this device driver. Therefore, this driver need not be relinked if a subsequent release of the VMS operating system contains extensive changes to the memory management component and no changes to the I/O component.

During the system bootstrap, the secondary bootstrap program (SYSBOOT) and the system initialization code perform checks on the version of the Executive loaded images and other images loaded into the system space against the base image. If an incompatibility in the version numbers is detected, the image is rejected and the bootstrap fails.

When loading a device driver, SYSGEN checks the version numbers of the driver against the base image. If an incompatibility in the version numbers is detected, the driver is not loaded.

The image activator and the Install Utility also perform version checks. When a mismatch between the set of version numbers of a privileged image with the base image is detected, the CMKRNL and CMEXEC privileges are removed, but the activation (or making the image into a known file, in the case of INSTALL) continues.

System Manager Release Notes

3.31 Monitor Utility Notes

3.31 Monitor Utility Notes

The release notes in this section pertain to the Monitor Utility.

3.31.1 Error in Display

V5.0

If the number of free packets on the SRP, IRP, or LRP lists exceeds 500, the Monitor Utility displays asterisks in those fields rather than the actual data. This affects both the POOL and the DECNET classes.

MONITOR must hold exclusive access to each list while counting free packets. Holding exclusive access for a significant length of time can have serious side effects. This change reduces the amount of time that MONITOR holds exclusive access to these lists.

3.31.2 I/O and RMS Problem Corrected

V5.4

In previous releases of VMS, a problem existed in the Monitor Utility that prohibited the gathering of I/O and RMS classes of data simultaneously. This problem has been corrected; the Mount Utility now allows I/O and RMS classes to be gathered simultaneously.

3.31.3 MONITOR CLUSTER Command Notes

The following release notes pertain to the MONITOR CLUSTER command.

3.31.3.1 Error Messages When Using MONITOR CLUSTER

V5.4

The MONITOR CLUSTER command may return the following error message:

```
-MONITOR-W-NODEINIERR, error during node initialization
%MONITOR-I-CONT, continuing...
%VPM-W-NOCONNECT, Unable to connect to remote node node-name
```

This error message can occur because the monitoring node is unable to collect the required data from the remote nodes. Possible causes for this condition are:

- The DECnet account being used is not set up properly.
There must either be a user specification associated with the VAX Performance Management (VPM) object or a nonprivileged user specification associated with the executor database for each node.

- The maximum number of DECnet links is exceeded.

The executor database on each system (especially the system requesting remote MONITOR data) must have a sufficient number of links available to establish connections to the remote VAXcluster nodes. Each remote node must have a free link, and the requesting system must have one free link for each node in the cluster. You can modify the number of links by using the NCP utility.

- Process quotas are set too low.

The process requesting MONITOR data must have sufficient process quotas to monitor nodes in a cluster. Three critical quotas are as follows (these calculations are not exact and may need tuning):

- ASTLM: Set to at least three times the number of nodes to be monitored, plus 10.
- FILLM: Set to at least two times the number of nodes to be monitored.
- JTQUOTA: Set to 2048; sufficient for most configurations.

3.31.3.2 Using MONITOR CLUSTER Might Introduce a Network Load

V5.4

Using the MONITOR CLUSTER command in a large VAXcluster configuration might introduce a noticeable network I/O load.

The VMS Monitor Utility uses DECnet to communicate between VAXcluster nodes. When a small number of nodes are involved, this load is negligible. However, in large VAXcluster configurations, MONITOR CLUSTER might create a noticeable load.

3.31.4 RMS Bucket and Multibucket Split Rates Invalid

V5.0

When you enter the MONITOR command, RMS/ITEM=LOCKING, MONITOR displays RMS bucket and multibucket split rates. However, because the counters are not maintained properly in RMS, MONITOR always displays a rate of zero for these items. This will be corrected in a future release of the VMS operating system.

3.32 OPCOM Changes

V5.2

Effective with VMS Version 5.2, there are several changes in the way the operator communications manager (OPCOM) works. These changes are:

- Operator request numbers in a cluster begin at 1 and increase by 1 each time a request is queued. They are not reset to 1 unless the entire cluster is shut down.
- The operator log file can be placed on any disk.
- The operator log file can be enabled or disabled for specific operator classes.
- Operator log files are no longer created on satellite nodes in a cluster by default.
- By defining logical names in the command procedure SYS\$MANAGER:SYLOGICALS.COM, the system manager can override the defaults for:
 - Whether an operator is enabled on OPA0:
 - Which operator classes the operator on OPA0: controls
 - Whether an operator log file is opened
 - Which operator classes are recorded in the log file

System Manager Release Notes

3.32 OPCOM Changes

- Location and name of the operator log file
- The number of messages sent by OPCOM has been greatly reduced, and the overhead of the OPCOM process is similarly reduced.

3.32.1 Log File Operator Classes

V5.2

The DCL command REPLY/LOG has been enhanced to allow individual operator classes to be selected for inclusion in the log file. When the command REPLY/LOG/ENABLE is used, the classes listed on the /ENABLE qualifier are added to the set currently saved in the log file. If no log file is open when the REPLY/LOG/ENABLE command is used, the log file is opened with the specified classes.

Similarly, when the command REPLY/LOG/DISABLE is used, the classes listed on the /DISABLE qualifier are removed from the set currently saved in the log file. If the REPLY/LOG/DISABLE command removes all operator classes, the log file is closed.

The new feature to allow specification of classes for the log file is controlled by the commands REPLY/LOG/ENABLE=*(list-of-classes)* or REPLY/LOG/DISABLE=*(list-of-classes)*. When /LOG is added to /ENABLE or /DISABLE, the classes refer to the log file rather than the current terminal. For more information, see the commands REPLY/LOG and REPLY/ENABLE and REPLY/DISABLE in the *VMS DCL Dictionary*.

3.32.2 OPCOM Default States

V5.2

OPCOM has the following default states:

- For all systems except workstations in a VAXcluster configuration:
 - OPA0: is enabled for all classes.
 - The log file SYS\$MANAGER:OPERATOR.LOG is opened for all classes.
- For workstations in a VAXcluster configuration, even though the OPCOM process is running:
 - OPA0: is not enabled.
 - No log file is opened.

3.32.3 Overriding the OPCOM Default States

V5.2

To override the OPCOM default states, define the following system logical names in the command procedure SYS\$MANAGER:SYLOGICALS.COM:

- OPC\$OPA0_ENABLE
If defined to be true, OPA0: is enabled as an operator. If defined to be false, OPA0: is not enabled as an operator. DCL considers any string beginning with *T* or *Y* or any odd integer to be true, all other values are false.

System Manager Release Notes

3.32 OPCOM Changes

- **OPC\$OPA0_CLASSES**

This logical name defines the operator classes to be enabled on OPA0:. The logical name can be a search list of the allowed classes, a list of classes, or a combination of the two, for example:

```
$ DEFINE/SYSTEM OPC$OPA0_CLASSES CENTRAL,DISKS,TAPE
$ DEFINE/SYSTEM OPC$OPA0_CLASSES "CENTRAL,DISKS,TAPE"
$ DEFINE/SYSTEM OPC$OPA0_CLASSES "CENTRAL,DISKS",TAPE
```

Note that OPC\$OPA0_CLASSES can be defined even if OPC\$OPA0_ENABLE is not defined. In that case, the classes are used for any operators that are enabled, but the default is used to determine whether or not to enable the operator.

- **OPC\$LOGFILE_ENABLE**

If defined to be true, an operator log file is opened. If defined to be false, no log file is opened.

- **OPC\$LOGFILE_CLASSES**

This logical name defines the operator classes to be enabled for the log file. The logical name can be a search list of the allowed classes, a comma-separated list, or a combination of the two.

Note that OPC\$LOGFILE_CLASSES can be defined even if OPC\$LOGFILE_ENABLE is not defined. In that case, the classes are used for any log files that are opened, but the default is used to determine whether or not to open the log file.

- **OPC\$LOGFILE_NAME**

This logical name supplies information to be used in conjunction with the default name SY\$MANAGER:OPERATOR.LOG to define the name of the log file. If the log file is directed to a disk other than the system disk, commands to mount that disk should be included in the command procedure SYLOGICALS.COM.

Example SYLOGICALS.COM

The following example shows how to disable the SECURITY class messages from being displayed on OPA0:, and also how to disable SECURITY class messages from being saved in the operator log file.

```
$ DEFINE/SYSTEM OPC$OPA0_CLASSES CENTRAL,PRINTER,TAPES,
DISKS,DEVICES,CARDS,NETWORK,CLUSTER,LICENSE,
OPER1,OPER2,OPER3,OPER4,OPER5,OPER6,OPER7,
OPER8,OPER9,OPER10,OPER11,OPER12

$ DEFINE/SYSTEM OPC$LOGFILE_CLASSES CENTRAL,PRINTER,TAPES,
DISKS,DEVICES,CARDS,NETWORK,CLUSTER,LICENSE,
OPER1,OPER2,OPER3,OPER4,OPER5,OPER6,OPER7,
OPER8,OPER9,OPER10,OPER11,OPER12
```

Note that since OPC\$OPA0_ENABLE or OPC\$LOGFILE_ENABLE was not defined, the defaults will determine whether the OPA0 operator is enabled and whether the logfile is opened.

System Manager Release Notes

3.32 OPCOM Changes

3.32.4 Removing Old Reply Commands — Requirement

V5.2

Prior to VMS Version 5.2, to override the default OPCOM states it was necessary to place REPLY commands in system startup files like SYS\$MANAGER:SYSTARTUP_V5.COM. This was done by defining SYS\$COMMAND to an enabled operator terminal, and then issuing the REPLY command. For example, to disable the SECURITY operator class on OPA0: you could enter:

```
$ DEFINE/USER SYS$COMMAND OPA0:  
$ REPLY/DISABLE=SECURITY
```

While this technique is still permitted, Digital recommends that all commands of this nature be removed from system startup files, and that the logical names in SYLOGICALS.COM be used to define the desired operator state.

3.33 Printer Execution Queue Problem Corrected

V5.3-1

Prior to VMS Version 5.3-1, a problem existed with printer execution queues. If the system manager mounted a new form with a different stock type while a job was executing, conflicting information could be produced in the database. These conflicts occasionally resolved themselves, while at other times they lead to database corruption. This problem has now been corrected.

3.34 Pseudoterminal Driver

V5.1

VMS Version 5.1 includes a pseudoterminal driver consisting of components PYDRIVER and TWDRIVER. This pseudoterminal driver is intended for the exclusive use of DECwindows; any other use of the driver is unsupported.

V5.4

VMS Version 5.4 includes a new pseudoterminal driver consisting of components FTDRIVER and a set of system services, PTD\$SERVICES_SHR.EXE. For more information on the new pseudoterminal driver, see Chapter 9 in the *VMS I/O User's Reference Manual: Part I*.

A future release of VMS will discontinue the use of the VMS Version 5.1 PYDRIVER/TWDRIVER, at which time the Version 5.1 images will no longer be shipped as part of VMS. To prepare for a future VMS release, Digital strongly recommends that you convert any code that uses the PYDRIVER/TWDRIVER to the new pseudoterminal driver.

3.35 RQDX3 Controller Notes

The release notes in this section pertain to the RQDX3 controller.

3.35.1 Device Unit Number Changed

V5.0

An error involving served satellite disks that change the device unit number of a disk by setting the high bit has been discovered. When this occurs, the disk cannot be accessed using the original device unit number. However, you can access the disk using the new unit number (old unit number + 128).

This problem occurs when the following conditions are present:

- Disks are accessed through the MSCP server
- Very large files are created (for example, creating paging and swapping files that are larger than 20,000 blocks)
- Highwater marking is present

You can correct this problem with either of the following solutions:

- 1 Do not create large files from a remote node when highwater marking is present. Instead, you should do this on the local node. If you are creating paging and swapping files, create them using their minimal sizes and boot them on your local node. Then, you can make them larger on the local node.
- 2 Do not use highwater marking when creating large files over a served network path. Turn off the highwater marking and then create the file. This will prevent sending of the MSCP command ERASE. Thus, the file will be allocated but not zeroed.

3.35.2 Frequent Controller Resets — Restriction

V5.0

If you are using RQDX3 controllers on a system that serves disks in a local area VAXcluster or a mixed interconnect cluster, and the RQDX3 controller does not contain a microcode revision level of 3.0 or later, you may see frequent controller resets. If your error log shows frequent controller resets during satellite booting, you should contact your local Digital Customer Services representative to obtain the latest microcode.

You can determine the controller type and microcode revision level by entering the command `ANALYZE/ERROR_LOG`.

3.36 Security Features — Notes

The release notes in this section pertain to new or changed system security features.

3.36.1 Department of Defense (DoD) Erase Pattern Corrected

V5.2

VMS Version 4.0 introduced the \$ERAPAT system service to allow sites to generate their own erase patterns. Digital supplies a sample MACRO source file that contains the Department of Defense (DoD) erase patterns for memory, disk, and tape. Included in this file are instructions on how to assemble and link the module and how to install

System Manager Release Notes

3.36 Security Features — Notes

the resulting ERAPATLOA.EXE system loadable image in the directory SYS\$LOADABLE_IMAGES.

VMS Version 5.2 contains several corrections that correct all problems relating to using a loadable, nonzero erase pattern. If you encounter problems with this service, submit a Software Performance Report (SPR).

3.36.2 **NETCONFIG.COM Security Enhancements**

V5.2

In VMS Version 5.2, the DECnet network configuration command procedure, NETCONFIG.COM, was enhanced to provide several options for limiting default access to your system. A new command procedure for existing networked systems, NETCONFIG_UPDATE.COM, was created for the same purpose.

Previously, NETCONFIG.COM created one default account named DECNET. That account provided default access to all Digital-supplied objects and user-written applications that were not restricted by other forms of access control, such as proxy accounts and access control strings. That type of default access is appropriate only for systems with very low security requirements.

Now, in place of the default account DECNET, individual accounts for the following Digital-supplied objects can be created:

- MAIL
- File access listener (FAL)
- PHONE
- Network management listener (NML)
- Loopback mirror (MIRROR)
- VMS Performance Monitor (VPM)

These accounts restrict default access to their respective objects. Therefore, a system manager can enable default access for those objects that are appropriate for the system and the network. In addition, logs can be produced for these accounts so that the usage of these objects can be monitored.

Previously, creating these accounts required several commands for each. Now, using NETCONFIG.COM (or NETCONFIG_UPDATE.COM), you can create an account for a Digital-supplied object by simply responding YES to the respective prompt.

For more information about the new options, refer to the *VMS Version 5.4 New Features Manual*. For more information about network security, refer to the *Guide to VMS System Security*.

3.36.3 OPCOM on Nonclustered MicroVAX Systems

V5.2

Beginning with VMS Version 5.2, OPCOM is started by default on nonclustered MicroVAX systems and all other VMS systems, regardless of hardware configuration. The current implementation of security auditing requires that OPCOM be present.

If your installation does not require security auditing, OPCOM (and the AUDIT_SERVER) can be disabled through the SYSMAN utility.

Note: Sites that want security auditing should not disable OPCOM, since the current implementation of security auditing requires that OPCOM be present. Digital intends to remove this requirement in a future release of VMS.

3.36.4 Passwords — New Security Alarms

V5.4

Starting with VMS Version 5.4, a site security administrator (SSA) can screen new passwords to ensure that they comply with a site-specific password policy.

Installing and enabling a site-specific password policy image requires both SYSPRV and CMKRNL privileges. In addition, if INSTALL and SYSPRV file-access auditing are enabled, multiple security alarms are generated when the shareable image is installed and the change to the SYSGEN parameter is noted on the operator console. The shareable image contains two global routines that are called by the VMS Set Password Utility whenever a user changes a password with the SET PASSWORD command.

Caution: The two global routines allow an SSA to obtain both the proposed password characters and the equivalent quadword hash value. Therefore, unauthorized use of the global routines by a malicious privileged user compromises your system's security.

Digital recommends that you place the following security alarm access control list entries (ACEs) on the shareable image and its parent directory:

```
$ SET ACL/ACL=(ALARM=SECURITY, ACCESS=WRITE+CONTROL+DELETE+SUCCESS+FAILURE) -
_ $ SYS$LIBRARY:VMS$PASSWORD_POLICY.EXE
$ SET ACL/ACL=(ALARM=SECURITY, ACCESS=WRITE+CONTROL+SUCCESS+FAILURE) -
_ $ SYS$COMMON:[000000]SYSLIB.DIR
```

You must also enable access control list (ACL) alarms using the following command:

```
$ SET AUDIT/ALARM/ENABLE=ACL
```

Once in place, the ACL alarms catch all attempts to replace or to modify the VMS\$PASSWORD_POLICY image.

3.36.5 Reestablishing Security Environment

V5.2

With VMS Version 5.2 and subsequent versions, an upgrade provides new files and directories under [VMS\$COMMON...]. If you had any special protections and ACLs before the upgrade, you need to reapply them to reestablish your previous security environment.

System Manager Release Notes

3.36 Security Features — Notes

3.36.6 Security Audit Alarm Settings Preserved Between System Boots

V5.2

In prior versions of VMS, the classes of security-auditing alarm events had to be set each time a system was booted (typically in the site-specific startup command procedure SYSTARTUP_V5). Beginning with VMS Version 5.2, the security alarm settings are preserved between boots in the permanent audit server database SYS\$MANAGER:AUDIT_SERVER.DAT.

Following the upgrade to VMS Version 5.2 or subsequent versions, you should remove any SET AUDIT/ALARM commands from your site-specific startup command procedure.

3.36.7 Security-Auditing Failure Mode Settings Preserved Across Initializations

V5.4

Prior to VMS Version 5.4, the site security administrator was required to reset the security-auditing failure mode each time the system was initialized. Now, the security-auditing failure mode setting is preserved across system initializations.

Site-security administrators should remove any existing SET AUDIT /FAILURE_MODE commands from system-specific command procedures, such as SYSTARTUP_V5.COM.

For more information on security auditing, see the *Guide to VMS System Security*.

3.36.8 Suppressing Duplicate Logging of Security Alarms by OPCOM

V5.2

VMS Version 5.2 and subsequent versions differentiate between a security alarm and a security audit in the security auditing software.

A **security alarm** is a real-time event that is broadcast to security operator terminals by the operator communication manager (OPCOM). Security alarms are not necessarily recorded onto permanent media (disk or tape), although the site-security administrator may choose to do so.

A **security audit** is a security event that is logged by the audit server process (AUDIT_SERVER) directly to the system security audit log file (SYS\$MANAGER:SECURITY_AUDIT.AUDIT\$JOURNAL). Security audits are never displayed on security operator terminals.

In a future release of VMS, the site-security administrator will be able to control whether a given system-security event generates an alarm, an audit, or both.

For compatibility with previous releases, VMS Version 5.2 and subsequent versions currently propagate all system-security events as both a system alarm and a system audit. Alarms are broadcast to all SECURITY class operators, and audits are logged in the system security audit journal file.

By default, OPCOM logs all SECURITY class messages in the operator log file, as in earlier releases. Because these entries now duplicate the entries in the system security audit log, to conserve disk space you might

want to disable the SECURITY class in the operator log file by issuing the command:

```
$ REPLY/LOG/DISABLE=SECURITY
```

See Section 3.32.3 for the recommended method of disabling an operator class each time the system is initialized.

Note: Because the audit server process directs important messages to both the CENTRAL and SECURITY class operators, disabling the SECURITY class will not prevent the receipt of critical information (as in previous releases).

3.36.9 SYSECURITY.COM Command Procedure — New Site-Specific Configuration File

V5.2

The installation and upgrade procedures create an empty SYSECURITY.COM command procedure, which is run prior to starting up the security-auditing server process. If you wish to direct your system security-audit journal file SECURITY_AUDIT.AUDIT\$JOURNAL or audit-server database AUDIT_SERVER.DAT in the SYS\$MANAGER directory to a disk other than the system disk, specify the command to mount the alternate disk in this file. This ensures that the alternate disk is mounted before the audit server process is started.

In addition to using the procedure SYSECURITY.COM to mount disks, you can use it to define the system logical name AUDIT_SERVER (to relocate the audit-server database file) or to define system logical names needed to resolve either the system security audit journal or system archive file's destination. For example, the following lines in the procedure SYSECURITY.COM mount the disk \$254\$DUA118 and redirect the audit server permanent database to the alternate volume:

```
$ if .not. f$getdvi("$254$dua118","mnt") -  
    then mount/system $254$dua118 audit audit$ /norebuild  
$ define/system/exec audit_server audit$:[audit]audit_server.dat
```

V5.4

Because you invoke the SYSECURITY.COM command procedure before you start the audit server, do not place any SET AUDIT commands in this file.

3.36.10 SYSUAF Template File — Change

V5.2

The file SYS\$SYSTEM:SYSUAF.TEMPLATE is used by the VMS installation procedure to initially create the System User Authorization File SYS\$SYSTEM:SYSUAF.DAT. (You can also use the template file to create a new User Authorization File (UAF) file on your system, using the DCL command COPY SYS\$SYSTEM:SYSUAF.TEMPLATE SYS\$SYSTEM:SYSUAF.DAT.) Beginning with VMS Version 5.2, all accounts in the SYSUAF template file, except for the SYSTEM account, are disabled when shipped, by setting the flag /FLAG=DISUSER.

System Manager Release Notes

3.36 Security Features — Notes

Additionally, all of the account passwords in the template file are set as expired, and the password lifetime on the DEFAULT account has been lowered from 180 days to 90 days. Digital recommends a maximum password lifetime of 90 days for unprivileged accounts, and a maximum lifetime of 30 days for privileged accounts.

3.36.11 User Authorization File (UAF) Notes

The following sections pertain to the User Authorization File (UAF).

3.36.11.1 Captive Accounts — Batch and Network Restrictions Removed

V5.4

Since VMS Version 5.2, accounts with the UAF flag CAPTIVE set have not been able to use the DCL command SUBMIT to create batch jobs or be the targets of network connections. This restriction was caused by the restrictions imposed in Version 5.2 to prevent users from obtaining unrestricted access to DCL while in a captive account.

These restrictions were relaxed in VMS Version 5.3-2. Captive accounts may now create batch jobs normally and be the target of network connections without modifications to NETSERVER.COM, SYLOGIN.COM, or LOGIN.COM.

Site security administrators who want to disable batch and network access for captive accounts should use the AUTHORIZE restricted-access qualifiers, /NOBATCH and /NONETWORK, to disable the job modes.

3.36.11.2 Captive Accounts — Security and Application PRINT Commands

V5.4

Because of the restrictions that were in place between VMS Versions 5.2 and 5.3-2 for batch jobs (see Section 3.36.11.1), some captive-account environments might need to be modified if they provide access to applications that support an internal PRINT command.

Unless an application ensures that the user has specified a PRINT queue as the target of a PRINT command, applications can often be tricked into submitting a print job to a batch queue. This incorrect submittal results in the execution of the print job by the batch subsystem as a normal batch job. Depending on the application, the mechanism that allows the incorrect submittal can be exploited by users to execute arbitrary DCL commands in captive accounts.

Applications that do not conform to your site's captive policy (for example, they do not use the PRINT command correctly) should be removed from the captive user's environment, or additional restrictions should be added to the user's account, such as adding the /NOBATCH qualifier.

Digital recommends that you review the set of applications that are available to captive accounts, to ensure that their behavior is consistent with your site's security policy for captive accounts.

V5.2

3.36.11.3 Changes to the CAPTIVE Flag

The User Authorization File (UAF) flag CAPTIVE has always been intended for accounts that perform individual (often privileged) functions (like backing up and restoring files) or for accounts tied to a menu system (often referred to as *turn-key* accounts). Typically, these accounts operate in a restricted environment and do not allow the user complete access to the command language interpreter (DCL).

However, the security of a captive account—an account with the CAPTIVE flag set—has depended on the captive command procedure under which the account runs. Prior to VMS Version 5.2, a captive command procedure was only truly captive if the author of the command procedure exactly followed the guidelines in the *Guide to VMS System Security*.

UAF Flag CAPTIVE — New Interpretation

Beginning with VMS Version 5.2, the UAF flag CAPTIVE has been enhanced to make writing captive command procedures easier and to increase the security of systems using captive command procedures that do not follow the guidelines in the *Guide to VMS System Security* exactly. The enhancements include the following:

- Accounts with the CAPTIVE flag set no longer have direct access to DCL. Command procedures that terminate to DCL (for example, as a result of an unhandled error or of pressing Ctrl/Y) now result in the error message CAPTINT and deletion of the process under which the procedure runs.
- Additionally, the INQUIRE command has been disabled for accounts with the CAPTIVE flag set. You must use the READ/PROMPT command instead. Using the INQUIRE command in a captive command procedure produces the error CAPTINQ which, if unhandled by a previous ON declaration, results in the error CAPTINT and deletion of the process.

For a complete list of restrictions imposed by the CAPTIVE flag, see the *Guide to VMS System Security*.

New UAF Flag RESTRICTED

Digital recognizes that these changes in the CAPTIVE flag may harm existing captive command procedures that depend on the behavior prior to Version 5.2 (see the subsection “Possible Incompatibilities with New Interpretation of CAPTIVE Flag”). Digital also recognizes that the previous behavior does have value in some situations – namely, to force the execution of a set of command procedures, after which the user is allowed normal access to DCL.

Therefore, the security restrictions formerly denoted by the CAPTIVE flag have been moved to a new UAF flag called RESTRICTED. Accounts in which the RESTRICTED flag is set obey all of the restrictions that were formerly implied by CAPTIVE. Future security enhancements in the area of captive accounts will most likely be tied to the CAPTIVE flag only.

Note: In a future release of VMS, Digital also intends to remove the SPAWN restrictions from the RESTRICTED flag. At such time,

System Manager Release Notes

3.36 Security Features — Notes

VMS will not treat **RESTRICTED** accounts differently than normal accounts once the login sequence has been completed. Customer written software should not use the **RESTRICTED** flag to prevent access to either the **SPAWN** command or the **LIB\$SPAWN** RTL routine.

STARLET Symbol **UAI\$V_CAPTIVE** — Value Change

VMS Version 4.4 introduced the system services **\$GETUAI** and **\$SETUAI** to provide a system service interface to the System User Authorization File (SYSUAF). These services allow privileged routines to retrieve and to modify information contained in any user's authorization record. The interface to these services uses the **\$UAIDEF** symbols defined in the file **STARLET.MLB**.

Because of the change in interpretation of the UAF flag **CAPTIVE** in VMS Version 5.2, it has been necessary to change the value of the public symbol **UAI\$V_CAPTIVE**, for two reasons:

- To allow Version 5.2 nodes to coexist securely with Version 5.1 nodes in the rolling upgrade environment
- To ensure that under Version 5.2, UAF records marked **CAPTIVE** will take on the additional restrictions by default

UAI\$V_RESTRICTED is the new name for the bit that was formerly **UAI\$V_CAPTIVE**. The symbol **UAI\$V_CAPTIVE** still exists, but now defines a previously unused bit in the UAF flags longword. The following table shows the new values (in decimal radix) for these two symbols:

Symbol	New Value
UAI\$V_RESTRICTED	3
UAI\$M_RESTRICTED	8
UAI\$V_CAPTIVE	16
UAI\$M_CAPTIVE	65536

The new **UAI\$V_RESTRICTED** flag functions exactly as the old **CAPTIVE** flag did prior to VMS Version 5.2. The new **CAPTIVE** flag has the following additional features:

- Returning direct command to DCL is not allowed
- Use of the DCL verb **INQUIRE** is not allowed

Captive command procedures that violate either of these new restrictions cause the process to be deleted with an appropriate error message. For complete information on captive flags, see the *Guide to VMS System Security*.

Images that were linked prior to VMS Version 5.2 will continue to function normally; however, they will be manipulating the **RESTRICTED** bit (as viewed from **AUTHORIZE**, for example).

Programmers who wish to take advantage of the new behavior of **CAPTIVE** must recompile and relink from source.

System Manager Release Notes

3.36 Security Features — Notes

Many Digital language products make the STARLET library definitions available to programmers. After installing VMS Version 5.2, you will need to reinstall these language products so that they reflect the changes to the STARLET definitions. For further details, refer to the installation guides for the language products that you have installed.

Some Digital language products include their own support for STARLET in a separate environment file that is not based on the STARLET library that is shipped with VMS. As a result, these language products will not automatically recognize the change in the definition of UAI\$V_CAPTIVE, even if they are reinstalled.

For these layered products, you will have to override the definition of UAI\$V_CAPTIVE (using the value shown in the preceding table) until the next release of the language product that includes support for the VMS Version 5.2 STARLET environment.

Possible Incompatibilities with New Interpretation of CAPTIVE Flag

During the initial stages of the upgrade, all user accounts which were previously marked CAPTIVE will be modified to use the new UAF flag RESTRICTED instead.

The upgrade procedure also scans the UAF and marks all accounts which are then set RESTRICTED to also be set CAPTIVE. This ensures that all accounts which were previously marked CAPTIVE are fully secured under VMS Version 5.2.

However, as a result of this change, those accounts that specifically rely on the old behavior of CAPTIVE will no longer function correctly.

Note: Network server accounts that are defined in the permanent network object database (NETOBJECT.DAT) prior to the VMS Version 5.2 upgrade will not be modified by the upgrade and should continue to work without modification.

Problems specifically related to the new interpretation of CAPTIVE can be diagnosed by enabling the PROCESS accounting class. These problems typically manifest themselves as a "Network Partner Exited" message on the node that initiates a DECnet connection and as either a CAPTINT or CAPTINQ error on the remote node.

You can use the following command to locate all process termination records due to either a CAPTINT or CAPTINQ error:

```
$ ACCOUNTING/TYPE=PROCESS/STATUS=(3895A,38952)/FULL
```

Affected accounts can be modified to restore the old behavior by executing the following commands from a suitably privileged account:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> MODIFY FOOBAR$SERVER/FLAG=(NOCAPTIVE,RESTRICTED)
UAF> EXIT
```

Digital strongly recommends that the site-security administrator carefully review the relevant sections in the *Guide to VMS System Security* before clearing the CAPTIVE flag on any account. Indiscriminately clearing the CAPTIVE flag could compromise the security of the captive account.

System Manager Release Notes

3.36 Security Features — Notes

-
- V5.2** **3.36.11.4 DISIMAGE Flag**
- Many sites effect their security by constructing alternate DCL command tables. Often these tables are used in conjunction with captive accounts to restrict the verbs (and images) you can access.
- The User Authorization File (UAF) flag DISIMAGE prevents the execution of arbitrary user-written images by disabling the RUN and MCR verbs and the foreign command mechanism in DCL. Accounts in which this flag is set can only execute commands defined in their command table CLITABLES.
- The DISIMAGE flag is intended to be used in combination with the RESTRICTED or DEFCLI UAF flags, which prevent the user from selecting an alternate CLI or CLITABLES using the /CLI or /TABLES qualifiers at login.
- Note:** If you, as a restricted user, retain access to the SET COMMAND verb, you can still run arbitrary images by using the Command Definition Utility to create your own verbs. Site security administrators must take this into account when creating captive environments. See the *Guide to VMS System Security* for more information on the DISIMAGE flag.
-
- V5.4** **3.36.11.5 RESTRICTED Accounts — Incompatibility Problems Corrected**
- With VMS Version 5.2, a number of compatibility problems were reported for the UAF flag RESTRICTED and the previous (VMS versions prior to Version 5.2) UAF flag CAPTIVE.
- These compatibility problems were corrected in VMS Version 5.3-1.
-
- V5.2** **3.36.11.6 UAF Record Length Enforcement**
- Beginning with VMS Version 5.2, checks have been added to the \$GETUAI and \$SETUAI system services to detect UAF records that are shorter than the UAF minimum record length (UAF\$K_FIXED). If the requested record is less than the minimum value, the services return an SS\$_ACCVIO error code.
- System programmers should ensure that any user-written software that accesses the UAF directly correctly maintains the minimum UAF record length.
- Note:** The format of the UAF record and the way in which the system modifies it is subject to change in future versions of VMS. Digital does not support direct access to the UAF.
-
- V5.2** **3.36.11.7 UAF Template File Changes**
- With VMS Version 5.2, the following changes have been made to the UAF template file (SYS\$SYSTEM:SYSUAF.TEMPLATE) for VMS Version 5.2:
- The UAF parameter PWDLIFETIME for the DEFAULT account has been lowered from 180 days to 90 days
 - The UAF parameter PWDLIFETIME for the accounts FIELD, SYSTEM, SYSTEST, and SYSTEST_CLIG has been lowered from 90 days to 30 days

- The DISUSER flag has been set for the accounts DEFAULT, FIELD, SYSTEST, and SYSTEST_CLIG

Because the DEFAULT account serves as a template for the ADD command in AUTHORIZE, site-security administrators may wish to clear the DISUSER flag on the DEFAULT account using the following commands:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> MODIFY DEFAULT/FLAG=NODISUSER
```

If you do not clear the DISUSER flag on the DEFAULT account, accounts created by some layered product installations may not function correctly. Depending upon the layered product installation procedures, this may in turn cause the Installation Verification Procedure (IVP) for the product to fail. If this occurs, clear the DISUSER flag on the DEFAULT account and reinstall the layered product.

3.37 Security Review Recommended

V5.2

Digital strongly recommends that all site security administrators take the following steps to improve security on existing VMS systems:

- Disable or remove all of the Digital default accounts, except for SYSTEM, in any active SYSUAF.DAT files, unless you have an explicit need for these accounts. The default accounts are FIELD, SYSTEST, and SYSTEST_CLIG. Previous versions of VMS also included accounts named USER and USERP, which you should remove if they are present on your system. This is critically important if you have used the VMSKITBLD.COM command procedure to build alternate system disks, which will have a SYSUAF.DAT file containing the default Digital username/password combinations.
- Seriously consider using the password generator for all privileged accounts at sites with medium to high security needs.

Network security managers should also consider the following additional steps:

- Review all network proxies and eliminate, if at all possible, any proxies into privileged accounts.
- Remove the default DECnet account and substitute separate accounts for each network object required for a particular node. Use generated passwords for these accounts.
- Review the *Guide to VMS System Security*. This manual has been significantly enhanced to describe all of the new features present in VMS Version 5.2.

System Manager Release Notes

3.38 SET HOST (CTDRIVER/REMACP/RTPAD) Notes

3.38 SET HOST (CTDRIVER/REMACP/RTPAD) Notes

The notes in this section describe some of the behaviors of the SET HOST facility prior to VMS Version 5.2 and some additional features that are provided with VMS Version 5.2 and subsequent versions.

3.38.1 \$CANCEL is Asynchronous

V5.2

The \$CANCEL system service performs an asynchronous cancel operation. This means that the application *must* wait for each I/O operation issued to the driver to complete prior to checking the status for that operation.

Prior to VMS Version 5.0, the VMS terminal driver performed synchronous cancel operations. This was a result of the way VMS was scheduled on asymmetric multiprocessing (ASMP) machines and uniprocessors, not a design feature of the VMS terminal driver. Starting with VMS Version 5.0, in an SMP environment the VMS terminal driver completes these cancel operations asynchronously.

CTDRIVER has always completed the cancel operations asynchronously. If a read request is pending and CTDRIVER receives a cancel request, CTDRIVER queues a message to be sent to the remote system, telling it to cancel the read request. Control then returns to the application calling \$CANCEL. Eventually, CTDRIVER sends the message to the remote system. The remote system then sends any data received prior to the cancel request back to CTDRIVER. Once CTDRIVER receives this data, the pending read request can be canceled. For slow networks, this could take *seconds* before the pending read request is actually canceled.

3.38.2 CTDRIVER Enforces SETMODE/SENSEMODE Buffer Size

V5.2

CTDRIVER now enforces the SETMODE/SENSEMODE buffer size. The valid buffer sizes are either 8 or 12 bytes with the default being 8 bytes. If any other buffer values are specified then CTDRIVER returns the status code SS\$_BADPARAM.

3.38.3 CTDRIVER's Output Buffering

V5.2

The SET HOST facility behaves differently from the VMS terminal driver in that it buffers output data from the program that is executing. This occasionally causes a perception problem for the user when the program is aborted with a Ctrl/C, a Ctrl/Y, or an out-of-band abort character. The user expects the program to abort immediately and the display to stop immediately. The components of the SET HOST facility try to preserve the feel of the VMS terminal driver by stopping the display as soon as the abort character is entered, which can discard a significant amount of program output.

When running between two VMS systems, the SET HOST facility is made up primarily of two elements: RTPAD (local VAX node), and CTDRIVER (remote VAX node). Both elements perform output buffering to enhance performance when using wide area networks. CTDRIVER performs the

System Manager Release Notes

3.38 SET HOST (CTDRIVER/REMACP/RTPAD) Notes

initial buffering, queues the buffers to be output later, and returns a successful write status. The user sees the results of the executing program only as the buffers of output data are displayed. This buffering causes a perception problem for the user when large sections of output are displayed on the local terminal, since the output displayed on the user's terminal is typically far behind the output generated by the application. The user is led to believe that the application is executing at the exact point that is producing the current display, when in reality the remote program can have completed execution.

The delay between executing an application and displaying its output leads to several anomalies in the effects of Ctrl/C, Ctrl/Y, and out-of-band abort characters.

3.38.3.1 Captive Command Procedures and Ctrl/Y

V5.2

A side effect of buffering by CTDRIVER is the perception of how Ctrl/Y works when using the SET HOST facility. Since CTDRIVER and RTPAD are trying to emulate the VMS terminal driver, the current read operation and all pending write operations are aborted when the user enters Ctrl/Y. However, the pending write operations also include all the buffered output that should have been displayed before the Ctrl/Y was entered, but due to the buffering was not.

The effect of the buffering can be especially confusing if a Ctrl/Y is entered when a captive command procedure is executing. During execution of captive command procedures, DCL has a Ctrl/Y pending. When this AST is delivered, DCL only reenables it; no other action is performed. In that case, if the program being executed only performs output, it appears that the program was aborted by the Ctrl/Y. Actually, the program completed execution before the Ctrl/Y was entered, and the Ctrl/Y merely discarded all the buffered output.

3.38.3.2 Ctrl/C, Ctrl/Y, and Out-of-Band Abort Character Processing

V5.2

Prior to VMS Version 5.2, if an application had a read operation pending and had queued a Ctrl/C, Ctrl/Y, or out-of-band abort character AST, it was possible for the application to queue additional read requests unknowingly after receiving the abort character AST, but before the read operation was aborted. This could occur because the CTERM protocol delivered two separate messages to CTDRIVER, the first describing the abort character and the second describing the aborted read operation. These events were sent to the application as soon as they were received by CTDRIVER, but they could be separated by several seconds when they reached the application, due to network delays in the CTERM protocol. If the application assumed that the read operation had completed when it received the abort character AST and responded by requesting an additional read, then multiple read requests would be outstanding, creating confusing displays and responses for the user.

Note: Such an assumption is really an error in the application program.

This behavior has now been changed. CTDRIVER does not deliver the abort character AST until it receives the second message; CTDRIVER then delivers both events, the abort character AST followed by the read completion event. The read status should be set very shortly after the

System Manager Release Notes

3.38 SET HOST (CTDRIVER/REMACP/RTPAD) Notes

abort-character AST is delivered to the application, and before the application has time to issue a new read request. Note, however, that these are still two asynchronous events, and that the application must still synchronize with the completing read operation.

V5.2 3.38.3.3 Extra Input Prompt Displayed Following Ctrl/C, Ctrl/Y, or an Out-of-Band Abort Character

After you enter a control character that causes input and output to be aborted, the system may display more than one input prompt. This occurs when you are connected to a system running a VMS version prior to VMS Version 5.2, because the older CTERM protocol does not synchronize RTPAD and CTDRIVER after the abort occurs.

VMS Version 5.2 extends the CTERM protocol for VMS-to-VMS connections to allow CTDRIVER to synchronize with RTPAD before displaying any more data on the terminal. The extra read prompt is not displayed, a behavior that better emulates the VMS terminal driver.

Note: This problem is corrected only between VMS Version 5.2 and later systems. If SET HOST is used between VMS Version 5.2 and older versions of VMS, the extra read prompt is still displayed.

V5.2 3.38.3.4 Output Line Not In Sequence Following Ctrl/C, Ctrl/Y, or an Out-of-Band Abort Character

After you enter a control character that causes input and output to be aborted, it is possible to get one more line of output data. This occurs when the application program calls \$QIO (directly or indirectly through RMS or language support routines) to output data to a buffer at the same time that the control character is entered.

When CTDRIVER receives the abort character (Ctrl/C, Ctrl/Y, or an out-of-band abort character) from the network, it flushes the current output buffers and aborts any pending read operations. However, if the application program has just called \$QIO with a write operation, that \$QIO write data is still buffered, and then displayed. That data may not be the next output in sequence from the user's point of view, since all the previous output buffers in CTDRIVER were flushed and the data in them was not displayed.

When not using the SET HOST facility, the effect of an abort character on the display is different, because the VMS terminal driver does not buffer output from the application that is executing. If the application program has just called \$QIO with a write operation when the abort character is entered, then this \$QIO write data is displayed. Because all write operations are sequential and do not complete until the output is actually displayed, the additional line displayed is in sequence. Since there is no break in the data, the user normally will not notice that there is an additional line.

System Manager Release Notes

3.38 SET HOST (CTDRIVER/REMACP/RTPAD) Notes

3.38.4 Ctrl/C Processing

V5.2

Application programs that use both the Ctrl/C asynchronous system trap (AST) and a Ctrl/C out-of-band AST may receive Ctrl/Y ASTs as well when using the SET HOST facility. This occurs when the first Ctrl/C character is entered from the keyboard, following the declaration of both AST routines.

The problem arises because the command terminal protocol CTERM only supports a single function per control character. The CTERM function code for Ctrl/C can be encoded to represent either a Ctrl/C out-of-band AST or a Ctrl/C AST, but not both. If the Ctrl/C AST is declared prior to the Ctrl/C out-of-band AST, then RTPAD removes the Ctrl/C AST.

A feature in RTPAD allows you to work around the CTERM protocol limitation by always declaring the Ctrl/C out-of-band AST (IO\$_M_INCLUDE bit set) before declaring the Ctrl/C AST. This places RTPAD in the correct state to handle both ASTs, thus allowing correct input to the application program.

3.38.5 New REMACP Image and RTTLOAD Command File

V5.2

The remote I/O ancillary control process image REMACP and the command file RTTLOAD.COM have been rewritten to provide more flexibility, as described in the release notes in this section.

3.38.5.1 Running Without RTTDRIVER

V5.2

The image REMACP now determines which driver is present in the system prior to accepting the network connection. Both drivers, CTDRIVER and RTTDRIVER, are now dynamically connected to the RTA device at the time the device is created. This allows you to load only the required remote terminal drivers, which is a benefit for small memory systems.

The logical names REM\$NO_CTDRIVER and REM\$NO_RTTDRIVER allow you to specify which drivers should not be loaded into the system. Defining either of these logical names (with any value) causes the appropriate driver *not* to be loaded during the startup of the REMACP process. If both logical names are defined, neither driver is loaded and the REMACP process is not started. The default if the logical names are not defined is still the previous behavior of loading both drivers.

If for some reason the other driver is needed after the system is already running, you must deassign the corresponding logical name and then execute the RTTLOAD command file.

Additionally, the image REMACP has been modified to ensure that the driver support is present prior to accepting the network connection. This allows RTPAD (SET HOST) to fail over dynamically to the older protocol if CTDRIVER is not loaded.

System Manager Release Notes

3.38 SET HOST (CTDRIVER/REMACP/RTPAD) Notes

V5.2

3.38.5.2 Setting the Maximum Number of Remote Users

You can set the maximum number of remote users at boot time or dynamically on a running system.

Defining the Range for the SYSGEN Parameter RJOBLIM

The allowable number of remote users of a system is determined by the SYSGEN parameter RJOBLIM. If the number of remote users exceeds the value of the parameter RJOBLIM, the image REMACP stops accepting additional incoming connections. The parameter RJOBLIM can take any value within a range defined at startup by the command file RTTLOAD.COM.

The command file RTTLOAD.COM has been modified to provide an extended range for the value of the parameter RJOBLIM (the allowable number of remote users). The minimum value for RJOBLIM is 0 and the default maximum value is 255. You can raise the maximum above 255 by changing the value of RJOBLIM in SYSGEN's parameter list or by defining the logical name REM\$MAX_TERMINALS. The largest possible value for REM\$MAX_TERMINALS is 32767. The command procedure RTTLOAD.COM uses the larger of the values of RJOBLIM and either REM\$MAX_TERMINALS (if it is defined) or the default 255 to compute the required process quotas for the REMACP process.

You can adjust the maximum number of remote users on a running system with the following procedure *after* all remote users have logged out.

- 1 Run the program SYS\$SYSTEM:STOPREM to properly shut down REMACP. This disables future remote logins.
- 2 Have all remote users log out.
- 3 Increase the value of RJOBLIM in SYSGEN's **active** list or define the logical name REM\$MAX_TERMINALS to the newly desired maximum number of remote users.
- 4 Execute the command file SYS\$MANAGER:RTTLOAD.COM. This provides REMACP with the correct quotas to support the specified number of remote users, and enables remote logins.

The SYSGEN Parameter RJOBLIM is Now Dynamic

You can modify the value of the parameter RJOBLIM (allowable number of remote users) dynamically by changing the value in SYSGEN's **active** parameter list within the range set at boot time. Possible values for RJOBLIM now range from 0 (no remote users) to 32767.

Increasing the value of RJOBLIM allows additional remote users to log in to the system. However, if you increase the value of RJOBLIM above the range set at boot time, the image REMACP stops accepting additional incoming connections because of process resource limitations.

Decreasing the value of RJOBLIM below the current number of remote users does not log out any of the remote users. However, once a user logs out, the image REMACP will no longer accept connections until the number of remote users drops below the value of RJOBLIM.

System Manager Release Notes

3.39 Show Cluster Utility — INCN_TIME Field Now Obsolete

3.39 Show Cluster Utility — INCN_TIME Field Now Obsolete

V5.4 With VMS Version 5.4, the INCN_TIME field of the SHOW CLUSTER command ADD (Field) has been removed.

In previous versions of VMS, the INCN_TIME field was an interpretation of the incarnation number in time format. Because some systems communications services (SCS) hosts no longer implement incarnation number in time format, the INCN_TIME field is no longer meaningful.

3.40 SHUTDOWN.COM — Change in Disk Dismount Reporting

V5.4 In previous versions of VMS, the orderly shutdown command procedure, SHUTDOWN.COM, listed each disk device to be dismounted as the operation occurred.

With VMS Version 5.4, the disk dismount operations still occur, but the procedure does not list the names of the disks. If you find the dismount listings useful, you can cause the orderly shutdown procedure to provide the listings by defining the logical name SHUTDOWN\$VERBOSE before you invoke the procedure.

3.41 STARTUP.COM — New Sequence of Operations

V5.4 In previous versions of VMS, the STARTUP.COM command procedure executed the site-specific command procedures in the following sequence:

- 1 SYS\$MANAGER:SYLOGICALS.COM
- 2 SYS\$MANAGER:SYPAGSWPFILES.COM
- 3 SYS\$MANAGER:SYCONFIG.COM
- 4 SYS\$MANAGER:SYSTARTUP_V5.COM

This sequence of operations is documented incorrectly in the *Guide to Setting Up a VMS System*.

With VMS Version 5.4, the sequence of operations for STARTUP.COM has changed. These changes correct inconsistencies between the VMS documentation and the observed behavior of VMS systems and allow for a more logical execution of the command procedures.

The sequence of operations for STARTUP.COM is now as follows:

- 1 SYS\$MANAGER:SYCONFIG.COM executes.
- 2 To add any new drivers to the system configuration, the SYSGEN command AUTOCONFIGURE ALL executes unless the user cancels it.

If the symbol STARTUP\$AUTOCONFIGURE_ALL is defined as zero ("0") or "FALSE" by SYS\$MANAGER:SYCONFIG.COM, this step is not performed.
- 3 SWAPFILE1.SYS, if present, is installed.

System Manager Release Notes

3.41 STARTUP.COM — New Sequence of Operations

- 4 The CONFIGURE process (swapable) starts. If the SYSGEN parameter NOAUTOCONFIG is set to 1, the CONFIGURE process is not started.

If the symbol STARTUP\$AUTOCONFIGURE_ALL is defined as zero ("0") or "FALSE" by SYS\$MANAGER:SYCONFIG.COM, this step is not performed.

- 5 SYS\$MANAGER:SYLOGICALS.COM executes.

The user is now assured that all devices are available (AUTOCONFIGURE ALL) or will become available (CONFIGURE). (In previous versions of VMS, it might have been necessary to configure devices within SYS\$MANAGER:SYLOGICALS.COM.)

- 6 SYS\$MANAGER:SATELLITE_PAGE.COM executes.
- 7 SYS\$MANAGER:SYPAGSWPFILES.COM executes.
- 8 SYS\$MANAGER:SYSECURITY.COM executes.
- 9 SYS\$MANAGER:SYSTARTUP_V5.COM executes.

3.42 SYSGEN Utility Notes

The release notes in this section pertain to new or changed SYSGEN parameters.

3.42.1 RECNXINTERVAL Parameter

V5.0

The SYSGEN parameter RECNXINTERVAL continues to specify the minimum amount of time that the connection manager will attempt to restore a failed connection to another VAXcluster node. However, a change was made so that the value specified is maximized against the time that it takes for a remote node to discover that the connection is broken.

The change means that the effective value used to time out a failed connection is the greater of RECNXINTERVAL on the local node or a value supplied by the remote node that is dependent on the type of hardware port and the value of certain SYSGEN parameters. This change was made to ensure that a node will not be removed from a VAXcluster configuration before it is able to discover that a communication problem exists.

It is possible to have a different value of effective RECNXINTERVAL for different connections. The Show Cluster Utility displays the effective value in the RECNXINTERVAL field.

The current values of RECNXINTERVAL are listed in Table 3-4.

Table 3-4 RECNXINTERVAL Values

RECNXINTERVAL	Minimum Value Based on Remote Port Type	Effective Value
20 seconds Ethernet port ¹	16 seconds	20 seconds
20 seconds CI port	30 seconds	30 seconds ²

¹Default value of the parameter.

²Computed from SYSGEN parameters as $3 * \max(2 * \text{PAPOLLINTERVAL}, \text{PASTIMOUT})$. This results in a value of 30 seconds using default parameter values.)

3.42.2 New PRIORITY_OFFSET Parameter

V5.4

Beginning with VMS Version 5.4, the new SYSGEN parameter PRIORITY_OFFSET specifies the difference in priority required by the scheduler for a process to preempt the current process. The PRIORITY_OFFSET parameter affects normal priority (0 to 15) processes only. The default value is 0.

For example, setting the PRIORITY_OFFSET value to 2 indicates that the current process is executing at priority 1. Therefore, a computable process at priority 2 or 3 would not be allowed to preempt the current process. However, a process with a computable priority of 4 or higher would preempt the current process.

3.42.3 New SCSI_NOAUTO Parameter

V5.4

Beginning with VMS Version 5.4, the SYSGEN Autoconfiguration Facility has a new parameter called SCSI_NOAUTO.

SCSI_NOAUTO disables the loading of the VMS drivers for small computer system interface (SCSI) devices. SCSI_NOAUTO functions the same as parameter VMUSD2 functioned in VMS Version 5.3. In Version 5.4, VMUSD2 is a special parameter reserved for Digital use.

For more information on SCSI_NOAUTO, see the *VMS Device Support Manual* and the *VMS I/O User's Reference Manual: Part I*.

3.42.4 Symmetric Multiprocessing (SMP) Not Supported on Uniprocessor Systems Running DECwindows Server

V5.4

Full symmetric multiprocessing (SMP) checking is not supported on uniprocessor systems running the DECwindows server (for example, workstations). This restriction does not apply to multiprocessor workstations (for example, the VAXstation 3520 workstation).

System Manager Release Notes

3.42 SYSGEN Utility Notes

Modifying the SYSGEN parameter MULTIPROCESSING on a workstation can result in unexpected system failures. If full SMP checking is desired, you must reboot the workstation and must not run DECwindows while full SMP checking is in effect.

3.42.5 TAPE_ALLOCLS Parameter

V5.2

Beginning with VMS Version 5.2, SYSGEN has a new parameter called TAPE_ALLOCLS.

TAPE_ALLOCLS determines the tape allocation class for the system. The tape allocation class is used to create a unique clusterwide device name for multiple access paths to the same tape.

You can also use the TAPE_ALLOCLS parameter to generate a unique clusterwide name for tape devices with identical unit numbers.

3.43 SYSLOA — Page Fault Problem Corrected

V5.3

In previous versions of VMS, a PGFIPLHI (page fault with IPL too high) crash would occasionally occur when a program counter (PC) was within an area of memory allocated to SYSLOA/MOUNTVER. This problem has been corrected.

3.44 SYSMAN Utility Notes

The notes in this section pertain to the SYSMAN Utility.

3.44.1 DO Command — Changes to Symbol Substitution and DCL Verification

V5.4

VMS Version 5.4 includes the following changes to the SYSMAN command DO:

- In previous versions of VMS, symbol substitution was not performed with the SYSMAN command DO:

Beginning with VMS Version 5.4, the SYSMAN command DO permits symbol substitution in the command line. For example:

```
SYSMAN> DO WRITE SYS$OUTPUT "This node is a
'F$GETSYI("HW_NAME")"
%SYSMAN-I-OUTPUT, command execution on node NNAME
This node is a VAX 8840
```

Without the symbol substitution capability, the same command line would not substitute the system name, as follows:

```
SYSMAN> DO WRITE SYS$OUTPUT "This node is a
'F$GETSYI("HW_NAME")"
%SYSMAN-I-OUTPUT, command execution on node NNAME
This node is a 'F$GETSYI("HW_NAME")
```

- In previous versions of VMS, the SYSMAN command DO turned DCL verification on when performing operations on remote nodes.

System Manager Release Notes

3.44 SYSMAN Utility Notes

Beginning with VMS Version 5.4, the SYSMAN command DO turns DCL verification off when performing operations on remote nodes. To change the DO command operation, you can use the new DCL command SET PROFILE/[NO]VERIFY to specify whether verification should be turned on or off.

For a detailed description of the changes to the DO command, see the *VMS SYSMAN Utility Manual*.

3.44.2 **PARAMETERS SET/STARTUP Command — Problem Corrected**

V5.4

Prior to VMS Version 5.4, the SYSMAN command PARAMETERS SET /STARTUP did not work correctly, and users were instructed to use the System Generation Utility (SYSGEN) to set the name of their site-independent startup command procedure.

This problem has been corrected; you can now use the SYSMAN command PARAMETER SET/STARTUP to set the name of a site-independent startup procedure to a given file specification.

For more information about the PARAMETERS SET command, see the *VMS SYSMAN Utility Manual*.

3.44.3 **SET PROFILE Command — Problem Corrected**

V5.4

The SET PROFILE command in the SYSMAN Utility allows you to set a default directory and privileges when executing SYSMAN commands on a remote system.

Prior to VMS Version 5.4, the profile context set by the SET PROFILE command sometimes was lost when you stopped execution of a SYSMAN command by pressing Ctrl/C, if the environment was set to nonlocal nodes. After pressing Ctrl/C, you needed to repeat any SET PROFILE commands to reestablish the profile context.

This problem has been corrected; the profile context is not lost if you stop execution of a SYSMAN command by pressing Ctrl/C.

3.45 **SYSTARTUP_V5.COM — Starting the Queue Manager**

V5.0-1

The following command is used as the example in SYS\$MANAGER:SYSTARTUP_V5.COM for starting the Batch/Print Queue Manager:

```
$ START/QUEUE/MANAGER/BUFFER_COUNT=10/EXTEND_QUANTITY=25 -  
_$ SYS$COMMON:[SYSEXE]JBCSYSQ.QUE.DAT
```

The local buffer count value (specified with /BUFFER_COUNT=10), the initial allocation, and the subsequent file extension value (specified with /EXTEND_QUANTITY=25) are appropriate for a small system with one batch and one print queue. These values should be increased for standalone time-sharing and clustered machines that support many queues.

System Manager Release Notes

3.45 SYSTARTUP_V5.COM — Starting the Queue Manager

The default values for `/BUFFER_COUNT` and `EXTEND_QUANTITY` are 50 and 100, respectively. These values are generally adequate to support 5 to 20 queues where the total number of concurrent jobs is typically less than 50. To efficiently support more queues and jobs, Digital recommends specifying larger values for these qualifiers when starting the queue manager. Note that the value for the `/EXTEND_QUANTITY` qualifier should be the same for all nodes in a cluster.

Increasing the local buffer count decreases the number of direct I/O reads on the queue file required to perform Batch/Print operations at the expense of job controller working set size and locking activity. When memory is available, a large number of local buffers increases system performance. However, if a small amount of memory is available, using 100 or more local buffers can decrease performance by causing excessive page faulting of the job controller process.

The value for extend quantity should be at least 20 percent of the size of the queue file when the queue file is in a steady state. If the value for the extend quantity is too small, fragmentation of the queue file can occur as a result of the many file extend operations being performed on the disk.

V5.4 For a description of VMS Version 5.4 enhancements to the `START/QUEUE/MANAGER/BUFFER_COUNT` command, see Section 3.7.3.

3.46 System Disk Size Recommendation: 100 Mbytes

V5.4 To support VMS Version 5.4, a system disk of greater than 100 Mbytes is recommended. When you use a smaller disk, additional tailoring is required prior to installing options such as DECwindows. This recommendation does not include the dump file space.

For information on tailoring, see the *VMS Version 5.4 Upgrade and Installation Manual*.

3.47 System Dump Analyzer (SDA) Utility — Requirements for VIRTUALPAGECNT System Parameter

V5.0-2 The Version 5.0 *VMS System Dump Analyzer Utility Manual* recommends that you set the system parameter `VIRTUALPAGECNT` to the size of the system dump file plus 3000 in order to do the following:

- Maintain sufficient virtual address space for the System Dump Analyzer to map a dump
- Load any required symbol tables
- Store stack information

Digital now recommends that you set `VIRTUALPAGECNT` to at least the size of the system dump file plus 4700. If your SDA sessions require many symbols (and invoke the `READ/EXECUTIVE` command), you should set `VIRTUALPAGECNT` to the size of the dump file plus 5750.

3.48 TA90E Tape Drive — Usage Guidelines

V5.3-2

VMS Version 5.3-2 provides support for the TA90E tape drive. The TA90E tape drive is an enhanced version of the TA90 tape drive that includes the ability to compact and block data records together automatically. Data compaction and record blocking increases the amount of data that can be stored on a single tape cartridge.

Using the TA90E tape drive, all records on a given piece of media can either be compacted and blocked, or they can be recorded in the same way that they would be recorded by a TA90 tape drive. Note that for the TA90E tape drive, once data compaction or noncompaction has been selected for a given cartridge, that status applies to the entire cartridge.

The default for the compaction and record-blocking feature is NOCOMPACTION. However, you can enable compaction by using any of the following DCL commands:

- INITIALIZE
- MOUNT
- SET MAGTAPE
- BACKUP

The following qualifier is used to control compaction for all four commands:

```
/MEDIA_FORMAT=[NO]COMPACTION
```

For example, to initialize a cartridge with compaction and record blocking enabled, enter the following command:

```
$ INITIALIZE MUA0: label /MEDIA_FORMAT=COMPACTION
```

Note that when you enable compaction, caching is automatically enabled regardless of the use of the /[NO]CACHE qualifier. For more information about this qualifier, see the *VMS Mount Utility Manual*.

To create a BACKUP tape with compaction and record blocking disabled, enter the following command:

```
$ BACKUP filespec MUA0: ... /MEDIA_FORMAT=NOCOMPACTION -  
_ $ /REWIND /INITIALIZE
```

For more information about using BACKUP with the TA90E, see Section 3.48.2.

3.48.1 Identifying the TA90E with SHOW DEVICE

V5.3-2

Use the SHOW DEVICE/FULL command to determine if a TA90E tape drive is available to your system.

The following example shows the system response for a TA90E tape drive with device name \$90\$MUA0. In the example, the TA90E tape drive is connected to a hierarchical storage controller (HSC) named KLX1, and data compaction is enabled.

```
$ SHOW DEVICE/FULL $90$MUA0:
```

System Manager Release Notes

3.48 TA90E Tape Drive — Usage Guidelines

The system responds as follows:

```
.  
. .  
Magtape $90$MUA0: (KLX1), device type TA90, is online, allocated,  
controller supports tape data caching (write-back cache enabled),  
controller supports compaction (compaction enabled), file-oriented  
device, available to cluster, error logging is enabled.
```

3.48.2 Limitation of Using BACKUP with the TA90E Tape Drive

V5.3-2

The TA90E tape drive compacts data and consequently places more than a single BACKUP block in a tape record. This data compaction compromises the data integrity of BACKUP by making BACKUP's exclusive OR (XOR) block recovery less effective than on noncompacting drives. However, in the event of errors in the data unpacking process, the errors can still be detected by BACKUP's cyclic redundancy check (CRC) and recovered by BACKUP's XOR blocks.

Note: The redundancy group feature of BACKUP is less effective when used with the TA90E data compaction and record blocking feature.

Digital is working on a way to ensure preservation of the data integrity model supported by BACKUP.

3.49 TDRIVER.MAR — Corrections

V5.2

A new version of the file TDRIVER.MAR is supplied in SYS\$EXAMPLES:. This version follows the recommended customer conventions of using underscores (_) in symbol names instead of dollar signs (\$).

The unit initialization routine now returns a status in R0 that is required for BI device drivers.

The numeric constant offset for VEC\$L_ISR was replaced with the correct symbolic offset. In addition, several comments have been corrected.

3.50 UNIBUS — Floating Interrupt Vector Change

V5.4

With VMS Version 5.0, the algorithm used to allocate interrupt vectors for UNIBUS peripherals changed. Because of this change, when systems are autoconfigured during booting, some may require UNIBUS peripherals to have their interrupt vectors modified by a Digital Customer Services representative. Note that this change does not affect most VMS systems.

Those systems affected include any system with two or more UDA, KDA, RQDX, or BDA controllers on the same bus with any other device that has a hard-wired floating interrupt vector alignment of four, such as the following:

- RX211
- DR11W OR DRV11W
- DMZ32

System Manager Release Notes

3.50 UNIBUS — Floating Interrupt Vector Change

- LNV21
- VS100
- VSV21
- IBQ01

Support for New Behavior Only

In previous VMS 5.x versions, SYSGEN supported the old and new vector allocation algorithms. The new algorithm was used as the default behavior, and the old algorithm was available through the VMS8 parameter. The VMS8 parameter was set to give either the old or new algorithm behavior.

With VMS Version 5.4, SYSGEN supports the new algorithm behavior only; the VMS8 parameter has been removed.

3.51 Upgrade Notes

The release notes in this section contain information that pertains to upgrading the VMS operating system.

3.51.1 Layered Product Cautions

V5.4

After you upgrade to VMS Version 5.4, you should not have to reinstall most layered products. Some layered products, however, require you to perform additional upgrade steps to ensure that the layered products are installed correctly on VMS Version 5.4. Some available layered products that exhibit compatibility problems with this version are listed in Table 3-5 and described in Sections 3.51.1.3 and 3.51.1.4.

For a complete listing of layered products available for VMS Version 5.4, see the appropriate appendix in the *VMS Version 5.4 Upgrade and Installation Manual*. If you require additional information on any upgrade procedure, or if you encounter problems, contact your Digital Customer Service representative.

Table 3-5 Layered Products with Special Requirements for VMS Version 5.4

Layered Product	Version Number
ALL-IN-1	2.4
DIGITAL Distributed File Service (DECdfs)	1.0, 1.1
VAX Public Access Communications	1.2
VAX TU70/72 Device Driver	1.2
VAX Workstation Software (VWS)	All

System Manager Release Notes

3.51 Upgrade Notes

V5.4 3.51.1.1 ALL-IN-1 Shareable Images Requirement for CDA Support
VMS Version 5.4 provides two new shareable images that are activated by the Compound Document Architecture (CDA) support for ALL-IN-1 Version 2.4. ALL-IN-1 is a privileged image; therefore, any images activated by ALL-IN-1 must also be installed as known images.

The new shareable images for VMS Version 5.4 are not installed as known images. If you require CDA support for ALL-IN-1 Version 2.4, you must install the two new shareable images as known images. If you do not require CDA support, no action is required.

To install the two shareable images for CDA support, add the following command lines to your ALL-IN-1 site startup file, OAS\$SITE_BUILD_SHARE:A1V24_SITE_START.COM:

```
$ CREATE SYS$SHARE:XDPS$DPSLIBSHR.EXE
$ INSTALL CREATE SYS$SHARE:XDPS$DPSCLIENTSHR.EXE
```

V5.4 3.51.1.2 DIGITAL Distributed File Service (DECdfs) Update Requirements
The VMS Version 5.4 distribution kit contains an update that is required for DIGITAL Distributed File Service (DECdfs) running on VMS Version 4.n systems.

If you are running DECdfs on VMS Version 5.0 or later, you need not perform the required update. However, if you are running DECdfs on a system running VMS Versions 4.4 through 4.7, you must apply the required update to that system.

DECdfs Version 1.0 runs on VMS Versions 4.4 through 4.7. DECdfs Versions 1.1 and 1.2 run on VMS Versions 4.6 and later. If you are running DECdfs Version 1.0 or 1.1, Digital recommends that you upgrade to DECdfs Version 1.2 before applying this update. If you choose to continue running an earlier version of DECdfs, you can also apply this update to the earlier version.

The required update is contained in a save set named DFSMUP012.A on the VMS Version 5.4 distribution kit. You can obtain the save set by extracting it from the VMS054.D save set or by copying it from SYS\$UPDATE from a system disk that has been upgraded to VMS Version 5.4. To extract the save set, load the first volume of your VMS Version 5.4 distribution kit (TK50 tape cartridge 1 of 2, or magnetic tape reel 1 of 3) and enter a command line in the following format:

```
$ BACKUP/VERIFY/LOG/SELECT=DFSMUP012.A source-drive:[ ]VMS054.D/SAV *
```

V5.4 3.51.1.3 VAX Public Access Communications Requirements
Sites using or installing VAX Public Access Communications Requirements must install this product before upgrading to VMS Version 5.4. If this product is installed on a previous version of VMS, it will work correctly on VMS Version 5.4. The Version 5.4 installation procedure contains a problem that will be corrected a future release of VMS.

3.51.1.4 VAX TU70/72 Device Driver Not Compatible

V5.4

The VAX TU70/72 Device Driver Version 1.2 requires special assistance in order to function correctly on VMS Version 5.4. The VAX TU70/72 Device Driver will not continue to work correctly on an upgraded system and will not install on VMS Version 5.4. Assistance for the VAX TU70/72 Device Driver is available from Digital Customer Support Centers.

3.51.1.5 VAX Workstation Software (VWS) Requirement

V5.4

After upgrading to VMS Version 5.4, you must reinstall the VAX Workstation Software (VWS). This reinstallation requirement applies to all versions of VWS. Otherwise, workstations that are running VWS will not operate correctly.

3.51.2 Rolling Upgrades from VMS Version 5.3 to Version 5.4

V5.4

To perform a VMS Version 5.4 rolling upgrade, all system disks in the cluster must be running at least VMS Version 5.3. If you are performing a rolling upgrade in a VAXcluster environment that contains systems running Version 5.3, you must perform additional steps and reboot each Version 5.3 system before upgrading. You do not have to perform these additional steps on systems running Versions 5.3-1 or 5.3-2 or on the system disk you are upgrading.

For detailed information about the upgrade procedure, see the *VMS Version 5.4 Upgrade and Installation Manual*.

3.51.3 VAXstation 8000 Upgrade Unsupported

V5.4

Beginning with VMS Version 5.4, the VAXstation 8000 computer is no longer supported. Therefore, do not perform an upgrade to VMS Version 5.4 on a VAXstation 8000.

3.51.4 VMS Kits Provided on RX33 Diskettes

V5.2

If you receive your VMS kit on RX33 diskettes, the DECwindows kit is provided on a TK50. Please be aware that Phase 1 of the upgrade procedure will delete any DECwindows files currently on your system. The file deletion occurs before the procedures ask what VMS and DECwindows options you want to select. If you are not able to use the TK50 DECwindows kit during the upgrade (no drive available), please be sure *not* to select any of the DECwindows options. If you need to have DECwindows installed on the system, please contact your Digital sales representative.

3.52 VAX 8000-Series Computer Notes

The release notes in this section pertain to VAX 8000-series computer systems.

System Manager Release Notes

3.52 VAX 8000-Series Computer Notes

3.52.1 **DMB32 Product Software Required for DMB32 Communications Controller**

V5.0 VAX 8200, 8250, 8300, 8350 and VAX 8530, 8550, 8700, and 8800 systems that include the DMB32 communications controller must install the DMB32 optional software product in order to use the controller's synchronous port. The VMS operating system kit does not contain the DMB32 software.

3.52.2 **Halting a VAX 8530, 8550, 8700, or 8800 System**

V5.0 After you halt a VAX 8530, 8550, 8700, or 8800 system, you must enter the `CLEAR RESTART_FLAGS` command to clear the `WARM_RESTART` and `COLD_RESTART` flags. For example:

```
>>> HALT
>>> CLEAR RESTART_FLAGS
```

Clearing these flags prevents the automatic boot and restart procedures from looping indefinitely when you enter the next `BOOT` command. Keep this in mind when you halt the system during the VMS installation procedure.

3.52.3 **SET TIME/CLUSTER Command**

V5.4 If you have a VAXcluster environment that includes a VAX 8530, 8550, 8700, 8800, 8820, 8830, or 8840 system, you must make sure the system consoles are connected and running the console program before you enter the `SET TIME/CLUSTER` command. Otherwise, the system fails.

3.52.4 **SET TIME Command — Problem**

V5.4 Normally, when you enter the `SET TIME` command, the date and time are stored internally. However, on a VAX 8530, 8550, 8700, 8800, 8820, 8830, and 8840 system, the date and time are sometimes stored incorrectly due to a protocol error in the console interface. If the date and time are stored incorrectly, the system asks you for the data and time each time you boot the system.

3.52.5 **VAX 8800 Systems Running SMP — Problem**

V5.0 There is a problem with UNIBUS operations on VAX 8800 processors when running symmetrical multiprocessing (SMP). The VAX 8800 NBIA (memory interconnect to VAXBI adapter) and UBA (UNIBUS adapter) can deadlock while waiting for each other to complete certain operations. Because both adapters can process exactly one transaction at a time and because they can also request the assistance of the other in order to complete a transaction, the deadlock situation is quite probable.

System Manager Release Notes

3.52 VAX 8000-Series Computer Notes

To avoid this deadlock situation, the VMS operating system forces PRIMARY affinity for all UNIBUS controllers configured in a VAX 8800 system. The enforcement of PRIMARY affinity prevents the deadlock situation from occurring. The requirement to limit access for UNIBUS devices to the PRIMARY processor is a VAX 8800 restriction and does not apply to other SMP systems.

If you have written a UNIBUS device driver that has been converted to run on SMP configurations, you may want to allow that driver to execute on both CPUs in a VAX 8800 configuration. To allow this, the driver must first guarantee that it does not perform any READ-MODIFY-WRITE operations to I/O address space. For example, it cannot perform a BISW #x,(R2), where R2 is pointing to a UNIBUS control and status register (CSR). If a device driver has been verified to behave correctly, then it can circumvent the restriction that forces all I/O operations to execute on the PRIMARY processor.

For a device driver to circumvent the PRIMARY affinity policy, it must set the UCB\$L_AFFINITY field of the unit control block (UCB) to -1 in the device driver's UNIT or CONTROLLER INITIALIZATION routine.

3.52.6 VAXBI 5 Restriction

V5.4 The VAX 8820, 8830, and 8840 systems support up to six VAXBI adapters: VAXBI 0 through VAXBI 5. The system cannot recognize devices connected to node 15; therefore, do not configure the last VAXBI adapter (VAXBI 5) as node 15.

3.53 VAX 9000 Computer — AUTOGEN with FEEDBACK Requirement

V5.4 For the VAX 9000 computer, AUTOGEN's initial parameter calculations are conservative. To obtain parameter values that match your system workload, you may need to perform AUTOGEN operations with the FEEDBACK parameter a number of times. Once the parameter values are within the requirements placed on the system, AUTOGEN with FEEDBACK correctly matches the system resources with your workload.

To provide a more reasonable starting set of parameters, Digital recommends that you add the following entries to SYS\$SYSTEM:MODPARAMS.DAT for the initial run of AUTOGEN:

```
MIN_SRPCOUNT = 3000
MIN_IRPCOUNT = 1500
MIN_LRPCOUNT = 100
MIN_MAXPROCESSCNT = 400
MIN_NPAGEDYN = 500000
MAX_VIRTUALPAGECNT = 300000
MIN_BALSETCNT = 350
MIN_SYSMWCNT = 2000
MIN_SCSCONNCNT = 40
MIN_LNMSHASHTBL = 2048
QUANTUM = 5
```

Note: By lowering the maximum page count on the SYSGEN parameter VIRTUALPAGECNT, the preceding values allow a large number of users. If you need to run large applications that require

System Manager Release Notes

3.53 VAX 9000 Computer — AUTOGEN with FEEDBACK Requirement

VIRTUALPAGECNT to be higher than 300,000, you must reduce the value of the BALSETCNT parameter.

3.54 VAX Computers — VMS Support

V5.3

Table 3-6 lists the versions of the VMS operating system and the VAX computers that these versions first supported.

Table 3-6 VMS Versions and Associated VAX Computers

VMS Version	First Computer Support
Version 5.0	VAX 6000 Model 2xx, VAX 8820-8840
Version 5.0-2A	MicroVAX 3300/3400
Version 5.1	VAX 6000 Model 3xx, MicroVAX 3800/3900
Version 5.1-B	VAXstation 3100, Model 30/40
Version 5.1-1	VAXstation 3520/3540
Version 5.2	VAXstation 6000 Model 4xx, MicroVAX 3100
Version 5.2-1	VAXstation 3100, Model 38/48
Version 5.3	No CPUs introduced
Version 5.3-1	No CPUs introduced
Version 5.3-2	VAX 4000
Version 5.4	VAXstation 3100 Model 76, VAX 9000, VAXft 3000

3.55 VAXcluster Reconfiguration Time Reduced

V5.4

The following VMS Version 5.4 modifications to the VAXcluster software have reduced state transition times; these modifications apply to all configurations.

- In the connection manager, a potential 1-second delay has been eliminated. This delay occurred during a state transition when a node was removed from the VAXcluster configuration.
- The lock database rebuild algorithm has been enhanced to increase parallelism, thereby reducing the elapsed time.
- During the execution of SYS\$SYSTEM:SHUTDOWN.COM, the lock manager now moves locally mastered resources to remaining nodes prior to the beginning of the state transition. This move results in fewer resources to be rebuilt during the state transition, thereby reducing transition times.

3.56 VAXft 3000 Computer Notes

The release notes in this section pertain to the VAXft 3000 system.

3.56.1 \$GETDVI and F\$GETDVI — Use to Obtain VAXft 3000 Device Information

V5.4

For VMS Version 5.4, the \$GETDVI system service and the F\$GETDVI lexical function provide the device-class and device-type information for VAXft 3000 devices listed in Table 3–7:

Table 3–7 Information Provided by \$GETDVI and F\$GETDVI

Device	Device Class	Device Type
RF31	DC\$_DISK	DT\$_RF31
TF70	DC\$_TAPE	DT\$_TF70
DSF32	DC\$_SCOM	DT\$_DSF32
CM	DC\$_MISC	DT\$_SFUN9
EP	DC\$_SCOM	DT\$_EP_LANCE
EF	DC\$_SCOM	DT\$_FT_NI
PW	DC\$_BUS	DT\$_SWIFT
SM	DC\$_SCOM	DT\$_SM_DSF32
SF	DC\$_SCOM	DT\$_SM_DSF32
GD	DC\$_MISC	DT\$_DMA_520

3.56.2 DELTA/XDELTA Utility — How to Bootstrap a VAXft 3000 System

V5.4

For VMS Version 5.4, the DELTA/XDELTA Utility supports the VAXft 3000 system.

To bootstrap VMS Version 5.4 with XDELTA on a VAXft 3000 system, enter the following command:

```
>>> BOOT/R5:7 DIxn
```

DI is the name of the disk from which to bootstrap the system, *x* is the I/O adapter, and *n* is the unit number.

3.56.3 System Dump Analyzer (SDA) Utility — How to Cause a VAXft 3000 System Failure

V5.4

For VMS Version 5.4, the System Dump Analyzer (SDA) Utility supports the VAXft 3000 system.

To cause a VAXft 3000 system failure, you must enter commands at the console prompt (>>>). To display the console prompt, press F5, then press Return.

At the console prompt, enter the following commands to cause the VAXft 3000 system to fail:

System Manager Release Notes

3.56 VAXft 3000 Computer Notes

```
>>> E PSL
>>> E/I/N:4 0
>>> D PC FFFFFFFF
>>> D PSL 41F0000
>>> C
```

3.57 VAXstation 3520 and 3540 Computer Notes

The release notes in this section pertain to the VAXstation 3520 and 3540 computers.

3.57.1 Console Support on the Graphics Terminal

V5.4

After you start DECwindows, if Ctrl/F2 are the first keys you press, the key sequence is not acknowledged. You must press Ctrl/F2 a second time to either bring up or take down the console window. This problem occurs only at the start of DECwindows.

3.57.2 Software Products Information

V5.4

Certain software products may not be available for the VAXstation 3520 and 3540 computers. See your Digital sales representative for a complete list of the software products available for your system.

3.58 VMS Executive Notes

The release notes in this section pertain to the VMS Executive.

3.58.1 Changes to Process Paging File Assignment

V5.0

In Version 4.*n*, a process was assigned to a paging file at process creation time. The page file chosen was the one with the most free, or unallocated, blocks.

A process may now use up to four page files simultaneously. The set of page files used changes dynamically as the process executes.

One consequence of this change is that paging file global sections are no longer forced to have their backing store in the primary pagefile, SYS\$SPECIFIC:[SYSEXEC]PAGEFILE.SYS. RMS global buffers are frequent users of pagefile global sections.

3.58.2 Extended Working Set and Virtual Address Space Sizes

V5.4

Prior to VMS Version 5.4, the maximum working set allowed was 100,000 pages and the maximum virtual address space allowed was 600,000 pages.

With VMS Version 5.4, the maximum working set allowed is 200,000 pages and the maximum virtual address space allowed is 1,000,000 pages.

3.58.3 PAGEFILE.SYS — Changes

V5.0

In Version 4.n, when the system was booted, SYSINIT looked for the primary page file, SYS\$SYSROOT:[SYSEXE]PAGEFILE.SYS. If PAGEFILE.SYS was not found an error message was displayed and the boot operation continued even though the VMS operating system did not support a configuration without a primary paging file.

In Version 5.0, if PAGEFILE.SYS is not found, SYSINIT issues the following informational message:

```
"%SYSINIT-I- PAGEFILE.SYS not found - system initialization continuing..."
```

Then, in STARTUP.COM, before any of the system overhead processes are created (for example, OPCOM, JOBCTL), the startup command procedure searches for SYS\$MANAGER:SYPAGSWPFILES.COM. If the site-specific file is found, it is invoked.

In addition, an abbreviated version of SYPAGSWPFILES.COM is included.

You may include any necessary commands in SYPAGSWPFILES.COM to accomplish the installation of paging and swap files (for example, volume initialization, mounting disks, SYSGEN INSTALL commands). The CONFIGURE process is running at the time the site-specific COM file is invoked so that in the absence of any controller or device errors, HSC-based disks will eventually be recognized by the system.

When control returns to STARTUP, at least one paging file must be successfully installed. If one is not installed, STARTUP reports the following error:

```
"%STARTUP-E-NOPAGFIL, no page files have been successfully installed."
```

In effect, these changes make the notion of a primary paging and swap file almost obsolete.

3.58.4 Paging File Recommendation

V5.0

Digital recommends that you reexamine the process of creating a minimal, primary paging file on the system disk and significantly larger paging files on alternate disks. In particular, the best paging file load balancing tends to occur when all paging files are created to approximately the same size.

3.59 VMSINSTAL — DCLTABLES Logical Name Not Used

V5.4

VMSINSTAL does not currently handle the parsing of a logical name for the file DCLTABLES.EXE. A layered product installation that inserts DCL verbs into the DCL table is inserted into the file SYS\$COMMON:[SYSLIB]DCLTABLES.EXE.

Before you install products that insert DCL verbs, proceed as follows:

- Place a copy of your working DCLTABLES.EXE file into the SYS\$COMMON:[SYSLIB] directory.

System Manager Release Notes

3.59 VMSINSTAL — DCLTABLES Logical Name Not Used

- After you have completed the product installation, copy the SYS\$COMMON:[SYSLIB]DCLTABLES.EXE file to the desired location. For all nodes in the cluster, install this version of the DCLTABLES.EXE file using the VMSINSTAL utility.

3.60 Volume Shadowing Phase II Notes

The release notes in this section pertain to VMS Volume Shadowing phase II.

3.60.1 AUTOGEN Operations — Volume Shadowing Adjustment Required

V5.4

If you performed an AUTOGEN operation after you upgraded or installed the VMS operating system, and now you want to use Volume Shadowing phase II for the first time, you must adjust the SYSGEN parameter NPAGEDYN before you run Volume Shadowing phase II.

For best results, perform an AUTOGEN operation on your system using the FEEDBACK parameter to modify the SYSGEN parameter NPAGEDYN. Match the setting of NPAGEDYN in MODPARAMS.DAT with that of your workload. Make the necessary adjustments as follows:

- 1 To make sure you have the necessary privileges, log in to the system manager's account.
- 2 Determine the current value for the SYSGEN parameter NPAGEDYN by entering the following command lines:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> USE CURRENT
SYSGEN> SHOW NPAGEDYN
```

- 3 Edit the file SYS\$SYSTEM:MODPARAMS.DAT and add the following entry, where *nnnnn* is the current SYSGEN value plus 36,000:

```
MIN_NPAGEDYN = nnnnn
```

- 4 Perform an AUTOGEN operation on your system by entering the following command line:

```
$ @SYS$UPDATE:AUTOGEN GETDATA REBOOT FEEDBACK
```

For more information about AUTOGEN, see the *Guide to Setting Up a VMS System*.

3.60.2 Batch and Print Jobs — Reentering after Conversion

V5.4

After you convert a disk to a Volume Shadowing phase II shadow set, batch and print jobs that were previously entered in the queuing system database (JBCSYSQUE.DAT) and whose job-related files reside on the device, will not execute as scheduled. Jobs in the pending, holding, timed release, and retained states before the conversion are candidates for this type of failure.

System Manager Release Notes

3.60 Volume Shadowing Phase II Notes

This problem with execution occurs because the job entry in the queue database contains the physical device name of the disk where files associated with the job reside (command procedures and files to be printed). The device name that must now be referenced is the shadow-set virtual unit to which the disk volume was added.

Before converting a disk to a Volume Shadowing phase II shadow set, Digital recommends that you use the following DCL command to make a list of all batch and print jobs that refer to files on the disk:

```
$ SHOW QUEUE/ALL/FULL/OUTPUT=listing-file
```

Then, locate all jobs that refer to the device and do one of the following:

- Run these jobs before performing the conversion.
- Delete the jobs from the queue database and reenter them after the conversion is finished.

3.60.3 Hierarchical Storage Controller (HSC) Revision Levels Required

V5.4

In VMS Version 5.4, the following Hierarchical Storage Controller (HSC) microcode revision levels are required for VMS Volume Shadowing software:

- V500 for HSC40 and HSC70
- V400 for HSC50

3.60.4 SHOW DEVICES Command

V5.4

For systems using VMS Volume Shadowing phase II, the SHOW DEVICES display does not list virtual units on nodes that have not mounted the shadow set.

Shadow set virtual units are not served to the cluster through the mass storage control protocol (MSCP). Virtual units are created and maintained in a distributed fashion on each node in the cluster. Individual physical members, however, are MSCP served; with the SHOW DEVICES command, they are displayed on all nodes from which the physical members are accessible.

3.60.5 Specify a Virtual Unit Name to Obtain \$GETDVI FREEBLOCK Information

V5.4

If you use the \$GETDVI system service to obtain FREEBLOCK count information for a shadow set, you should specify the virtual unit name on the \$GETDVI service. If you specify the device name of one of the shadow set members, the \$GETDVI service returns a value of zero.

System Manager Release Notes

3.60 Volume Shadowing Phase II Notes

3.60.6 **SYSGEN Parameter VMSD3 — Special Parameter to Control Failover**

V5.4

With VMS Version 5.4, a special SYSGEN parameter, VMSD3, controls failover of shadowed volumes. The VMSD3 parameter specifies in hexadecimal the number of seconds (from 1 to 255) during which recovery of a repairable shadow set is attempted. If the high-order bit (%X80000000) is off, recovery of a failing member is attempted for 5 seconds before that member is removed from the shadow set. If the bit is on, the low-order byte is used dynamically to specify the number of seconds in which to attempt the recovery of a failing member. If zero is specified, it is ignored and the 5-second value is used.

The following example shows how to set the failover attempt time to 10 seconds:

```
$ MCR SYSGEN
SYSGEN> USE CURRENT
SYSGEN> SET VMSD3 %X8000000A
SYSGEN> SHOW VMSD3
```

Parameter Name	Current	Default	Min.	Max.	Unit	Dynamic
VMSD3	-2147483638	0	0	-1		D

```
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
$
```

3.61 **VWS Workstations — Setting of Multiprocessing SYSGEN Parameter**

V5.0

If you have a workstation that is running VWS, do not set the SYSGEN parameter MULTIPROCESSING to the value 2; the correct value is 1 (default). Multiprocessing is not supported on VAX workstations running VWS.

3.62 **YFDRIVER Terminal Port Driver Supports New Baud Rate**

V5.3-2

With VMS Version 5.3-2, YFDRIVER terminal port driver supports 38,400 baud (and 50 baud) on the following controllers:

- CXY08
- CXA16
- CXB16
- CXF32
- DSH32
- DHT32
- DHQ11

Note: Due to the speed table implementation in the DHU11 and DHV11 controllers, a speed setting of 38,400 baud (and 50 baud) is not supported on DHU11 and DHV11 controllers.

4 Programmer Release Notes

This chapter includes information about the VMS Version 5.4 operating system that is of interest to both the application and system programmer.

For information about the new features included in VMS Version 5.4, see the *VMS Version 5.4 New Features Manual*.

4.1 \$CANCEL is an Asynchronous Operation

V5.2 The \$CANCEL system service performs an asynchronous cancel operation. This means that the application *must* wait for each I/O operation issued to the driver to complete prior to checking the status for that operation.

For a more detailed discussion, see Section 3.38.

4.2 \$CHANGE_ACL System Service — Lock Correction

V5.4 In previous versions of VMS, a problem in the \$CHANGE_ACL system service caused a lock to be acquired and not released until the image was run down. When you used the \$CHANGE_ACL system service to manage an object's access control list (ACL), the lock problem caused an error to be returned.

This problem has been corrected; a lock no longer occurs.

4.3 \$CMKRNL — Failure of Customer-Written Programs

V5.2 The VAX Procedure Calling Standard prohibits the exchange of information between calling and called procedures by means of any general purpose register with the exception of registers R0 and R1. R0 and R1 are reserved for the return of status from the called procedure to the calling procedure. As described in the *Introduction to VMS System Routines*, registers R2 through R11 must be preserved across a procedure call. If a called procedure uses these registers, it must save and restore them using the procedure entry mask mechanism.

Certain customer-written programs that violate these requirements of the VAX Procedure Calling Standard will fail under VMS Version 5.0, even though they may have run successfully under previous versions of VMS. Specifically, these programs contain code that invokes a procedure by means of the \$CMKRNL system service, and attempt to pass context to that procedure in R2.

A change in the system service dispatcher in VMS Version 5.0 exposes this coding infraction. Prior to VMS Version 5.0, the dispatcher modified only R4; as of VMS Version 5.0, it also uses R2. In future releases of VMS, it may use other registers.

Programmer Release Notes

4.3 \$CMKRNL — Failure of Customer-Written Programs

These customer-written programs (and other programs using prohibited registers) should be fixed to comply with the VAX Procedure Calling Standard and pass context to the procedure called through \$CMKRNL in an appropriate way, such as within an argument list. See the *Introduction to VMS System Routines* for additional information.

4.4 ACL Editor Notes

The release notes in this section pertain to the ACL (Access Control List) editor.

4.4.1 Use of Protected Entries — Correction

V5.4 In previous versions of VMS, the ACL editor encountered problems when dealing with ACLs that contain protected entries. The problems varied depending on the content of the ACL before and after editing.

This problem has been corrected; protected entries no longer present problems for the ACL editor.

4.5 Activating an Image with System Version Mismatch — Change

V5.2 Prior to VMS Version 5.2, running an image linked with a system symbol table (SYS\$SYSTEM:SYS.STB) other than that of the running system resulted in a successful image activation with CMKRNL and CMEXEC privileges removed.

Beginning with VMS Version 5.2, running such an image fails and displays the following error message:

```
SS$_SYSVERDIF, system version mismatch, please relink
```

You should inspect user programs that activate other images linked against the system symbol table (for example, programs that call LIB\$FIND_IMAGE_SYMBOL) to determine whether they depend on the obsolete behavior of VMS versions prior to Version 5.2. If so, remove the dependency on the obsolete behavior from any such user program; then relink, under VMS Version 5.2 or a subsequent version of VMS, both the calling program and the image that failed to activate.

4.6 DCL Subroutine — Modifications to Subroutine Entry Point Scoping

V5.4 To make the scoping of subroutine entry points more intuitive, the following restrictions have been added for VMS Version 5.4:

- Subroutine entry points that are defined within another subroutine are local to that subroutine. You cannot call a subroutine if the subroutine entry point is within a separate subroutine block. For example, the following call is not valid for VMS Version 5.4:

4.6 DCL Subroutine — Modifications to Subroutine Entry Point Scoping

```

$ CALL BAR
$
$ MAIN: SUBROUTINE
$
$     BAR: SUBROUTINE
$     ENDSUBROUTINE
$
$ ENDSUBROUTINE

```

The following call is valid for VMS Version 5.4 because BAR is defined within MAIN:

```

$ MAIN: SUBROUTINE
$
$     BAR: SUBROUTINE
$     ENDSUBROUTINE
$
$ CALL BAR
$ ENDSUBROUTINE

```

- If a subroutine entry point is located within an IF-THEN-ELSE block, you cannot call this subroutine from outside the IF-THEN-ELSE block. For example, the following call is not allowed in VMS Version 5.4:

```

$ IF 1
$ THEN
$     FOO: SUBROUTINE
$     ENDSUBROUTINE
$ ENDIF
$ CALL FOO

```

- Every SUBROUTINE command must have a matching ENDSUBROUTINE command to delimit the subroutine. This is not a new restriction, but it is now enforced more stringently.

In the following example, the entry point for subroutine B is defined within subroutine A because there is no ENDSUBROUTINE to delimit A (that is, EXIT is not a delimiter of A). Therefore, subroutine B is inaccessible from outside subroutine A.

```

$ A: SUBROUTINE
$ EXIT
$
$ B: SUBROUTINE
$ ENDSUBROUTINE

```

4.7 DCL Substring Assignment Problem Corrected

V5.4

In previous versions of VMS, the result of a substring assignment was corrupted if the replacement string was greater than 256 characters. For example:

```

$ AA = F$FAO("!256*A")
$ BB = F$FAO("!500*B")
$ BB[5,256] := 'AA'

```

With VMS Version 5.4, corruption does not occur; the original substring value is maintained and an error is reported.

Programmer Release Notes

4.8 Debugger Notes

4.8 Debugger Notes

The release notes in this section pertain to the VMS Debugger. Unless specified otherwise, the release notes apply to both the command and DECwindows interfaces of the debugger.

4.8.1 Corrected Problems or Restrictions

The following problems or restrictions in previous versions of the debugger have been corrected:

V5.0

- In VMS Version 5.0:
 - If you defined a symbol with the `DEFINE/COMMAND` command, you could not then specify that symbol as the first or only element of a command string in another `DEFINE/COMMAND` command. This problem has been corrected; there are no longer any restrictions on using a previously defined symbol in another `DEFINE/COMMAND` command.
 - If you used the `STEP` command to execute a line of code that caused certain exceptions, and an application-declared exception handler was available to field such exceptions, the debugger suspended execution within the handler. This problem has been corrected; the debugger now allows uninterrupted execution of the exception handler in such cases.
 - In a COBOL program, depositing a value into a variable of type `SIGNED LEADING SEPARATE` modified adjacent memory locations. This problem has been corrected; depositing a value no longer affects adjacent locations.

V5.2

- In VMS Version 5.2:
 - You could not invoke the debugger with a program that was linked with more than 25 images, including any Run-Time Library images. The maximum number of images was raised to 61 images for Version 5.3-2. The problem has now been corrected; there are no longer any restrictions on the number of images that can be linked with your program.
 - To achieve good debugger startup performance with large images (more than 5000 blocks), you had to adjust your working set quota (`WSQUOTA`) to 3000 or more. This problem has been corrected; you no longer have to adjust your working set quota to achieve good startup performance with large images.

V5.2

- In the DECwindows interface for VMS Version 5.2:
 - There was a problem with the dimming of the Scope field of the Show Variable dialog box. This problem has been corrected.
 - There was a problem with the dimming of the Full button in the Examine Code dialog box. This problem has been corrected.

Programmer Release Notes

4.8 Debugger Notes

- The state of the With Operands and Full buttons in the Examine Code dialog box did not reflect a previous SET MODE OPERANDS command. This problem has been corrected; the state now reflects a previous SET MODE OPERANDS command.
- The Deposit Code... dialog box generated a syntax error when depositing instructions while running an Ada program. This problem has been corrected; syntax errors are no longer generated.
- The SET OUTPUT NOTERMINAL command was enabled but should have been disabled. This problem has been corrected; the SET OUTPUT NOTERMINAL command is now disabled.
- The debugger signaled an access violation error if you tried to use the Exit option in the Processes dialog box when the Process field was empty. This problem has been corrected; the debugger no longer signals an access violation error.
- The debugger signaled 'invalid DLE' errors when attempting to display the source code for an Ada package body. This problem has been corrected; the debugger now displays the source code correctly.

V5.4

- In VMS Version 5.3:
 - If you entered a GO command and specified an address expression after the program terminated, any breakpoints, tracepoints, or watchpoints that you set previously were ignored. This problem has been corrected; any previously set breakpoints, tracepoints, or watchpoints are no longer ignored.
 - The low/high bound expressions in Pascal subrange expressions were restricted to being constants. This problem has been corrected; the subrange expressions no longer have such a restriction.
 - The Parameters subtopic in the online help caused some confusion. To eliminate any confusion, the Parameters subtopic has been deleted from the HELP MESSAGES topic.
 - There were restrictions on specifying a search list for SYS\$LIBRARY. This problem has been corrected; any such restrictions have now been removed.
 - The technique for setting watchpoints in installed writable shareable images was not included in the debugger documentation. The VMS Version 5.4 edition of the *VMS Debugger Manual* outlines the technique for setting watchpoints.

4.8.2 Debugger Problems or Restrictions

This section describes problems or restrictions with the debugger.

Programmer Release Notes

4.8 Debugger Notes

-
- V5.3** **4.8.2.1 \$WAKE Call Followed by \$HIBER Call**
If a program running under the debugger issues a \$WAKE call followed by a \$HIBER call, the debugger hibernates.
-
- V5.2** **4.8.2.2 Debugger Commands Disabled in DECwindows Interface**
When using the DECwindows interface, you can enter debugger commands in the COMMAND box. However, the debugger issues an error message when you try to enter a disabled command. The following commands are disabled in this mode of operation:
- CANCEL WINDOW
 - EXPAND
 - MOVE
 - SELECT/PROGRAM
 - SET MARGINS
 - SET MODE NOSCREEN
 - SET OUTPUT [NO]SCREEN_LOG
 - SET OUTPUT [NO]TERMINAL
 - SET TERMINAL
 - SET WINDOW
 - SHOW MARGINS
 - SHOW TERMINAL
 - SHOW WINDOW
-
- V5.4** **4.8.2.3 Examining LABEL[*n*] for a Code Location**
A command in the form EXAMINE LABEL[*n*] or EXAMINE LABEL(*n*), where LABEL is a label for a code location and *n* is an integer, causes an access violation error. In this case, the debugger does not handle the error.

Note that this problem does not occur when the label marks the start of data storage, as in a MACRO program.
-
- V5.4** **4.8.2.4 MACRO Source Correlation Problem**
The following source-line correlation problem affects the debugging of MACRO programs that invoke the \$FAO or \$FAO_S system service macro or an application-defined macro that contains any of the following directives:
- .IRP
 - .IRPC
 - .REPT
 - .REPEAT

In such cases, the debug symbol table information about source lines and line numbers might be unreliable. During a debugging session, the problem occurs as soon as the macro is invoked and persists as long as execution is within the module in which the macro is invoked.

The screen-mode source display (SRC) is not updated correctly when the debugger suspends execution. The pointer in the source display is typically one or two lines off from the line at which execution is actually suspended.

To determine the exact point at which execution is suspended, use the screen-mode instruction display INST. The arrow in display INST correctly points to the instruction at which execution is suspended. Pressing keypad 7 puts displays SRC and INST side by side so that you can quickly compare the MACRO source code and the decoded instruction stream. Note that the line number information in display INST might be unreliable (as in SRC).

The problem shows up also when the debugger issues a "stepped to...", "break at...", or "trace at..." message, which might indicate the wrong source line.

Specifying a line number in a command might also give incorrect results. For example, do not specify SET BREAK %LINE 34. When setting breakpoints or tracepoints, specify a routine name, a label, or a memory address instead of a line number.

4.8.2.5 Register Window in DECwindows Interface

V5.3

When using the debugger's DECwindows interface, you may see a FIND_LINE internal error when you shrink the REG (register) window vertically with the window's resize button. You must then close the REG window and create a new one. Proceed as follows (the procedure describes one of the possible ways to create a REG window):

- 1 Choose Close Window from the File menu of the REG window.
- 2 Choose Window Setups from the Customize menu.
- 3 Choose the bottom window layout, which positions the REG and INST windows side by side between the SRC and OUT windows.

4.8.2.6 RUN/DETACHED Command Entered After LINK/DEBUG Command

V5.3

If you link a program using the command LINK/DEBUG and then execute the program in a detached process (using the command RUN /DETACHED), the debugger goes into an infinite loop. To circumvent this problem, use the following syntax when specifying the RUN/DETACHED command:

RUN/DETACHED/OUTPUT=device:/INPUT=device:

4.8.2.7 SET IMAGE Command Limitation

V5.3

For some large programs with many program sections (usually caused by many FORTRAN routines with many COMMON blocks) the debugger may receive an internal error during the processing of a SET IMAGE command. In such cases, the image cannot be debugged.

Programmer Release Notes

4.8 Debugger Notes

4.8.2.8 Using Concealed Rooted-Directory Logical Names for Source Files

V5.4

When compiling a program with the /DEBUG qualifier, if you use a rooted-directory logical name to specify the location of the source file, make sure that it is a *concealed* rooted-directory logical name. To create a concealed rooted-directory logical name, use the syntax illustrated in the following command line:

```
DEFINE/TRANSLATION_ATTRIB=CONCEALED ROOTDIR DISK3$: [USER.DIR1.]
```

If the rooted-directory logical name is not concealed and you move the source file to another directory after compilation, you cannot then use the debugger SET SOURCE command to specify the new location of the source file.

4.8.2.9 Using Debugger Commands in DCL Command Procedures

V5.2

Before VMS Version 5.2, you could use a DCL command procedure to invoke the debugger and issue debugger commands contained in that command procedure. For example:

```
$ ! Procedure to run PROG2 under debugger
$ ! control and issue debugger commands
$ !
$ RUN PROG2
SET BREAK %LINE 17
GO
EXIT
$ SHOW SYSTEM
$ LOGOUT
```

Beginning with VMS Version 5.2, you can no longer put debugger commands directly into a command procedure. Instead, you must create a temporary file containing the debugger commands and assign the logical name DBG\$INPUT to point to that file. For example:

```
$ CREATE DEBUG_COMMANDS.TMP
SET BREAK %LINE 17
GO
EXIT
Ctrl/Z
EXIT
$ DEFINE/USER DBG$INPUT DEBUG_COMMANDS.TMP
$ RUN PROG2
$ DELETE DEBUG_COMMANDS.TMP;0
$ SHOW SYSTEM
$ LOGOUT
```

Another way to work around this problem is to establish a single-process debugging configuration, as described in Section 4.8.4.

4.8.2.10 Using the Abort Key or Stop Button After a SPAWN Command

V5.2

If you use the SPAWN command either from the DCL level or from within a debugging session, the debugger abort key or Stop button is disabled after you log out or return from the spawned subprocess.

In the debugger command interface, the abort key is Ctrl/C by default. In the DECwindows interface, the abort button is the Stop button in the main window.

The only way to reenable the abort key or Stop button is to log out and log back in.

V5.0 **4.8.2.11 Using the Debugger on a VAXstation Running VWS**
There is a problem with the handling of Ctrl/Y when the debugger is running in its own window and you have entered the command SET MODE SEPARATE. Ctrl/Y is ignored when the keyboard is attached to the debugger window. To make Ctrl/Y take effect, attach the keyboard to the window from which you invoked the debugger (by pointing at that window with the mouse); then press Ctrl/Y.

V5.2 **4.8.2.12 Using the DECwindows Interface**
The following problems or restrictions are specific to the DECwindows interface:

Note: The startup time for the DECwindows debugger is about 1 minute, depending on network and system load. Do not attempt to manipulate the user interface until the following three debugger windows are completely initialized:

- **Main window**
- **Source window**
- **Output window**

Any attempt to manipulate the debugger interface before these windows are initialized may freeze your workstation.

In addition, when you first invoke the debugger's online help system, it may take up to a minute to display the first help topic. Subsequent help topics are displayed within a few seconds after you invoke them.

- The Modules dialog box does not correctly cancel all modules when used with an Ada program. (To display the Modules dialog box, choose Modules... from the Data menu.)
- The SCROLL/BOTTOM and SCROLL/TOP commands do not work correctly.
- The SCROLL/LEFT and SCROLL/RIGHT commands do not work.
- The SHOW DISPLAY command does not correctly show the values of the window parameters (height, width, x, y) in pixels.
- You can use the EXTRACT/SCREEN_LAYOUT command to save the current settings of the debug windows (by creating a debugger command procedure with the necessary information), but with the following restrictions:
 - The command does not correctly show the values of the window parameters (height, width, x, y) in pixels.
 - The command issues a SET TERMINAL command, which is disabled in the DECwindows debugger.

Programmer Release Notes

4.8 Debugger Notes

- The command issues a CANCEL DISPLAY/ALL command, which causes the debugger to produce an internal error if not deleted from the command procedure.
- The command does not issue the window information for the COMMAND box or the main window.
- You cannot use the DISPLAY/NODYNAMIC command to make a debugger window a NODYNAMIC window.
- The command SELECT/OUTPUT PROMPT does not cause debugger output to be sent to the PROMPT window.
- The default window size of the predefined register window, REG, is not large enough to display three columns of register information.
- If any of the text shown in the main window is sufficiently large to run off the right edge of the window, the main window fails to expand to show all the information. In that case, the STOP button also disappears. To work around this problem, expand the main window such that it is wide enough to show the STOP button.
- When the debugger issues a debugger command as you manipulate the DECwindows interface with the mouse, the command is echoed in the COMMAND box in all uppercase characters, even if the language is case sensitive.
- When using the Windows dialog box to modify a debugger window, be sure to select the name of the window from the list box of the dialog box. When displayed, the Windows dialog box might have a window name preselected in the Window field, but relying on this preselection can produce internal errors. (To display the Windows dialog box, choose Windows... from the Customize menu.)
- Do not give the PROMPT display the SCROLL attribute; this causes an access violation error. Also, if you use the PROMPT display with the OUTPUT attribute, debugger output is lost.
- You cannot select more than one entry from the list box of a dialog box.
- If you edit a command recalled in the COMMAND box, press Ctrl/E (move to end of line) before pressing Return. This prevents the COMMAND box from breaking the command line into two parts, which causes syntax errors.
- If you are reading mail in DECwindows Mail, displaying a debugger dialog box may fill in the text of the last read mail message into the dialog box. To work around this problem, select an object in a debug window, then cancel the selection by clicking on it again.
- You can enter only integer data in the Height, Width, X and Y text fields of the Windows dialog box. Do not use expressions, for example, %PAGE. (To display the Windows dialog box, choose Windows... from the Customize menu.)

4.8.2.13 Vector-Support Restrictions and Problems

V5.4

The following are problems and restrictions with the debugger's support for vectorized programs:

- When the programming language is BLISS, COBOL, or RPG, you must specify a type qualifier to deposit into %VMR. For example,


```
DEPOSIT/QUADWORD %VMR = %HEX 0FFFFFFF
```
- When the programming language is PL/I, COBOL, or DIBOL, the command EXAMINE %VMR displays %VMR as an array of bits instead of as a hexadecimal quadword. Type the command EXAMINE/HEX/QUADWORD %VMR to obtain the default behavior for other programming languages.
- When the vector mode is synchronized (that is, if you have entered the command SET VECTOR_MODE SYNCHRONIZED), the debugger suspends execution twice at any breakpoints that were set on vector instructions. To resume execution from such breakpoints, you must enter the GO or STEP command twice.

4.8.2.14 Watchpoints in Installed Writable Shareable Images

V5.3

The technique for setting watchpoints in installed writable shareable images is as follows:

- When using the command interface, enter the command SET WATCH /NOSTATIC.
- When using the DECwindows interface, proceed as follows:
 - 1 Choose Watch... from the Control menu.
 - 2 Choose 'Set nonstatic watchpoint' from the Set/Cancel (upper) option menu. Note that, if you choose 'Set watchpoint', the debugger might display the following message:

```
"Internal debugger error in DBGEVENT\DBG$FIND_EVENT_ID -
no matching event ID"
```

4.8.3 Obsolete Debugger Commands

V5.2

The following debugger commands and command qualifiers are obsolete in VMS Version 5.2 and are no longer documented:

Obsolete Command or Qualifier	Reason
ALLOCATE	The debugger now allocates and deallocates memory automatically. This command now has no effect.
CANCEL EXCEPTION BREAK	This command duplicates the effect of the newer command CANCEL BREAK/EXCEPTION, which better conforms to the general command format for canceling breakpoints.
SET DISPLAY	The DISPLAY command now enables you to create a new display, in addition to modifying an existing display.

Programmer Release Notes

4.8 Debugger Notes

Obsolete Command or Qualifier	Reason
SET EXCEPTION BREAK	This command duplicates the effect of the newer command SET BREAK /EXCEPTION, which better conforms to the general command format for setting breakpoints.
SET MODULE/ALLOCATE	The debugger now allocates and deallocates memory automatically. This qualifier now has no effect.
UNDEFINE	This command duplicates the effect of the newer command DELETE, which conforms to the analogous DCL command DELETE.
UNDEFINE/KEY	This command duplicates the effect of the newer command DELETE/KEY, which conforms to the analogous DCL command DELETE/KEY.

4.8.4 Use of Single-Process (Pre-Version 5.2) Debugging Configuration

V5.2

Prior to VMS Version 5.2, the debugger and the program being debugged ran in the same process. This debugging configuration is referred to as the single-process configuration. Beginning with VMS Version 5.2, the debugger consists of the following two parts that run in separate processes:

- DEBUG.EXE, a relatively small kernel debugger image
- DEBUGSHR.EXE, a larger main image

When you invoke the debugger, if a separate process cannot be created to run the main debugger image, the debugger issues one or more messages and automatically uses the single-process configuration (where the debugger shares a single process with the user program). For example, if quotas are not sufficient to create a subprocess for the main debugger, the messages are as follows:

```
%DEBUG-E-CANTCREATMAIN, could not create the VAX DEBUG subprocess
-SYSTEM-F-EXQUOTA, exceeded quota
%DEBUG-I-SHRPRC, VAX DEBUG will share user process
```

In the single-process configuration, you can use the debugger to debug a program that normally runs in only one process. However, you cannot use the following additional debugger features that are available beginning with VMS Version 5.2:

- You cannot use the multiprocess debugging configuration.
- You cannot use the debugger's DECwindows interface.
- You do not have the benefit of reduced interference between the debugger and the program being debugged.

In the single-process configuration, use the sequence Ctrl/Y and then the DCL command DEBUG to abort a debugger command or program execution from within a debugging session. Using Ctrl/Y returns you to the DCL level. The DCL command DEBUG invokes the debugger, which then displays its prompt. You cannot use the SET ABORT_KEY command to reassign the abort function to another control-key sequence.

The single-process configuration avoids the restrictions that are described in Section 4.8.2.10 and Section 4.8.2.9. If you wish to use the single-process debugging configuration because it avoids the restrictions described in Section 4.8.2.10 and Section 4.8.2.9, you can do so by making the following logical name assignment before invoking the debugger:

```
$ DEFINE DBG$PROCESS NONE
```

Note: To avoid the restrictions of the default configuration (see Section 4.8.2.10 and Section 4.8.2.9), use the single-process configuration (established when the definition of `DBG$PROCESS` is `NONE`) only when necessary. The single-process configuration is unsupported and may not be available in future releases of the debugger. If you encounter any problems using the default or multiprocess configurations (other than those mentioned in these release notes), please submit a Software Problem Report (SPR).

4.9 DECnet — File Access Protocol Extensions

V5.1

The DECnet file access protocol (DAP) support in VMS has been enhanced to properly handle compound document (for example, DDIF) files. This support is fully upward compatible with earlier versions of the DAP protocol. One enhancement has extended the actual length of the SYSCAP field in the CONFIG message. The field is still below the defined maximum field length as defined in earlier versions of the DAP protocol; therefore, the change is compatible with implementations conforming to DAP Versions 5.6 and later.

4.10 DECwindows Notes

The release notes in this section pertain to the DECwindows interface.

4.10.1 CDA Toolkit Notes

The following release notes contain information about the Compound Document Architecture (CDA) Toolkit.

4.10.1.1 Base Converters — Analysis Back End

V5.4

The Analysis Back End converters have the following new functionality for VMS Version 5.4:

- The Analysis Back End now operates on a `CONVERT AGGREGATE` basis instead of the earlier `CONVERT DOCUMENT` basis. With this change, considerably less memory is needed to convert large documents.
- DTIF date-formatted fields are now shown in the comment field with a decimal translation of the DTIF hexadecimal encoding.
- In previous versions, a DTIF decimal translation was attempted, but the resulting value for large integers was incorrect, with no diagnostic warning.

Programmer Release Notes

4.10 DECwindows Notes

With VMS Version 5.4, DTIF variable-integer items containing a value that would translate to a decimal integer of more than 20 digits are displayed in their hexadecimal encoding, along with a diagnostic message indicating that the value is too large for translation.

4.10.1.2 Base Converters — PostScript Back End

V5.4

The PostScript Back End converters have the following new functionality for VMS Version 5.4:

- New support has been added for dithering color and multiplane gray-scale images into black and white images.
- You can now flag single-page Encapsulated PostScript documents as EPSF documents (documents in the format used by a number of desktop publishing programs) by adding the string "EPSF-2.0" to the comment on the first line of the PostScript file. This comment enables PostScript users to recognize the file as an Encapsulated PostScript document.
- If the font is not resident on the PostScript printer, the PostScript Converter can include the font definition in the PostScript output. This support was added for the DECmath fonts used by the DECwrite product.
- The Adobe Symbol Font and the DECmath fonts are now supported. These fonts are used by DECwrite for equation editing.
- The PostScript Converter now outputs the text "(atend)" instead of "(at end)" in the comments at the beginning of the PS file. With this change, the PostScript Converter adheres to EPSF guidelines.
- In previous versions, you could not generate a blank page; new page directive behavior was unpredictable and access violations occurred. This problem has been corrected; you can now generate a blank page.

4.10.1.3 Base Converters — Text Back End

V5.4

In previous versions, the Text Back End incorrectly wrote the file type (extension) into the DDIF\$_DHD_TITLE item twice. This problem has been corrected; the Text Back End now writes the input file name into the DDIF\$_DHD_TITLE item of the document header.

4.10.1.4 Corrections to CDA Toolkit

V5.4

The following corrections have been made to the CDA Toolkit for VMS Version 5.4:

- In previous versions, the LL1_GALLEY_SELECT item in the LL1 aggregate was improperly marked as an inheritable attribute when, in fact, it is a directive and not an attribute. The LL1_GALLEY_SELECT item is now correctly marked as a noninherited directive.
- The ENTER_SCOPE routine requires that an aggregate handle be passed if certain scope codes are specified. However, in previous versions, if an aggregate handle was not passed for the scope codes that required it, the routine did nothing and did not return an error status.

Programmer Release Notes

4.10 DECwindows Notes

This problem has been corrected; `ENTER_SCOPE` returns `CDA$_INVAGGTYP` when a null aggregate handle or an aggregate of the wrong type is passed if the scope code required a specific aggregate type.

- In previous versions, `PRUNE_AGGREGATE` misused an internal data structure and, in certain cases, could have caused an access violation or returned pointers to deallocated memory. This problem has been corrected; `PRUNE_AGGREGATE` no longer causes access violations or returns pointers to deallocated memory.
- A restriction on the specification of external references has been removed. In previous versions, before processing the document, the user or application had to change default directories to the location of the document. Otherwise, some external references could not be found and the CDA Toolkit responded with a fatal error.

This problem has been corrected; CDA documents can now be referenced from a location other than where the document resides without affecting the CDA Toolkit's ability to locate the external references.

- The segment attribute `SGA_COMPUTE_C` enumerated value is now interpreted correctly for the `COPY_COMPUTE` and `REMOTE_COMPUTE` cases. That is, with `COPY_COMPUTE`, the segment content is only imported from an external reference if the segment has no content of its own; with `REMOTE_COMPUTE`, the content from the external reference is always imported and replaces any content the segment already contains.

4.10.1.5 New Attribute Rules for Files (Input Only) — UCX and VMS Services for PCS Servers

V5.4

The ULTRIX Connection Software (UCX) and VMS Services for PCS Servers are disk and file servers that run on VMS systems. These products allow systems other than those running VMS to create and access CDA files (DDIF and DTIF) on VMS systems. The CDA files created by these servers do not contain the RMS semantic tag and do not have the standard CDA (fixed-length, 512-byte) record format. CDA applications based on VMS cannot read the files created through these servers, even though the file contents are valid, because the CDA Toolkit requires that CDA files have a semantic tag and have fixed-length, 512-byte record format.

To allow sharing of CDA files, the CDA Toolkit has relaxed its requirements, on input only, for files created by these servers. The new file attribute rules are as follows:

- If the file has a semantic tag, it must be organized sequentially and have fixed-length, 512-byte records.
- If the file does not have a semantic tag, it must be organized sequentially and be able to accept any record format.

By accepting other file formats, applications that use the CDA Toolkit on VMS systems can access the nontagged files.

Programmer Release Notes

4.10 DECwindows Notes

Note that to invoke the TYPE command, the semantic tag must be correctly set and the file format must be changed to fixed-length, 512-byte records. You can convert a DDIF or DTIF file to the proper format by passing the file through a conversion on the VMS system. An example of setting the semantics and file format on a DDIF file is as follows:

```
$ CONVERT/DOCUMENT infilename.DDIF/FORMAT=DDIF outfilename.DDIF/FORMAT=DDIF
```

The output file from this conversion has the same contents as the input file, but it has fixed-length, 512-byte records and a DDIF semantic tag.

4.10.1.6 New Functionality Added to CDA Toolkit

V5.4

The following functionality has been added to the CDA Toolkit for VMS Version 5.4:

- Performance of the CDA Toolkit has been improved.
- Each aggregate in a DDIF or DTIF document that has a user-longword item can be modified or read by an application; the meaning of the longword is application dependent. In previous versions, the CDA Toolkit did not allow an application to write into this longword. Now, the application program can write into the user-longword item for all aggregates.
- By using the CDA file routines, CDA applications running on VMS can now open and read CDA files on ULTRIX nodes.
- General floating-point numbers in CDA files can be provided (by LOCATE_ITEM) or stored (by STORE_ITEM) using IEEE 754 single-precision and double-precision floating-point formats.
- Style guides can now refer to their own external references. Before merging the style guide's layout and definitions into the root document, you can resolve type and content references by using the definitions in the style guide.
- For documents with deeply nested segments, the processing time has been significantly improved when applying the DDIF input processing options.
- You can now store the user-longword item in any aggregate by using the STORE_ITEM routine when you specify the DDIF\$_USER_CONTEXT item code.

4.10.1.7 New Messages to Clarify Errors

V5.4

New messages have been added to clarify errors encountered during CDA Toolkit document conversion. These meanings and their messages are as follows:

- *CDA\$_INVINPDMN Invalid input domain.*
An invalid input domain was specified for the front end; only DDIF and DTIF are supported as domains.
- *CDA\$_INVOUTDMN Invalid output domain.*
An invalid output domain was specified for the back end; only DDIF and DTIF are supported as domains.

- *CDA\$_DCVNOTFND Domain converter not found.*

The required domain converter could not be found; only DTIF-to-DDIF domain conversion is supported.

- *CDA\$_ICVNOTFND Input converter not found.*

The specified input converter could not be found; verify spelling of format name and check the input converters installed on your system.

- *CDA\$_OCVNOTFND Output converter not found.*

The specified output converter could not be found; verify spelling of format name and check the output converters installed on your system.

V5.4 4.10.1.8 NEXTAggregate Routine — Correct Usage

Certain improper uses of the NEXTAggregate routine that worked coincidentally in previous versions are no longer permitted.

The NEXTAggregate routine is intended to return handles to successive aggregates in a sequence of aggregates. In previous VMS versions, if an address of an aggregate-valued item (as returned by LOCATE_ITEM) was passed to NEXTAggregate, the first aggregate in the sequence was returned. This improper use of NEXTAggregate is not permitted in VMS Version 5.4.

The correct way to access the first aggregate in a sequence is to dereference the item address returned by LOCATE_ITEM.

V5.4 4.10.1.9 REMOVEAggregate Routine — Correct Usage

In VMS Version 5.4 and previous versions, you can use the REMOVEAggregate routine improperly to remove the only remaining aggregate from a sequence of aggregates, or even to remove an aggregate that is not a member of a sequence.

This improper use of REMOVEAggregate is not recommended and may not be permitted in future versions. Use the REMOVEAggregate routine only to remove aggregates from a sequence, not to remove single aggregates from their parents.

4.10.2 DECterm Terminal Emulator

The following release notes contain information about the DECterm terminal emulator.

V5.4 4.10.2.1 Color Table Report — Reporting Problem Corrected

In previous versions of VMS, DECterm responded to the <CSI>2\$u escape sequence with a color table report, which started with <CSI> rather than <DCS>.

This problem has been corrected. With VMS Version 5.4, the color table report is sent in the same format as on the VT340, where D...D is the text containing the color table information. For example:

```
<DCS>2$sD...D<ST>
```

Programmer Release Notes

4.10 DECwindows Notes

V5.4 4.10.2.2 CREATE/TERMINAL Command — Negative Values Problem Corrected
Prior to VMS Version 5.4, you could enter negative values for numeric /WINDOW_ATTRIBUTES subfields of the DCL command CREATE /TERMINAL (for example: WIDTH and HEIGHT). The negative values would be passed to the DECterm controller. The DECterm controller would then fail (crash), and all existing DECterms controlled by that process would fail as well.

This problem has been corrected. With VMS Version 5.4, negative values are ignored and the default value is used.

V5.1 4.10.2.3 DECterm Fonts
All DECwindows terminal fonts are for private use by DECterm and should not be used by other applications. The terminal emulator fonts have several problems:

- Missing or incorrect characters exist in many fonts.
- Spacing problems exist in many fonts.
- The 28-point fonts are not actually two times the height of the 14-point font (DECterm requirement).
- Line-drawing characters in many fonts (particularly Double_Wide) do not join properly.
- The Narrow, Wide, and DoubleWide 14-point fonts at 75 dots per inch (dpi) do not have the same cell size as the Normal and Bold 14-point fonts.

V5.4 4.10.2.4 ReGIS Locator Report — Omitted Coordinates
In response to the R(P(I)) command or in multiple-input mode when the locator position is outside the addressable area, DECterm sends a ReGIS locator report with omitted coordinates. For example, the report omits coordinates when you type the A key to generate the report, where <CR> is a carriage return (ASCII code 13):

```
A[]<CR>
```

V5.4 4.10.2.5 VT52-Mode Cursor Addressing — Restriction Removed
Prior to VMS Version 5.4, a DECterm change restricted VT52-mode cursor addressing (ESC Y) to 24 rows and 80 columns, as on the VT100 and other VT terminals. However, DECterm incorrectly checked for 24 columns and 80 rows; as a result, most VT52-mode applications did not work.

This problem has been corrected. DECterm no longer places a restriction on VT52-mode cursor addressing other than the limits imposed by the window size and the format of the ESC Y sequence.

4.10.3 DECwindows Server and Driver Notes

The release notes in this section pertain to DECwindows server and driver.

-
- V5.4** **4.10.3.1 Client-Number Problem Corrected**
- The DECwindows server is limited internally to a maximum of 32 clients. (Note that all DECterm windows that are started by the Session Manager count as a total of one client). However, other limitations may further restrict the number of clients that can connect to a server.
- One of these limitations related to the enqueue quota that the server was running has been corrected. In previous versions, a maximum of 15 local clients were allowed if the server had an enqueue quota of 30. Now a maximum of 32 local clients is allowed.
- If you need a larger enqueue quota on the server, you can increase the value of the SYSGEN parameter PQL_DENQLM (the default enqueue limit for any process that does not specify the enqueue limit when it is created).
-
- V5.3-1** **4.10.3.2 Misspelled Cursor Screen Boundary \$QIO Constant**
- In VMS versions prior to Version 5.3-1, there is a misspelled \$QIO constant (IO\$K_DECW_CURSOR_BOUNDRIES) listed in the definition file (\$DECWDEF). The file is called by the definition macro \$DECWGBL found in SYS\$LIBRARY:DECW\$DRIVER.MLB.
- Server/driver code written with the misspelling for versions prior to Version 5.3-1 should be corrected to reference the constant, IO\$K_DECW_CURSOR_BOUNDS, as specified in the *VMS DECwindows Device Driver Manual*. The constant is the function modifier in the \$QIO call that sets the boundaries of the display to which the cursor is limited.
-
- V5.3** **4.10.3.3 Problems and Restrictions**
- The following problems and restrictions exist in the DECwindows server:
- Configuring windows with bit gravity other than FORGET can result in corruption within the window. To recover from this type of corruption, expose the windows entirely or shrink to icon and click on the icon to open the window you just reconfigured.
 - When the server is replaying events collected during a synchronous grab, motion events may be time-stamped with the current time during replay instead of the time the event was collected.
-
- V5.3-1** **4.10.3.4 Setting a Cursor Pattern \$QIO Call**
- The SET CURSOR PATTERN function of the \$QIO service has been enhanced to support varying size cursor patterns. This new format allows the caller to load cursor patterns without regard to the hardware format. In a multiscreen (multihead) environment, hardware with different-sized cursors can now be used. For more information, refer to the *VMS DECwindows Device Driver Manual*.

Programmer Release Notes

4.10 DECwindows Notes

V5.4 4.10.3.5 Virtual Memory Space Problem with Large Pixmaps

The DECwindows display server allocates virtual memory for storing pixmap data. If very large pixmaps are allocated, the virtual memory space for the server can exceed the server's page file quota and cause the server to fail. This problem is especially noticeable on the VAXstation 3100/SPX and the 24-plane versions of the VAXstation 3520 and 3540 systems:

- The VAXstation 3100/SPX allows larger pixmaps to be allocated than do other color servers.
- The 24-plane versions of the VAXstation 3520 and 3540 allocate more memory for the same size of pixmap.

When the failure occurs, pressing Return displays a user name prompt at the top of the workstation screen.

You can increase the amount of page file quota for the server by defining the logical name `DECW$SERVER_PAGE_FILE` to be a value larger than 25,000. For example, enter the following command line to increase the page file quota to 50,000:

```
$ DEFINE DECW$SERVER_PAGE_FILE 50000
```

You can then define the logical name in the command file `SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.COM`. If you do not have a `SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.COM` file, see the `SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.TEMPLATE` file.

For the server to use its full page file quota, the system parameter `VIRTUALPAGECNT` must be at least as large as the page file quota. You might also need to increase the size of your page file to accommodate any increase.

V5.1 4.10.3.6 VAXstation Configurations

The following notes pertain to VAXstation configurations:

- On a VAXstation 2000 workstation, the keyboard and mouse serial lines are TTA0 and TTA1, respectively. Terminal operations such as SET/SHOW TERMINAL and SET HOST DTE do not work for these devices. The terminal lines are TTA2 and TTA3.
- The server looks for a number at the end of its process name. If it finds a number, it considers that number to be its server number and listens for connections on that number rather than on 0. The default value is 0. This is normally resolved by the DECwindows startup command files.
- You cannot depend on the values of BlackPixel and WhitePixel being 0 and 1. Their values will differ depending on the hardware.
- Put Image is restricted to a maximum width of 1024 pixels for GPX servers.

Programmer Release Notes

4.10 DECwindows Notes

- The X11 protocol allows the server to “arbitrarily transform” the components of a cursor in order to meet the requirements of the display. Since neither the VAXstation 2000 nor the VAXstation II monochrome workstation supports recoloring cursors, you should not expect the colors you specify for the cursor to actually be reproduced on the hardware.
- The DECwindows server contains a facility called a **condition handler** that detects problems that might otherwise cause the server to stop and tries to let the server continue. When the condition handler intercepts a problem like this, it sends an “Implementation” error to the client, disconnects the client, or both.

When the condition handler recovers from an error, the server might lose resources such as memory. Therefore, after a number of these interceptions, the condition handler broadcasts a warning message to all users on the workstation indicating that the server may be running in a degraded mode and suggesting that it be restarted. If you get messages like this, you should restart the server at the next convenient opportunity. Enter the following command from a privileged account (it may be one logged into a DECwindows terminal emulator window) to restart the server:

```
$ @SYS$MANAGER:DECW$STARTUP RESTART
```

- A number of X11 protocol requests and corresponding XLIB requests take an unsigned word (“short” in C) as a width or height argument. A common application mistake is to calculate a width or height incorrectly and obtain a small negative number. The protocol interprets this as a large unsigned number. The DECwindows server does not deal with large widths and heights correctly in many cases. You may get an “Implementation” error returned by the condition handler, or by the server directly detecting that the number is too large.

Note that the numbers the server has trouble with are generally greater than 32,767, or combinations of coordinates and width/height greater than 32,767. Coordinates in the range of any supported display devices are much smaller than this number.

After the server runs out of buffer space when trying to write events, errors, or replies to a client, it hangs in HIB state and retries the write to the client periodically. The server does not service other clients while it is in this state.

After a timeout period, the server disconnects the offending client and services the remaining clients. This problem can happen when a client is working slowly and generating a lot of requests. However, the most common occurrence is when the client is being debugged.

4.10.4 Display PostScript Notes

The following release notes pertain to Display PostScript.

Programmer Release Notes

4.10 DECwindows Notes

V5.4 4.10.4.1 Ada Bindings Not Available
With VMS Version 5.4, Ada bindings are not available for Display PostScript.

V5.4 4.10.4.2 VAX Calling Standard Bindings — DPS\$PRINTF Routine Not Implemented
In the VAX Calling Standard Bindings, the DPS\$PRINTF routine is documented but not implemented.

4.10.5 SET DISPLAY Command

V5.3 When displaying graphics to an ULTRIX server, the ULTRIX node name must be placed within quotation marks (""") if the name contains any lower case characters.

For example, to display on node "uuu," type the following command:

```
$ SET DISPLAY/CREATE/TRANSPORT=TCPIP/NODE="uuu"
```

The quotation marks are optional for a node name composed of all uppercase characters.

4.10.6 User Interface Language (UIL) Compiler Notes

The following release notes contain information about the User Interface Language (UIL) compiler.

V5.4 4.10.6.1 Built-In Tables — Additions
The built-in UIL arguments listed in Table 4–1 have been added for VMS Version 5.4.

Table 4–1 Additional UIL Resources

Widget Object	Additional Arguments
attached_dialog_box	direction_r_to_l
caution_box	direction_r_to_l
color_mix	hue_label, light_label, sat_label, black_label, white_label, gray_label, full_label, option_label, his_label, rgb_label, color_model, direction_r_to_l, grab_key_syms, no_resize, take_focus
command_window	direction_r_to_l
dialog_box	direction_r_to_l
menu_bar	menu_extend_last_row, direction_r_to_l
popup_menu	menu_extend_last_row, direction_r_to_l
pulldown_menu	menu_extend_last_row, direction_r_to_l
radio_box	menu_extend_last_row, direction_r_to_l
work_area_menu	menu_extend_last_row, direction_r_to_l

(continued on next page)

Table 4-1 (Cont.) Additional UIL Resources

Widget Object	Additional Arguments
list_box	spacing, direction_r_to_l
file_selection	auto_unmanage, auto_unrealize, direction_r_to_l
selection	auto_unmanage, auto_unrealize, direction_r_to_l
help_box	direction_r_to_l
main_window	direction_r_to_l
message_box	direction_r_to_l
option_menu	direction_r_to_l
popup_attached_db	direction_r_to_l
popup_dialog_box	direction_r_to_l
scale	direction_r_to_l
scroll_bar	direction_r_to_l
scroll_window	direction_r_to_l
separator	direction_r_to_l
window	direction_r_to_l
work_in_progress_box	direction_r_to_l

V5.4

4.10.6.2 Convenience Translation Files Made Public

The XUI Toolkit UIL convenience translation files `dwtxlattext.uil` and `dwtxlataarg.uil` that were public on UWS but not on VMS have been made public in VMS Version 5.4. These UIL files contain all toolkit output text. You can use the UIL files to translate the default English text to other languages.

V5.4

4.10.6.3 Corrected Problems

With VMS Version 5.4, the following UIL problems have been corrected:

- In previous versions, multiple-segment compound strings in UIL were not created with the specified or default character set. This problem has been corrected; multiple-segment compound strings are now created in this context.
- In previous versions, constraint attributes were allowed on nonconstraint widgets. This problem has been corrected; constraint attributes are no longer allowed on nonconstraint widgets.
- In previous versions, compiling a UIL file that compiled and produced a usable UID file could produce a UID file that caused the application's `DRMFetchWidget` call to return with a `DRMNotFound` status. This problem has been corrected; an incorrect status is no longer returned.
- In previous versions, when the default character set is `iso_hebrew` or `iso_hebrew_lr`, concatenating two simple strings generated the following error:

```
%UIL-F-SUBMIT_SPR, internal error - submit an SPR
```

This problem has been corrected; an error no longer occurs.

Programmer Release Notes

4.10 DECwindows Notes

- In previous versions, specifying DEC_KANJI or DEC_HANZI as the default character set generated the following error:

Support for this character set may be removed in a future release.

This problem has been corrected; an error no longer occurs.

- In previous versions, large value tables could not fit into DRM context buffers. This condition generated the following error:

```
%UIL-F-SUBMIT_SPR, internal error - submit an SPR
```

This problem has been corrected; an error no longer occurs.

4.10.6.4 Specifying Multiline Compound Strings

V5.3

In VMS Version 5.3, the UIL compiler does not consistently process newline characters (\n) that are embedded in compound strings. The effect of a newline character that is embedded in a compound string is now solely dependent on the character set specified, and the result may not always be the creation of a multiline compound string.

To guarantee the creation of a multiline compound string, you must use the SEPARATE clause in the COMPOUND_STRING function and the concatenation operator (&) to join the segments into a multiline compound string. The SEPARATE clause takes the form SEPARATE = *boolean-expression*, and implements the newline character for VMS Version 5.3. For example, in VMS Version 5.1, the UIL compiler would generate a multiline compound string from the following input:

```
value
    sample_string : compound_string ("Hello\nWorld!");
```

To guarantee the same result in VMS Version 5.3, you must input the following:

```
value
    sample_line1 : compound_string ("Hello", separate = true);
    sample_line2 : compound_string ("World!");
    sample_string : sample_line1 & sample_line2;
```

To retain VMS Version 5.1 behavior of the newline character (\n) in a compound string, compile your UIL specification file using the /VERSION qualifier as follows:

```
$ UIL/VERSION=V1 MY_FILE.UIL
```

See the *VMS DECwindows User Interface Language Reference Manual* for more information on the COMPOUND_STRING function. See the *VMS DECwindows User Interface Language Reference Manual* for more information on the /VERSION qualifier.

4.10.7 VAX C Definition File Requirements

V5.1

During the VAX C installation procedure, you have the option to extract the VAX C definition files (.h files), or leave the .h files in the text library. If you extract the definition files, you are able to use #include control lines of the following form:

```
#include <filename.h>
```

All DECwindows sample C programs assume that the .h files were extracted; the samples contain `#include <module_name.h>` notation for the included files. The DECwindows programming documentation also makes this assumption.

VAX C should be installed using the option to extract the library modules.

If you have already installed VAX C and you did not extract the .h files, the DECwindows sample C programs do not work. To correct this problem, reinstall VAX C and extract the .h files.

4.10.8 Xlib Routines — Notes

The following release notes pertain to Xlib routines.

4.10.8.1 Corrected Xlib AST Reentrance Problem

V5.4

In previous releases of VMS, the following AST routine sequence could cause an infinite wait or data corruption:

- 1 A non-AST routine waits for an event.
- 2 An AST routine executes and waits for event or reply (for example, the routine calls `XWindowEvent` or `XSync`).
- 3 The AST routine completes.

For VMS Version 5.4, Xlib correctly handles an AST routine that modifies the event queue (adds or removes events by way of Xlib calls) while a non-AST routine is waiting for an event (for example, the AST routine has called `XNextEvent` or `XMaskEvent`). The call to Xlib from the non-AST routine now completes as soon as there is an event to return.

4.10.8.2 X\$DISPLAY_STRING Routines — Correction

V5.4

Prior to VMS Version 5.4, `XDisplayString` returned an expanded form of the display name that was passed to the `XOpenDisplay` routine. This condition caused pseudodevices (for example, `WSA30`) to be translated to a `NODENAME::SERVER.SCREEN` format. In some circumstances, this extra translation precluded opening a second connection to a server.

This problem has been corrected. The `XDisplayString` and `X$DISPLAY_STRING` routines now operate as specified in the Xlib documentation. The routines return the string that was passed to the `XOpenDisplay` routine or, if null, a logical name translation of `DECW$DISPLAY`.

4.10.8.3 Xlib Programming Restriction

V5.1

The `PUT IMAGE` routine does not implement image format conversions for Z pixmap images with a depth greater than one if the depth does not match the depth of the server.

4.10.9 XUI Toolkit Notes

This section contains release notes that pertain to the XUI Toolkit.

Programmer Release Notes

4.10 DECwindows Notes

-
- V5.3** **4.10.9.1 Changes from VMS Version 5.1 to VMS Version 5.2**
In the VMS Version 5.2 XUI Toolkit, both the Intrinsics and DRM (XUI Resource Manager) can be initialized as many times as required by the application. This was not the case for Version 5.1.
- This change is an extension of the MIT R3 intrinsics and should not be used by applications that wish to remain R3 compatible. This change was proposed to the X Consortium for inclusion in the R4 intrinsics.
-
- V5.4** **4.10.9.2 Corrections to the XUI Toolkit**
The following XUI Toolkit corrections have been made for VMS Version 5.4:
- In previous versions, a DECwindows problem introduced in the R3 intrinsics prevented applications that read in user X default files from opening more than one display. This problem has been corrected.
 - In previous versions, a DECwindows problem prevented toggle button gadgets from redrawing themselves after applications had changed their value by way of SetValue (although their values had changed). This problem only affected visible toggle button gadgets modified through SetValue. This problem has been corrected.
-
- V5.4** **4.10.9.3 Discrepancies Between DECwindows Toolkit and MIT R3 Intrinsics**
The version number of the intrinsics supplied as part of the DECwindows kit in VMS Versions 5.1 and 5.2 does not match the MIT R3 intrinsics: The MIT R3 intrinsics version number is 11003; the DECwindows intrinsics version number is 7001. Applications or widgets that switch between the two intrinsics should be aware of this discrepancy.
- The DECwindows routine XtNameToWidget in VMS Versions 5.1 and 5.2 does not conform to the MIT R3 intrinsics. The specification states that the first component of the names parameter is matched against the children of the passed reference widget; instead, the implementation matches the first component of the names parameter against the reference widget, not against the children. To use the DECwindows version, add the name of the reference widget to the beginning of the name list.
-
- V5.4** **4.10.9.4 DRM Routines — Unavailable VAX Bindings**
For VMS Version 5.4, VAX bindings for the following DRM (XUI Resource Manager) routines are not available:
- FETCH COLOR LITERAL
FETCH ICON LITERAL
FETCH LITERAL
- Use the corresponding C bindings for these routines, as are documented in the *VMS DECwindows Toolkit Routines Reference Manual*.
-
- V5.3** **4.10.9.5 Font-Unit Values — Change in Properties**
In previous versions, the XUI Toolkit used the QUAD_WIDTH and RESOLUTION properties of a font to determine the font-unit value for a dialog box. Now, the AVERAGE_WIDTH and RESOLUTION_Y properties are used. The font-unit value for the default DECwindows font remains the same, but the value could be different for any other font.

4.10.9.6 Internal Format of Compound Strings

V5.3

The internal format of compound strings has changed. Compound strings are now stored in CDA format. This change is transparent to applications that treated compound strings as opaque entities.

4.10.9.7 List Box Dynamic Sizing

V5.3

The preferred way to change the list box width is with the Set Values routine. The list box does not support dynamic dimension changes. Therefore, placing list boxes inside attached dialog boxes—with attachments to both the left and right side of the attached dialog box—may, under certain circumstances, lead to the Items Selectable area not spanning the full width of the list box. When an attached dialog box changes size, all dialog boxes for which that box is a root are dynamically resized.

Also, the preferred way to change the list box height is through the ItemsCount attribute. Modifying the height attribute will not reconfigure the number of visible items. For example, doubling the list box height but not modifying the ItemsCount attribute results in a list box only half full of items, with the remaining area left blank.

4.10.9.8 Problems and Restrictions in the XUI Toolkit

V5.3

The following DECwindows problems and restrictions exist:

- There is a problem with widgets that pop up other widgets directly over themselves. The first widget does not see the LeaveWindow event that is produced as the popped-up widget is mapped into the pointer location. This is due to a problem in the MIT R3 intrinsics event dispatching mechanism.

For example, a widget specifies the following translation syntax:

```
<EnterWindow>: highlight()  
<LeaveWindow>: un_highlight()  
<Btn2Down>: popup_menu()
```

When the pointer enters the widget's window, the widget is highlighted. When MB2 is pressed, the pop-up menu is displayed. A LeaveWindow event should be dispatched to deselect the widget as the pointer is moved into the pop-up menu. This LeaveWindow event is not delivered and the widget is left in the highlighted state when the menu pops down.

This problem will be corrected in a future release.

- XUI Toolkit dialog boxes perform an XGrabKey on the Tab key so that they can “synchronously” transfer focus to the next child within the dialog box. If a dialog box receives a Tab key while the Toolkit is “filtering” events (for example, while another modal dialog box is up), the original dialog box does not see the Tab event and never calls XAllowEvents to unfreeze the keyboard. You must quit the application and restart it to unfreeze the keyboard.

This problem will be corrected in a future release.

Programmer Release Notes

4.10 DECwindows Notes

- If you run an application linked against an earlier Toolkit version and an application linked against the current Toolkit, you will not be able to cut from one and paste to the other. No error message appears. Applications formerly shared a common clipboard, but applications linked against the current Toolkit share a separate and independent clipboard. This incompatibility was necessary to fix a more serious source of potential problems with the clipboard and steps have been taken to ensure that no similar clipboard incompatibility will exist among future versions.
- Unlike other Toolkit callbacks, the destroy callback only returns two arguments: widget id and tag. The reason argument is NULL. Applications therefore should avoid setting destroy callbacks to call 'general' callback routines (handling numerous actions such as activate, arm, and disarm) that depend on a reason argument. For Ada developers this may be particularly important, since Ada requires that all declared arguments be passed.
- In certain circumstances, the help widget's list box selectable area does not span the entire width of the widget. However, all items may still be selected by clicking the mouse button on the item text.
- There is a problem with the `DwtToggleButtonSetState` routine that affects toggle buttons with on/off pixmaps. If the widget has not been realized, `SetState` correctly updates the toggle button value but not the on/off pixmap. If the widget has been realized, it may display the wrong pixmap. To circumvent this problem, use the `SetValues` mechanism instead setting `DwtNvalue` to `True`. This correctly updates the pixmap as well as the toggle button value, regardless of whether the widget is realized.
- Pop-up dialog boxes with no icon button (`DwtNnoIconify` set true) are initially created as icons (`DwtNiconic` true). Not only does the icon pop-up not have an icon box (and therefore cannot be popped up), but operations such as `SetInputFocus` on the icon pop-up cause an access violation.
- Using Pascal to call `DWT$TOGGLE_BUTTON_SET_STATE` causes a problem with the `newstate` parameter. This parameter is defined as a Pascal Boolean variable. Although the data type allocation size is a byte, only the low-order bit is significant. However, the Toolkit routine tests the entire byte for a value of zero to indicate False.

For example, the following example does not work properly with a `BUTTON_SET` value of `True`.

```
DWT$TOGGLE_BUTTON_SET_STATE (widget, (NOT Button_Set), FALSE);
```

The following workaround forces the byte to be tested:

```
DWT$TOGGLE_BUTTON_SET_STATE (Widget,  
    (UAND((NOT Button_Set)::UNSIGNED,1))::BOOLEAN, FALSE);
```

- When you use accelerators on push-button and toggle-button gadgets, only the first gadget child of a widget parent may have a # operator, such as `#override`, in its button accelerator specification. All gadget button accelerators of a widget parent have the same # operator as the first gadget child.

See the *VMS DECwindows Guide to Application Programming* for information about specifying an accelerator for a pushbutton or toggle-button gadget and for information on the syntax of creating an accelerator specification.

V5.4

- A menu problem was traced to the menu not clearing its submenu field if the subwidget was destroyed. That problem has been corrected; the menu now sets the submenu field to null.

However, if an application destroys the original submenu and then immediately updates the field with the new menu (through `SetValues`), the submenu field is updated with the new widget id. Only phase-1 destroy has completed, and the menu does not know that its original submenu has been destroyed. When phase 2 executes (later in `MainLoop`), the parent menu is informed that its submenu has been destroyed, and it sets that field to null even though it now points to the new widget. This condition can result in the following error:

```
X - NOT A VALID WINDOW
```

To work around this problem, go back to `MainLoop` and wait for the phase-2 destroy to complete and the menu to clear its submenu field before updating through `SetValues`.

This problem will be corrected in a future release.

- Right-to-left compound strings are displayed left to right in the dialog-box title resource. This problem will be corrected in a future version.
- `CSText` widget class and record definition are defined in the MIT C bindings but are not included in the VMS bindings. Normally, this problem would affect only applications developers who are writing their own widget as a subclass of the `CSText` widget using the VMS bindings. However, widget writing is supported in the MIT C bindings only.

4.10.9.9 Redrawing Widgets

V5.4

Generating widget and gadget redraws (exposes) by calling `SetValues` without visual changes is not supported.

In previous versions of XUI, some widgets incorrectly redisplayed after `SetValues` whenever the argument list contained a visual field, even if that field did not change. For example, an application could initiate a pushbutton redraw by passing an unchanged borderwidth in `SetValues`. These unnecessary redraws have been eliminated.

Applications can now redraw widgets either by changing a visual field or by calling `XClearArea` on the widget window.

4.10.9.10 Selection Push Buttons

V5.3

Setting the Ok and Cancel push button labels to null or empty strings does not remove the push buttons, but instead results in blank labels.

Programmer Release Notes

4.11 Device Driver Debugging — POOLCHECK Enhancements

4.11 Device Driver Debugging — POOLCHECK Enhancements

V5.4 Special pool-checking code in the VMS memory allocation and deallocation modules helps isolate certain system failures reliably and quickly.

With VMS Version 5.4, the pool-check mechanism has been enhanced to facilitate the debugging of a device driver. A new bit (bit 5) in the **flags** byte of the POOLCHECK system parameter validates the look-aside list deallocation operation. In addition, the POOLCHECK bugcheck can now sense several types of failures beyond a corrupted packet.

For more information on the POOLCHECK enhancements, see the *VMS Device Support Manual*.

4.12 Device Support Notes

The release notes in this section pertain to support for devices connected to a VAX processor.

4.12.1 INVALIDATE_TB Macro — New Version

V5.4 VMS Version 5.4 supplies a new version of the INVALIDATE_TB macro in SYS\$LIBRARY:LIB.MLB. The INVALIDATE_TB macro allows privileged code, such as a device driver that is not supplied by Digital, to modify a single page-table entry (PTE) while any translation buffer entry that maps it is invalidated. The macro also allows such code to invalidate the entire translation buffer. The previous version of this macro was described in the *VMS Device Support Manual* for VMS Version 5.0.

Code that uses the old version of the INVALIDATE_TB macro will continue to run properly, without reassembly, in VMS Version 5.4. If you are reassembling privileged code that invokes this macro in VMS Version 5.4, you must use the syntax of the new version of the macro.

The arguments accepted by the new INVALIDATE_TB macro differ from those accepted by the previous version. The new macro also contains enhanced syntax checks that generate assembly-time warning messages when code improperly invokes the macro. Because code generated by the new version may be slightly larger than that produced by the previous version, code reassembly may generate “Branch destination out of range” errors.

For a description of the syntax and operation of the new version of the INVALIDATE_TB macro, consult the *VMS Device Support Manual*.

4.12.2 SPI\$MAP_BUFFER Macro — PRIO=HIGH Parameter Added

V5.4 The PRIO=HIGH parameter has been added to the SPI\$MAP_BUFFER macro to allow for low-priority and high-priority mapping of buffers that are used for Small Computer System Interface (SCSI) transfers.

You can use the `PRIO=HIGH` parameter to avoid a deadlock condition that can occur when several devices are operating simultaneously and one encounters an error. If the error requires a `REQUEST SENSE` command be sent to the failing device, invoking the `SPI$MAP_BUFFER` macro can stall and lead to a deadlock under certain circumstances. The solution to the deadlock problem is to issue the `SPI$MAP_BUFFER` macro with the high-priority parameter for the `CHECK CONDITION` command, as follows:

```
SPI$MAP_BUFFER PRIO=HIGH
```

By default, `SPI$MAP_BUFFER` operations are processed at low priority. If mapping resources are not available for an `SPI$MAP_BUFFER` operation, a low-priority request stalls, and a high-priority request fails with `SS$_INSFMAPREG` status.

For more information about the `SPI$MAP_BUFFER` macro, see the *VMS Device Support Reference Manual*.

4.12.3 **SYSGEN — New Order for Unit Control Blocks (UCBs)**

V5.4

In previous versions of VMS, SYSGEN added unit control blocks (UCBs) to the device data block (DDB) list in the order in which the UCBs were created.

With VMS Version 5.4, SYSGEN adds UCBs to the DDB list in order of unit number. This change should not affect application software.

4.13 **Directory Size Limitation Removed — Effect on RSX-11 Compatibility Mode**

V5.2

VMS Version 5.2 contains file-system enhancements that substantially improve the performance of large directories. In addition, the former restriction of directory size to 1024 blocks has been removed. Directory size is now limited only by the availability of contiguous disk space.

Lifting this size restriction may have an impact on programs that execute in RSX-11 compatibility mode and use wildcards in file operations. Because the RSX-11 wildcard context cannot address directories larger than 1024 blocks, such programs cannot process files in a directory that exceeds 1024 blocks in size. This is a permanent restriction of the RSX-11 compatibility mode environment, and users must avoid using directories exceeding 1024 blocks in this environment.

4.14 **DSA Disk Drivers — Alternate Host Information Change**

V5.4

The VMS operating system maintains alternate host information for disks that can be accessed by two or more paths. In previous versions of VMS, the `DSDRIVER` and `DUDRIVER` disk class drivers updated the alternate host information for disks that are accessed through the Mass Storage Control Protocol (MSCP) server. This update reflected the most recent server that had become available.

Programmer Release Notes

4.14 DSA Disk Drivers — Alternate Host Information Change

Load balancing for the MSCP server is now available. With load balancing, the alternate host information is not used to locate another path to the disk during failover. Instead, all eligible paths are considered and the least-loaded path is selected. This change does not affect failover, but it might result in a change in the information returned by the \$GETDVI system service or the SHOW DEVICE command. For example, the secondary path might refer to a serving node that has been shut down. Alternate path information for direct paths (those that do not involve the server) continue to be updated when new direct paths appear.

For more information on MSCP server load balancing, see the *VMS VAXcluster Manual*.

4.15 GBBDriver — New Output Driver

V5.2

A new output driver (GBBDriver) has been developed for the VAXstation 3520 and 3540 systems. The existing output driver GCBDriver supports output for the new VAXstation 3100 system. For the server programmer, a new function modifier (GB\$K_LEGGS_WAIT_FOR_PKT) in the output-buffered and output-direct FDT \$QIO call has been added in the output driver. This expands the existing packet wait \$QIO function in output drivers to support the wait function for Low End Graphics Subsystem (LEGSS) packets.

4.16 I/O Device Driver Notes

The release notes in this section pertain to the I/O device drivers supplied as part of the VMS operating system.

4.16.1 Logical End-of-Volume Detection Now Always in Effect

V5.4

In previous versions of VMS, logical end-of-volume detection on skip file or skip record operations was in effect only when the tape was mounted foreign.

With VMS Version 5.4, logical end-of-volume detection is always in effect. (Note that when a tape is not mounted foreign, use of the logical I/O functions IO\$_SKIPFILE and IO\$_SKIPRECORD is not supported.)

For more information on logical end-of-volume detection, see the Magnetic Tape Drivers chapter in the *VMS I/O User's Reference Manual: Part I*.

4.16.2 Opening a Sequential-Media File Now More Efficient

V5.4

With VMS Version 5.4, opening a file on sequential media (magnetic tapes and RV20 disks) is more efficient than with previous VMS releases. For information on lookups by file ID, see the ACP-QIO Interface chapter in the *VMS I/O User's Reference Manual: Part I*.

4.16.3 User EOT Mode Correction

V5.4

In previous versions of VMS, you could not use the user end-of-tape (EOT) mode as documented in the ACP-QIO Interface chapter of the *VMS I/O User's Reference Manual: Part I*.

This problem has been corrected for VMS Version 5.4. You can now use the user EOT mode to write beyond the end of the tape when a volume is mounted with the magnetic tape ancillary control process (ACP).

4.17 IF-THEN-ELSE Construct and \$STATUS Symbol

V5.2

Most DCL commands generate status values when they complete. However, there are several commands that do not change the values of \$STATUS and \$SEVERITY; for example, IF, GOTO, CONTINUE, and STOP. A list of these commands can be found in the *Guide to Using VMS Command Procedures*.

The IF-THEN-ELSE-ENDIF construct was incorrectly setting \$STATUS, which masked the resulting status condition from commands executed within the block. In VMS Version 5.2, this has been fixed to maintain the last value of \$STATUS set inside an IF-THEN-ELSE-ENDIF block. A command procedure can then test the value of \$STATUS following the ENDIF command.

4.18 Languages — Reinstallation Required

V5.2

To most easily use a VMS system routine from a given high level language, language-specific definitions need to exist for the following:

- The routine
- The routine's error messages
- Routine-specific constants

For most languages, these definitions are built from files that exist in the VMS system when the language is installed. If the language is installed before a system routine exists, the language-specific definitions for that system routine will not be present.

The routines PROCESS_SCAN, DEVICE_SCAN, and the Clusterwide Process Services (CWPS) extensions did not exist prior to VMS Version 5.2. The libraries for a language that was installed prior to VMS Version 5.2 do not contain the definitions that should be used when using the new system routines from that language.

To build the language-specific definitions required for using PROCESS_SCAN, DEVICE_SCAN, and the CWPS extensions, reinstall each language for which you wish to use these routines.

Programmer Release Notes

4.19 LIBRARIAN Routines — Locate Mode Caution

4.19 LIBRARIAN Routines — Locate Mode Caution

V5.0 When you use the Librarian Utility (LIBRARIAN) in locate mode, the contents of a descriptor may not point to an internal LBR buffer for subsequent LBR routine calls.

4.20 Linker Utility — Open Image Library Support

V5.4 If you are developing portable applications using Network Application Support (NAS) products, a second image library, similar to SYS\$LIBRARY:IMAGELIB, is used. The second image library contains components that conform to NAS conventions rather than to VMS conventions. By default, the Linker Utility does not search this library because it may contain symbols that do not conform to the VMS global symbol-naming rules.

If you want the open image library to be searched, you must define the logical name LNK\$OPEN_LIB with any string value that is not null. If the LNK\$OPEN_LIB logical name is defined at link time, the Linker Utility searches SYS\$LIBRARY:OPEN_LIB in the same way it searches SYS\$LIBRARY:IMAGELIB. The open image library search is in addition to any other searches, and it is done after user libraries are searched and before other system libraries are searched.

The Linker Utility library search is conducted in the following order:

- 1 User libraries, if defined with LNK\$LIBRARY_nnn
- 2 SYS\$LIBRARY:OPEN_LIB, if LNK\$OPEN_LIB logical is defined
- 3 SYS\$LIBRARY:IMAGELIB, unless /NOSYSSHR is specified
- 4 SYS\$LIBRARY:STARLET, unless /NOSYSLIB is specified

4.21 Message Router Version 3.0 Installation

V5.0 The following list contains problems that occur when you use Message Router Version 3.0 with VMS Version 5.0 and subsequent versions:

- The Message Router VMS Gateway (MRGATE) requires the MAIL image to run with SYSPRV. Unlike VMS Version 4.n, VMS Version 5.0 does not install the MAIL image with SYSPRV. Therefore, if you are running MRGATE on VMS Version 5.0, log in to the SYSTEM account and use the following command to assign SYSPRV to the MAIL image:

```
$ INSTALL REPLACE MAIL/PRIVILEGES=SYSPRV
```

- If you are running the Directory Service part of Message Router Version 3.0 on VMS Version 5.0, you may see the following error messages:

```
DDS-E-OPSYS, Operating system interface error  
LIB-E-BADBLOCK, bad block address
```


Programmer Release Notes

4.21 Message Router Version 3.0 Installation

These messages indicate that part of the virtual memory is not being released. You will see these error messages when new nodes join the Directory Service network and when the Directory Service servers are running (for example, when you use the MBMAN SUSPEND command). These are erroneous messages; the Directory Service continues working.

- The SUBMIT command works differently in VMS Version 5.0 from how it works in VMS Version 4.n. In Version 4.n, any logical names specified in the /LOG qualifier are translated at submission time. In Version 5.0, the logical names are translated when the job starts, at which point the logical names may not have been defined.

If you are using exception reporting in Message Router Version 3.0, the change to the SUBMIT command in VMS Version 5.0 can cause the exception reporting batch submission to fail. The batch jobs are entered on the batch queue, but the jobs fail and do not leave a log file indicating the reason for failure, because no logical name is defined for the log file.

To avoid this problem, edit the command procedure that starts up the exception reporting batch jobs (SYS\$COMMON:[SYSMGR]MB\$\$ER_START.COM) as follows:

- 1 Change the /LOG qualifier of the SUBMIT command from /LOG=MB\$SCRATCH:MB\$'component'_'node'.LOG to /LOG=MB\$ROOT:[MB\$SCRATCH]MB\$'component'_'node'.LOG
 - 2 Change the /LOG qualifier of the SUBMIT command from /LOG=MB\$SCRATCH:MB\$NET_"mb\$\$mgmnt_node".LOG to /LOG=MB\$ROOT:[MB\$SCRATCH]MB\$NET_'mb\$\$mgmnt_node'.LOG
- The *Message Router Version 3.0 Release Notes* state that the verification procedures require that DECnet and the Queue Manager be running. However, the verification procedures also require that at least one queue be defined in the system startup command procedure. To do this, add the following command line to your system startup procedure:

```
$ INITIALIZE/QUEUE/BATCH queue-name
```

Refer to the *VMS DCL Dictionary* for more information about initializing batch queues.

4.22 Modular Executive Notes

The release notes in this section pertain to the Modular Executive.

Programmer Release Notes

4.22 Modular Executive Notes

4.22.1 Instructions for Loading a Site-Specific Executive Loaded Image

V5.0

This section contains step-by-step instructions for preparing a site-specific executive loaded image, for loading this image into the operating system, and for removing the image. The example creates an MTACCESS.EXE executive loaded image. A similar example can be found in SYS\$EXAMPLES:DOD_ERAPAT.MAR on the VMS operating system.

Preparing and Loading the Executive Loaded Image

1 Create the source module MTACCESS.MAR.

a. Include the following macro to define system service vector offsets:

```
$SYSVECTORDEF ; Define system service vector offsets
```

b. Use the following macros to define the system service entry point:

```
SYSTEM_SERVICE MTACCESS, - ; Entry point name  
                <R2,R4>, - ; Registers to save  
                MODE=KERNEL,- ; Mode of system service  
                NARG=6 ; Number of arguments
```

The instruction following the preceding macros is the first instruction of the \$MTACCESS system service.

c. Use the following macros to declare the desired program sections (PSECT):

```
DECLARE_PSECT EXEC$PAGED_CODE ; Pageable code PSECT  
DECLARE_PSECT EXEC$PAGED_DATA ; Pageable data PSECT  
DECLARE_PSECT EXEC$NONPAGED_DATA ; Nonpageable data PSECT  
DECLARE_PSECT EXEC$NONPAGED_CODE ; Nonpageable code PSECT
```

2 Assemble the source module by using the following command:

```
$ MACRO/OBJ=MTACCESS MTACCESS+SYS$LIBRARY:LIB.MLB/LIB
```

3 Link the module to create an MTACCESS.EXE executive loaded image. You can link the module by using a command procedure as follows:

```
LINK /NOSYSSHR/NOTRACEBACK -  
      /SHARE=MTACCESS -  
      /MAP=MTACCESS /FULL /CROSS -  
      /SYMBOL=MTACCESS -  
      SYS$INPUT/OPTION  
MTACCESS, -  
SYS$LIBRARY:STARLET/INCLUDE:(SYS$DOINIT), -  
SYS$SYSTEM:SYS.STB/SELECTIVE  
VECTOR_TABLE=SYS$SYSTEM:SYS.STB
```

Programmer Release Notes

4.22 Modular Executive Notes

```
COLLECT=NONPAGED_READONLY_PSECTS/ATTRIBUTES=RESIDENT, -  
EXEC$NONPAGED_CODE  
COLLECT=NONPAGED_READWRITE_PSECTS/ATTRIBUTES=RESIDENT, -  
EXEC$NONPAGED_DATA  
COLLECT=PAGED_READONLY_PSECTS, -  
EXEC$PAGED_CODE  
COLLECT=PAGED_READWRITE_PSECTS, -  
EXEC$PAGED_DATA  
COLLECT=INITIALIZATION_PSECTS/ATTRIBUTES=INITIALIZATION_CODE, -  
EXEC$INIT_CODE, -  
EXEC$INIT_000, -  
EXEC$INIT_001, -  
EXEC$INIT_002, -  
EXEC$INIT_PFNTBL_000, -  
EXEC$INIT_PFNTBL_001, -  
EXEC$INIT_PFNTBL_002, -  
EXEC$INIT_SSTBL_000, -  
EXEC$INIT_SSTBL_001, -  
EXEC$INIT_SSTBL_002
```

- 4 Prepare the executive loaded image to be loaded.
 - a. Copy MTACCESS.EXE images produced by the preceding link command into the SYS\$LOADABLE_IMAGES directory. Note that privilege is required to put files into this directory.
 - b. Add an entry for the MTACCESS.EXE image in the SYS\$UPDATE:VMS\$SYSTEM_IMAGES.IDX data file.

You add an entry by using the SYSMAN Utility. The SYSMAN command is as follows:

```
SYSMAN SYS_LOADABLE ADD _LOCAL_ image_name  
/LOAD_STEP = {INIT | SYSINIT} -  
/SEVERITY = {WARNING | SUCCESS | FATAL | INFORMATION} -  
/MESSAGE = "error message text"
```

The image name defines the file specification of the image to be loaded. The default directory is <SYS\$LDR> and the default file type is EXE.

The /LOAD_STEP qualifier has the following two images:

INIT	Image to be loaded by the system initialization code.
SYSINIT	Image to be loaded by the SYSINIT process.

The /SEVERITY qualifier has the following parameters:

WARNING	If error loading the image, output the error message and continue processing.
SUCCESS	Continue even if there is an error loading the image. No message is issued.
FATAL	If error loading the image, output the error message and BUGCHECK.
INFORMATION	Always output the message and continue.

The /MESSAGE qualifier is a supplied error message text to be issued under the appropriate condition.

Programmer Release Notes

4.22 Modular Executive Notes

For example, you can add the following entry to VMS\$SYSTEM_IMAGES.IDX for MTACCESS.EXE:

```
$ SYSMAN SYS_LOADABLE ADD _LOCAL_ MTACCESS -  
_ $ /LOAD_STOP = SYSINIT -  
_ $ /SEVERITY = WARNING -  
_ $ /MESSAGE = "failure to  
load installation-specific - $MTACCESS service")
```

This entry specifies that the MTACCESS.EXE image is to be loaded by the SYSINIT process during the bootstrap. If there is an error loading the image, the following messages are printed on the console terminal:

```
%SYSINIT-E-failure to load installation-specific $MTACCESS service  
-SYSINIT-E-error loading <SYS$LDR>MTACCESS.EXE, status = "status"
```

- c. Invoke the SYS\$UPDATE:VMS\$SYSTEM_IMAGES.COM command procedure to generate a new system image data file. The system bootstrap uses this image data file to load the appropriate images into the system.
- d. Shut down and reboot the system, which loads the site-specific MTACCESS.EXE executive loaded image into the system. Subsequent calls to the \$MTACCESS system service use the site-specific routine.

As the default, the system bootstrap loads all images described in the system image data file (VMS\$SYSTEM_IMAGES.DATA). You can disable this function by setting the special SYSGEN parameter LOAD_SYS_IMAGES to 0.

Removing the Executive Loaded Image

You can remove an executive loaded image by using the following procedure:

- 1 Use the following SYSMAN command (based on the specific example in the preceding instructions).

```
$ SYSMAN SYS_LOADABLE REMOVE _LOCAL_ MTACCESS
```

- 2 Repeat steps c and d from instruction 4.

4.22.2 New Description for \$MTACCESS

V5.0

The \$MTACCESS service allows sites to provide their own routine to interpret an output accessibility field in VOL1 and HDR1 labels of ANSI-labeled magnetic tapes. The site can override the default routine by providing an MTACCESS.EXE executive shareable image.

4.23 National Character Set (NCS) Change

V5.4

Prior to VMS Version 5.4, the NCS Norwegian collating sequence had A-umlaut (Ä) collating as if it were two separate characters: A and E.

This problem has been corrected. The collating value for A-umlaut is now treated as if it were AE-ligature (Æ).

4.24 Processor Register Definition Symbols

V5.0

The following internal processor registers (IPRs) are no longer common to all VAX processors; their definitions have been removed from \$PRDEF.

- NICR—Interval Clock Next Interval Register
- ICR—Interval Clock Interval Count Register
- TODR—Time of Day Register
- ACCS—Accelerator Control Status Register
- ACCR—Accelerator Reserved
- PME—Performance Monitor Enable

New CPU-specific processor register definition macros have been added to STARLET.MLB to define the CPU-specific IPRs. The macro names have the format \$PRxxxDEF, where xxx is the number associated with the processor (for example, \$PR780DEF will define PR780\$_ACCS).

The only legitimate references to these registers are in CPU-dependent code. These references must use the new CPU-dependent IPR definitions.

Note, however, that time-wait loops must never directly refer to the clocks. They must use a time-wait macro that is independent of the CPU. A new, CPU-independent, time-wait macro called TIMEDWAIT has been added to LIB.MLB. This should eliminate any need for hand-coded time-wait loops.

There should no longer be any references to PR\$_ICR or PR\$_TODR to do time-wait loops. TIMEDWAIT allows for up to six special-purpose instructions to be placed in its timing loop. However, the loop timing is based on having one BITx and one conditional branch instruction embedded within the loop. Therefore, if you have a loop with no embedded instructions, you may want to adjust the TIME argument accordingly. A good rule of thumb is to add 25 percent to the time argument if the loop has no embedded instructions.

To refer to PR\$_TODR for logging purposes, use EXE\$READ_TODR and EXE\$WRITE_TODR. These two new, loadable, CPU-dependent routines have been added for code that must reference this type of value.

4.25 RA92 DSA Disk — Defining Symbol DT\$_RA92

V5.3-2

Beginning with VMS Version 5.3-2, the RA92 DSA disk is supported. However, the symbol DT\$_RA92, which would normally be defined by the \$DCDEF macro, is not defined in the system libraries. This definition will be included in a future version of the VMS operating system. As an interim measure, programs that need to identify RA92 drives should define the symbol DT\$_RA92 as 64 (decimal).

4.26 Record Management Services (RMS) Notes

The release notes in this section pertain to the VMS Record Management Services (RMS).

Programmer Release Notes

4.26 Record Management Services (RMS) Notes

4.26.1 Appending to Shared Sequential Files — Restriction Removed

V5.3-1

The temporary restriction against using the deferred write (DFW) option when appending records to shared sequential files, which was documented in the *VMS Version 5.0 Release Notes*, has been removed. The problem of potential corruption to sequential files was corrected with VMS Version 5.2.

4.26.2 Expiration of RMS Files — Change

V5.4

In previous versions of VMS, DCL commands that opened RMS disk files interfered with the expiration scheme. Under certain circumstances, the following DCL commands could reset the Expiration Date and Time, unintentionally postponing the point at which the disk file was considered expired:

- DIRECTORY
- DELETE (where the file is not deleted because of selection criteria)
- PURGE (where the file is not purged because of selection criteria)
- Several lexical functions (those that open a file to accomplish a function)

With VMS Version 5.4, these DCL commands use a new RMS XABITM, XAB\$_NORECORD, which suppresses the update of the Expiration Date and Time and no longer interferes with the expiration scheme. Thus, sites cannot rely on the behavior of these DCL commands to delay expiration of RMS files. (Note that the expiration scheme is not enabled by default on VMS disk volumes.)

Applications that perform maintenance functions on RMS disk files can use the new RMS XABITM. For more information on the XAB\$_NORECORD XABITM, see the *VMS Version 5.4 New Features Manual*.

4.26.3 RAB\$V_ASY Qualifier Now Supported for Process-Permanent Files

V5.4

In previous versions of VMS, the RAB\$V_ASY qualifier had been documented as supporting process-permanent files when, in fact, it did not. With VMS Version 5.4, the RAB\$V_ASY qualifier is supported for process-permanent files such as SYS\$INPUT; operations on process-permanent files are now identical to operations on image files.

Incorrect use of the RAB\$V_ASY qualifier could result in RMS\$_BUSY or RMS\$_ACT errors. If these errors occur, verify the setting of RAB\$V_ASY or use the \$WAIT service to synchronize with I/O completion.

4.26.4 RMS Indexed File Local Buffers — New Default

V5.4

With VMS Version 5.4, the number of local buffers that are allocated to an RMS indexed file has increased slightly.

Instead of always allocating two buffers (the default when none was specified), the default is now based on the depths of the indexes of the indexed file. The new default is two plus the maximum depth index for either the primary index or any secondary index. For shared indexed files, each buffer is associated with a lock and therefore consumes an ENQLM unit. A process that opens a large number of shared indexed files can receive an "Exceeded ENQLM" error (SS\$_EXENQLM).

This new default is expected to save I/Os in many untuned situations. For situations in which the new default is not desired, one of the following options can be used to change the number of local buffers:

- The application can override the new indexed file default for a file access by using the RAB\$_MBF field.
- The system manager can set the processwide indexed file default with the SET RMS/INDEXED/BUFFER=*n* command.
- The system manager can set the systemwide indexed file default with the SET RMS/INDEXED/SYSTEM/BUFFER=*n* command.

Note, however, that a minimum of two buffers is still always allocated for an indexed file record stream.

4.26.5 RMS Statistics Restrictions

V5.1

The following restrictions apply to the use of RMS statistics:

- RMS statistics cannot be gathered on files residing on ODS-1 (On Disk Structure Level 1) disks.
- RMS statistics are not maintained for process-permanent file accesses. Process-permanent file accesses are those that are not released on image rundown. These are typically accesses resulting from the DCL OPEN command. If a file is accessed both as a process-permanent file and by a user image, then only operations done by the user image are counted in the RMS statistics. Enable or Disable the gathering of RMS statistics with the SET FILE/[NO]STATISTICS command.

4.26.6 XAB\$_NUL Option — Clarification

V5.0

The *VMS Record Management Services Manual* states that you can use the XAB\$_NUL option with string-type keys only. Actually, you can use this option with all key types. Note however, that RMS sets the null value to 0 for keys other than string-type keys.

Programmer Release Notes

4.27 Recovery Unit Journaling Notes

4.27 Recovery Unit Journaling Notes

The release notes in this section pertain to recovery unit journaling.

4.27.1 Appending to Write-Shared Sequential Files

V5.2

If records are appended to a write-shared sequential file containing fixed-length records using recovery unit journaling, and the recovery unit is not committed (either `SYS$ABORT_RU` is called, or a system failure occurs), recovery overwrites each appended record in the recovery unit with zeros. Subsequent readers of the file will read these zeroed records. This behavior is necessary because other shared accessors may also have appended records to the file following the zeroed records, and those other record numbers cannot be changed. There is no support for deleted records in sequential files.

4.27.2 Exclusive Access to Recovery Unit Journalled Files — Restriction

V5.0

You might receive an error message if you attempt to insert or update a record in an indexed file and all of the following conditions are true:

- The file is marked for recovery unit journaling.
- The file has a secondary key that does not allow duplicate secondary keys.
- The file is opened for exclusive access.

The error message is as follows:

```
%RMS-F-DUP, duplicate key detected (DUP not set)
```

To prevent this error, open the file for shared access.

Digital expects to correct this problem in a future release of VMS.

4.27.3 Moving Recovery-Unit Journalled RMS Indexed Files to Systems Running VMS Version 4.7 and Earlier

V5.2

There is a restriction on moving indexed files that have been marked for recovery unit journaling, modified within a transaction, and then unmarked for recovery unit journaling, to systems running VMS Version 4.7 and earlier where VAX RMS Journaling is not installed.

You must first make a new copy of the file using the Convert Utility. You can then transfer the converted copy of the file to the system running VMS Version 4.7 or earlier.

4.27.4 **SET FILE/AI_JOURNAL or SET FILE/BI_JOURNAL Command — Correct Usage**

V5.0

If you use the SET FILE/AI_JOURNAL or the SET FILE/BI_JOURNAL command without the CREATE keyword and you specify a journal that is already being used, the SET command cannot open the journal. The SET command issues the FLK (file currently locked by another user) error message. The SET FILE command does not allow you to re-mark a file for journaling using the same journal specification without the CREATE keyword.

If you want to create a journal with the same name as a previously created journal, use the CREATE keyword with the SET FILE/AI_JOURNAL or the SET FILE/BI_JOURNAL command.

The following example illustrates how to create a journal with the same name as a previously created journal:

```
$ CREATE X.X  
Ctrl/Z  
$ SET FILE X.X /BI_JOURNAL=(CREATE, FILE=X_JOURNAL)  
$ SET FILE X.X /BI_JOURNAL=(CREATE, FILE=X_JOURNAL)
```

4.27.5 **SYSGEN Parameter PIOPAGES — Change in Usage**

V5.4

Earlier versions of VAX RMS Journaling required that the value of the SYSGEN parameter PIOPAGES be raised for applications that have many simultaneously active transactions or many files joined to a single active transaction.

With VMS Version 5.4, the PIOPAGES value no longer needs to be raised. If you previously raised PIOPAGES to avoid the RMS DME (dynamic memory exhausted) error, you should now be able to restore PIOPAGES to its previous value.

4.27.6 **VFC Format Sequential Files Partially Supported for Before-Image or Recovery Unit Journaling**

V5.0

You cannot execute \$UPDATE on variable fixed-length control (VFC) sequential files when using before-image or recovery unit journaling. The VFC sequential file format is indicated by the symbolic value FAB\$C_VFC in the FAB\$B_RFM field of the FAB. The following error condition results if you attempt to execute \$UPDATE on a VFC format sequential file marked for before-image journaling, or on a VFC format sequential file modified within a recovery unit:

```
JNS, operation not supported by RMS journaling
```

For more information about this error message, see the *VAX RMS Journaling Manual*.

Digital expects to remove this restriction in a future release of VMS.

Programmer Release Notes

4.27 Recovery Unit Journaling Notes

4.27.7 WRTJNL_BIJ Error Message

V5.0

The WRTJNL_BIJ error message may return a zero completion status value (STV) rather than the message DEVICEFULL if the device on which the before-image journal resides becomes full when RMS is trying to write to the before-image journal. If you receive a zero STV value in this situation, submit a Software Performance Report (SPR). The *VAX RMS Journaling Manual* lists the information you need to include with VAX RMS Journaling-specific SPRs.

4.28 Run-Time Library (RTL) Notes

The release notes in this section pertain to the Run-Time Library.

4.28.1 LIB\$CREATE_VM_ZONE Routine — New Flags Added

V5.4

The **flags** argument to the LIB\$CREATE_VM_ZONE routine is the address of a longword integer that contains flag bits that control various options.

With VMS Version 5.4, the flags listed in Table 4–2 have been added to the **flags** argument:

Table 4–2 New Flags Added to LIB\$CREATE_VM_ZONE

Bit	Flag Value	Description
6	LIB\$M_VM_NO_EXTEND	Zone cannot be extended.
7	LIB\$M_VM_TAIL_LARGE	Add areas larger than extend-size to the end of the area list.

The LIB\$M_VM_NO_EXTEND flag prevents the zone from growing beyond its initial size. If you specify the LIB\$M_VM_NO_EXTEND flag, you must also specify an initial size. In this case, the extend size is ignored.

Allocations that are larger than the extend-size area can result in the creation of new areas. If the LIB\$M_VM_TAIL_LARGE flag is set, these new areas are added at the end of the area list. This addition provides better memory reuse when allocating small blocks and very large blocks from the same zone.

Bits 8 through 31 are reserved and must be 0. For more information on the **flags** argument, see the *VMS RTL Library (LIB\$) Manual*.

4.28.2 LIB\$DECODE_FAULT Use with Vector Processor

V5.4

LIB\$DECODE_FAULT does not explicitly pass the state of the vector processor as parameters to the user action routine. To alter the vector state, the user action routine must execute vector instructions. The user action routine must be careful to leave the vector processor in a known state, because LIB\$DECODE_FAULT does not reset the vector processor to the state it was in before the exception.

4.28.3 LIB\$GET_VM Routine Performance Degradation

V5.0-1

In VMS Version 5.0, performance degradation in the LIB\$GET_VM routine occurred under the following conditions:

- The program created a zone, defaulting all the parameters.
- The program allocated many small pieces of memory that totaled a large portion of memory.
- The program made few calls to LIB\$FREE_VM.

This degradation problem has now been corrected.

4.28.4 LIB\$VERIFY_VM_ZONE and LIB\$SHOW_VM_ZONE Zone Analysis Problem

V5.1

The routines LIB\$VERIFY_VM_ZONE and LIB\$SHOW_VM_ZONE can, under specific conditions, incorrectly determine that a virtual memory zone is corrupted.

If a program causes a zone to have one or more 8-byte blocks of free memory, the routine LIB\$VERIFY_VM_ZONE incorrectly returns the status LIB\$_BADZONE. In the same situation, LIB\$SHOW_VM_ZONE reports that the area free list is corrupted with an invalid block size.

A sample of the incorrect output from LIB\$SHOW_VM_ZONE is shown in Example 4-1. Note that it is the zone analysis that is incorrect; the memory zone itself is not corrupted.

Example 4-1 Sample Output of Routine LIB\$SHOW_VM_ZONE

```
Zone ID = 00073600, Zone name = ""
Algorithm = LIB$K_VM_FIRST_FIT
Flags = 00000000
Initial size = 16 pages Current size = 16 pages in 1 area
Extend size = 16 pages Page limit = None
Requests are rounded up to a multiple of 8 bytes,
naturally aligned on 8 byte boundaries
8 bytes have been freed and not yet reallocated
144 bytes are used for zone and area control blocks, or 1.7% overhead
Area Summary:
      First      Last      Pages      Bytes not yet
      address   address  assigned  allocated
      -----
00073E00 00075DFF          16          8184
Scanning Free List for Area at 00073E00
**** ERROR -- invalid block size ****
Link Analysis for Current Block:
```

(continued on next page)

Programmer Release Notes

4.28 Run-Time Library (RTL) Notes

Example 4-1 (Cont.) Sample Output of Routine LIB\$SHOW_VM_ZONE

	Previous	Current	Next
	-----	-----	-----
Block adr	: 00062EF0	00073E00	00062EF0
Forw link (abs):	00073E00	00062EF0	00073E00

Block size = 8192

Block contents:

```
① 00000000 00000000 00062EF0 00000008 ....?..... 00000 00073E00
00000000 00000000 00000000 00000000 ..... 00010 00073E10
(510 matching lines skipped)
```

- ① The key to recognizing that this is the known problem is the value 00000008 in the first longword of the block dump.

This problem will be corrected in a future release of the VMS operating system.

4.28.5 Obsolete PPL\$ Routines and Replacements

V5.4

The following PPL\$ routines are obsolete as of VMS Version 5.4:

- PPL\$INITIALIZE — Superseded by PPL\$CREATE_APPLICATION, which provides greater functionality
- PPL\$FIND_SYNCH_ELEMENT_ID — Superseded by PPL\$FIND_OBJECT_ID

4.28.6 PPL\$TRIGGER_EVENT Routine Memory Problem Corrected

V5.4

Prior to VMS Version 5.4, a problem in the PPL\$TRIGGER_EVENT routine caused PPL\$ to lose a small amount of its internally managed shared memory during each call. After a large number of calls to PPL\$TRIGGER_EVENT, PPL\$ ran out of internal shared memory. The exact number of calls varied, depending upon the size of the PPL application specified in the call to PPL\$INITIALIZE. At this point, any calls to routines that created PPL objects or that caused the process to block returned a PPL\$_INSVIRMEM error.

This memory problem has been corrected; the PPL\$TRIGGER_EVENT routine no longer loses memory.

4.28.7 RTL Language Support Enhancements

V5.0

The following enhancements have been made to the Run-Time Library for language support:

- COBOL Support for ANSI 85. File Status variable returns ANSI 85 values and new error messages to go with ANSI 85 file status codes.

- FORTRAN descending key support. Key direction specified in open statement.

4.28.8 RTL Screen Management—SMG\$CREATE_PASTEBOARD and SMG\$CREATE_VIRTUAL_KEYBOARD Restriction

V5.0

If a program calls both SMG\$CREATE_PASTEBOARD and SMG\$CREATE_VIRTUAL_KEYBOARD, make sure that SMG\$CREATE_PASTEBOARD is called first. Otherwise, the program does not function correctly.

4.28.9 Scalar Math Routines — SYS\$SHARE:UVMTHRTL.EXE Now Obsolete

V5.4

As of VMS Version 5.4, one shareable image provides the scalar math routines for all VAX systems; SYS\$SHARE:UVMTHRTL.EXE is no longer needed.

The logical names MTHRTL and UVMTHRTL are both defined to point to SYS\$SHARE:MTHRTL.EXE. Therefore, you should remove any references to MTHRTL and UVMTHRTL (or to UVMTHRTL.EXE) in site-specific files. These logical names are unlikely to be defined in future versions of VMS.

4.29 Self-Modifying Item Lists with \$GETxxx Services

V5.2

A problem can occur if you use self-modifying item lists with the following services:

- \$GETDVI
- \$GETDVIW
- \$GETJPI
- \$GETJPIW
- \$GETLKI
- \$GETLKIW
- \$GETMSG
- \$GETQUI
- \$GETQUIW
- \$GETSYI
- \$GETSYIW
- \$GETTIM
- \$GETTUI

When any one of these services collects data, it makes multiple passes through the item list. The number of passes needed depends both on which item codes are referenced and the state of the target process. If the item list is self-modifying—that is, if the addresses for the output buffers in the item list point back to the item list—the service replaces the item-list information with the collected data. Therefore, incorrect data may be returned or unexpected errors may occur when the service reads the item list again.

Programmer Release Notes

4.29 Self-Modifying Item Lists with \$GETxxx Services

A program using self-modifying item lists that appears to work normally can fail when a system has processes that are swapped out of memory, or when a process is on a remote node. System load or the order of the item list entries can also cause such a program to fail.

To prevent confusing errors, Digital recommends that you not use self-modifying item lists.

4.30 SET HOST/DTE/DIAL Command — DMF-32 Controller Problem

V5.0

The SET HOST/DTE/DIAL command does not work with the DMF-32 controller because the modem sends a response character to the host when it detects a carrier signal. The DMF-32 controller drops any input until it sees the carrier signal.

One solution is to modify the example autodialer provided in SYS\$EXAMPLES:DT_DF03.MAR to perform an IO\$_SENSEMODE!IO\$_M_RD_MODEM \$QIO to check for a carrier signal. If set, the autodialer should assume success and continue.

4.31 TLZ04 Tape Drive — Defining Symbol DT\$_TLZ04

V5.4

Beginning with VMS Version 5.4, the TLZ04 tape drive is supported. However, the symbol DT\$_TLZ04, which would normally be defined in the \$DCDEF macro, is not defined in the system libraries. This definition will be included in a future release of the VMS operating system. As an interim measure, programs that need to identify TLZ04 tape drives should define the symbol DT\$_TLZ04 as 32 (decimal).

4.32 VAX 9000 Computer — BI Device Drivers Conformance Requirement

V5.4

Developers of code for VAXBI options that run on the VAX 9000 computer must be aware that field-type instructions are not legal in I/O space. Other VAX implementations allowed this type of illegal reference. Because of the prefetch address process in the hardware, field-type instructions fail on the VAX 9000 computer.

You should check all current BI drivers for references to instructions that use bit-field operands to control status registers (CSRs) or to other I/O space regions. Examples of such instructions are as follows:

- BBS, BBC, BBSS, BBSC, BBCS, BBCC
- FFS, FFC
- EXTV, EXTZV
- CMPV, CMPZV
- INSV

4.33 VAX Ada Run-Time Library Notes

The release notes in this section pertain to the VAX Ada run-time library.

4.33.1 'STORAGE_SIZE Attribute Change

V5.4

When applied to an access type, the predefined attribute 'STORAGE_SIZE returns the actual size of the collection that is allocated for that type, as opposed to the size requested in a length clause. If you do not specify a size for the collection, the value zero is returned.

4.33.2 CALENDAR.SPLIT Improvement

V5.4

With VMS Version 5.4, the accuracy of the values returned by the procedure CALENDAR.SPLIT is improved.

4.33.3 CLOSE Procedures — Change in Implementation

V5.4

With VMS Version 5.4, the implementation of the CLOSE procedures provided by the Ada input/output packages is changed. When you attempt to close a file, one of four results occurs:

- The CLOSE procedure succeeds. The file is closed.
- The exception STATUS_ERROR is raised because the file was already closed.
- An error occurs. An exception such as USE_ERROR is raised, but the file is left open.
- An error occurs. An exception such as USE_ERROR is raised, and the file is closed.

When an error occurs, you can determine whether the file is open or closed: first handle the exception, then call the Ada input/output function IS_OPEN.

Note: Prior to VMS Version 5.4, only the first three results could occur.

4.33.4 CONSTRAINT_ERROR Now Raised in Place of NUMERIC_ERROR

V5.4

With VMS Version 5.4, the CONSTRAINT_ERROR exception is raised wherever the Ada standard previously required the NUMERIC_ERROR exception to be raised. This change was made in response to Ada interpretation AI-00387.

Programmer Release Notes

4.33 VAX Ada Run-Time Library Notes

4.33.5 Procedures to Improve AST Handling and Time Slicing

V5.2 With VMS Version 5.2, two procedures have been added to the Ada run-time library to improve AST handling and time slicing:

- EXPAND_AST_PACKET_POOL (Section 4.33.5.2)
- REQUEST_TIME_SLICE (Section 4.33.5.3)

A new Ada package that you can use to call these procedures from your Ada program is described in the following section.

4.33.5.1 Ada SYSTEM_RUNTIME_TUNING Package

V5.2 The ADA SYSTEM_RUNTIME_TUNING package defines interfaces to allow user programs to change various parameters normally chosen by the VAX Ada run-time library that affect Ada program execution.

You can use the SYSTEM_RUNTIME_TUNING package to call the following new procedures from your Ada program. The package is supplied in the VAX Ada Version 2.0 predefined program library.

```
package SYSTEM_RUNTIME_TUNING is
  subtype AST_PACKET_REQUEST_TYPE is NATURAL range 0 .. 1_048_576;
  procedure EXPAND_AST_PACKET_POOL (
    REQUESTED_PACKETS      : in AST_PACKET_REQUEST_TYPE;
    ACTUAL_NUMBER          : out NATURAL;
    TOTAL_NUMBER           : out NATURAL);
  pragma INTERFACE(RTL, EXPAND_AST_PACKET_POOL);
  pragma IMPORT_PROCEDURE(EXPAND_AST_PACKET_POOL,
    "ADA$EXPAND_AST_PACKET_POOL");

  procedure REQUEST_TIME_SLICE (REQUESTED_VALUE :   DURATION);
  pragma INTERFACE(RTL, REQUEST_TIME_SLICE);
  pragma IMPORT_PROCEDURE(REQUEST_TIME_SLICE,
    "ADA$SET_TIME_SLICE");
end SYSTEM_RUNTIME_TUNING;
```

4.33.5.2 EXPAND_AST_PACKET_POOL Procedure

V5.2 The EXPAND_AST_PACKET_POOL procedure adds more AST packets to the pool of packets used by the VAX Ada AST_ENTRY attribute. The procedure allows you to create 1,048,576 packets. The procedure call succeeds only if there is enough virtual memory to satisfy the request. (A single AST packet currently consumes 32 bytes of dynamic memory.)

When you use the AST_ENTRY attribute to handle an AST, an AST packet is used by the Ada run-time library to hold the AST parameter. An AST packet is in use from the time the AST is delivered by VMS until the receiving Ada task completes the accept statement that receives the AST parameter. If the peak number of ASTs delivered by VMS, but not yet accepted, exceeds the size of the AST packet pool, an unrecoverable error occurs. The error message states that the AST packet pool has been exhausted. The EXPAND_AST_PACKET_POOL procedure can help eliminate the error by increasing the size of the AST packet pool.

Programmer Release Notes

4.33 VAX Ada Run-Time Library Notes

Before you increase the AST packet pool, try to minimize the peak number of AST packets required by your program. To do this, ensure that the accepting task has a high priority and is not delayed by an interaction with any other task before or during the accept statement for the AST. You should consider using the `EXPAND_AST_PACKET_POOL` procedure to increase the size of the pool only after you have concluded that the AST arrival rate is so high that it momentarily exceeds your program's rate of servicing the ASTs.

Note: Using the `EXPAND_AST_PACKET_POOL` procedure does not help if your program's average AST arrival rate is greater than its average AST service rate because eventually your program will run out of AST packets. In this case, you need to revise your program according to your application in order to reduce the AST arrival rate.

Input Parameters

The `REQUESTED_PACKETS` parameter is the minimum number of additional packets desired. More may be allocated because of rounding to the next storage boundary. To determine the current size of the pool, you can specify 0 for the value of the `REQUESTED_PACKETS` parameter.

Output Parameters

The `ACTUAL_NUMBER` parameter indicates the number of packets that were added to the pool.

The `TOTAL_NUMBER` parameter indicates the total number of AST packets in the pool. (Note that this number includes AST packets currently in use for the delivery of an AST.)

Exceptions

The `STORAGE_ERROR` exception is raised if the request could not be satisfied because of insufficient memory. When the `STORAGE_ERROR` exception is raised, an attempt is made to release any AST packets allocated in partial fulfillment of the request.

The `PROGRAM_ERROR` exception may be raised for certain other errors. If the `PROGRAM_ERROR` exception is raised, a chained condition indicates a detailed reason for the failure.

Other exceptions may be raised as well.

4.33.5.3 `REQUEST_TIME_SLICE` Procedure

V5.2

The `REQUEST_TIME_SLICE` procedure conditionally modifies the time-slice setting of the program. This entry point can only make time slicing run faster than it already runs, or enable it if not enabled. The request is always overridden by the value specified by a pragma `TIME_SLICE` in an Ada main program or by a debugger `SET TASK /TIME_SLICE` command.

The `REQUEST_TIME_SLICE` procedure is primarily intended to be called from within an Ada shareable image, or from an object file exported by an `ACS EXPORT` command where it cannot be decided in advance whether there will be an Ada main program. However, you can also use this

Programmer Release Notes

4.33 VAX Ada Run-Time Library Notes

procedure (as often as desired) to override an Ada main program that does not specify a pragma `TIME_SLICE`.

A call to the `REQUEST_TIME_SLICE` procedure has no effect if any of the following is true:

- The `REQUESTED_VALUE` argument is 0.0 or negative (time slicing cannot be disabled by this routine).
- The `REQUESTED_VALUE` argument is greater than a previously specified time-slice value that successfully set the time slice.
- Time slicing has either been activated or turned off by a pragma `TIME_SLICE`.
- A debugger `SET TASK/TIME_SLICE=t` command has been entered.

If none of the preceding conditions is true, the `REQUESTED_VALUE` argument sets the time slice.

In the following cases, the time slice set by a call to the `REQUEST_TIME_SLICE` procedure is overridden:

- An image containing an Ada main program that has a pragma `TIME_SLICE` is activated.
- A debugger `TASK/TIME_SLICE=t` command is entered.
- The `REQUEST_TIME_SLICE` routine is called again with a `REQUESTED_VALUE` greater than zero but less than the value set by the call.

Input Parameters

The `REQUESTED_VALUE` parameter is the requested new time-slice value.

Exceptions

The `PROGRAM_ERROR` exception may be raised for certain errors. If the `PROGRAM_ERROR` exception is raised, a chained condition indicates a detailed reason for the failure.

Other exceptions may be raised as well.

4.33.6 Restriction on END_OF_FILE Function

V5.0

Because of an RMS restriction, the `END_OF_FILE` function in the `SEQUENTIAL_IO` and `SEQUENTIAL_MIXED_IO` packages raises the `USE_ERROR` exception when it is called for a file opened on a remote DECnet node. The other input-output packages are not affected. Until the restriction is removed, you can avoid the error by opening the file with an `OPEN` or `CREATE` procedure and setting the `FORM` parameter to the following:

```
"FILE;SEQUENTIAL_ONLY NO;"
```

Note: Disabling the “sequential only” mode incurs a performance penalty on all network file access.

4.33.7 VAX Vector Registers Not Saved During Task Switches

V5.4 The VAX Ada run-time library does not currently save vector registers during task switches. Therefore, when you call procedures that contain VAX vector instructions, you should make all of the calls from the same task. (Note that the main program is itself a task—the main task.)

4.34 VAX C Notes

The release notes in this section pertain to VAX C.

4.34.1 Mixing D_FLOAT and G_FLOAT Modules

V5.2 If you have VAX C Version 3.0 or later and VMS Version 5.2 or later, you can mix D_FLOAT and G_FLOAT modules within the same program. To do this, include the files `STDIO.H`, `STDLIB.H`, `MATH.H`, and `UNIXLIB.H` in all G_FLOAT modules of the program and compile those G_FLOAT modules with `/DEFINE=("CC$mixed_float")`. Then link all modules against the files `VAXCRTL.EXE` or `VAXCRTL.OLB`.

Note: You must use the include files shipped with Version 3.0 of VAX C or later, or compiling with `/DEFINE=CC$mixed_float` will have no effect.

Modules that use only D_FLOAT variables do not have to contain the above include files. Similarly, they do not need to be compiled with the `/DEFINE` option.

If you are linking a program against the file `VAXCRTL.OLB`, including the preceding definition files in each module and compiling each module with the `/DEFINE` option will produce a minor gain in program efficiency and may significantly reduce the size of any executable produced.

Note: Digital strongly recommends linking against the file `VAXCRTL.EXE` instead of `VAXCRTLG.EXE`. Libraries linked against the file `VAXCRTLG.EXE` may not be usable by programs that use D_FLOAT variables and library routines.

4.34.2 VAX C Run-Time Library — Changes

V5.0 Beginning with VMS Version 5.0, the VAX C Run-Time Library (RTL) contains five new malloc routines (`malloc_opt`). The new malloc routines take advantage of the VMS RTL memory management routines `LIB$GET_VM` and `LIB$FREE_VM`. The performance and capabilities of these routines have been considerably improved. This includes using a zone algorithm that is first fit with no boundary tag. Subsequently, each allocation is zero-filled and aligned on an octaword boundary.

Previous versions of the malloc routines imitated the UNIX version of malloc for memory allocation or deallocation procedures. The new malloc routines do not imitate this behavior. This is exemplified when you try to sequence a freeing of dynamic memory and then try to access that memory.

Programmer Release Notes

4.34 VAX C Notes

Equivalent VMS routines have been created for each malloc routine. For example the VMS operating system routine corresponding to malloc is VAXC\$MALLOC_OPT. To take advantage of this feature, you may find it helpful to include the following macro definitions at the beginning of your program.

```
#define malloc VAXC$MALLOC_OPT
#define calloc VAXC$CALLOC_OPT
#define free VAXC$FREE_OPT
#define cfree VAXC$CFREE_OPT
#define realloc VAXC$REALLOC_OPT
```

Note that these routines are reentrant and may be used in asynchronous system trap (AST) routines.

4.34.3 VAX C Run-Time Library — Error Checking Enhanced

V5.4

For VMS Version 5.4, error checking on the fopen() and freopen() functions has been enhanced. Illegal characters in the key argument now generate an error condition.

Because some systems support "t" as a modifier, many programs use "rt" as an open key. Prior to VMS Version 5.4, the VAX C Run-Time Library (RTL) would interpret "rt" as "r." To catch user errors at the earliest possible point and simplify debugging, VAX C RTL now generates an error for illegal characters.

4.34.4 VAX C Run-Time Library — Socket Routines

V5.2

The socket routines listed in Table 4-3 are available in the VAX C Run-Time Library. Socket routines are used for interprocess communication.

The routines listed in Table 4-3 are a subset of the Berkeley 4.2Bsd Inter-Process Communication library. They provide facilities for creating and using AF_INET (Internet) sockets, and require that Version 1.2 of the VMS/ULTRIX Connection product be installed in order to operate. See the VMS/ULTRIX Connection Version 1.2 release notes for information on how to access these routines. See the *ULTRIX-32 Supplementary Documents, Volume III* for an overview of their use.

Table 4-3 VAX C Run-Time Library Socket Routines

accept	inet_netof
bind	inet_network
connect	inet_ntoa
gethostbyaddr	listen
gethostbyname	nothl
gethostent	ntohs
gethostname	recv

(continued on next page)

Table 4–3 (Cont.) VAX C Run-Time Library Socket Routines

getnetbyaddr	recvfrom
getnetbyname	recvmsg
getnetent	select
getpeername	send
getservbyname	sendmsg
getsockname	sendto
getsockopt	setsockopt
htonl	shutdown
htons	socket
inet_addr	socket_pair
inet_lnaof	vaxc\$get_sdc
inet_makeaddr	

Version 3.0 of the VAX C product contains the include files `in.h`, `inet.h`, `netdb.h`, and `socket.h`, which support the use of the routines listed in Table 4–3. The routines are not currently defined as symbols in the VAX C Run-Time Library, but might be defined in some future release of VMS. When they become defined, they might cause multiple definition warnings in programs linking against the VAX C Run-Time Library which also define these symbols.

4.35 VAX MACRO Notes

The release notes in this section pertain to VAX MACRO.

4.35.1 Caution on Using NOP Instruction as a Delay Mechanism

V5.0

Digital recommends that you do not use the VAX MACRO instruction NOP (No Operation) as a means of delaying program execution.

The delay time caused by the NOP instruction is dependent on processor type. For instance, the VAX 8600, VAX 8650, VAX 8800, VAX 8700, VAX 8550, or VAX 8530 processors execute the NOP instruction more quickly than other VAX processors.

Whenever you must have a program wait for a specified time period, you should use a macro or code sequence that is not dependent on the processor's internal speed. For example, you can use the `TIMEDWAIT` macro, which is documented in the *VMS Device Support Manual*. You can also use the Set Timer (`$SETIMR`) and Wait for Single Event Flag (`$WAITFR`) system services, as described in the *VMS System Services Reference Manual*, to force such delays.

Programmer Release Notes

4.35 VAX MACRO Notes

4.35.2 Use of TIMEDWAIT Macro on MicroVAX and VAXstation 3100 Series Systems

V5.3

The TIMEDWAIT macro, when used on MicroVAX or VAXstation 3100 Series systems, may result in a delay that is up to 10 times the desired value. This occurs because of an error in the routine that calculates the base delay constants (CPU\$L_UBDELAY and CPU\$L_TENUSEC) that are referenced in the TIMEDWAIT macro. Note that the delay will never be less than the value specified.

This error will be corrected in a future release of the VMS operating system.

4.35.3 VAX MACRO Corrected Problems

V5.4

The following VAX MACRO problems that existed in VMS Version 5.0 have been corrected:

- Source-line correlation DST (Debug Symbol Table) records were generated incorrectly for repeat loops (.REPEAT, .IRP, and .IRPC constructs). This problem caused VAX DEBUG to display incorrect source records for all but the first iteration (expansion) of a loop. If a repeat directive indicated more than one iteration, VAX DEBUG continued to display incorrect source records for relatively located instructions outside of the loop.
- In certain cases, the assembler reported a .TRANSFER symbol to VAX SCA as two separate symbols. In addition, the symbol was identified as a read reference instead of the correct address reference.
- Using both the /DIAGNOSTICS and /ANALYSIS_DATA qualifiers during assembly caused the assembler to receive an access violation or to generate a bad .DIA file. The file contained extra REGION/FILE entries for each diagnostic message generated.
- The assembler was using \$PUT to output messages, which resulted in messages not going to SYS\$ERROR when SYS\$ERROR differed from SYS\$OUTPUT. The assembler was changed to use \$PUTMSG instead.
- The assembler was receiving an access violation when DEBUG was enabled in the source file (with a directive) and repeat macros were used afterward. This problem has been corrected; enabling DEBUG in the source file does not break the assembler.
- The assembler was receiving an access violation when a local label was followed by a continuation character (hyphen) and the module was assembled with the /ANA switch. The assembler has been modified.

4.35.4 VAX MACRO Problems

The following is a list of problems that exist in VAX MACRO:

Programmer Release Notes

4.35 VAX MACRO Notes

V5.0

- If there are two or more errors in a VAX MACRO source record, the generated diagnostic message does not point correctly (using the “!” character) at other than the first error.
- When concatenated source files are assembled, the records from each source file are numbered sequentially starting from 1 in the assembly listing. Thus, the same line number can be assigned to multiple source records within a single module. This problem may also occur in the generated DST, causing VAX DEBUG to display incorrect source records. To avoid this problem for debugging purposes, manually concatenate the source files (forming a single input file) before assembly.
- Assembly errors may occur if the string “.REPT” is found in comments included within a .REPT loop.
- The %EXTRACT macro string operator is parsed by the assembler even if it is found in an unsatisfied conditional block or in a comment.
- If the total length of an actual argument in a macro call is larger than that which the assembler can handle (about one block of data), the “%MACRO-E-LINTOOLONG, Line too long” error message is normally issued. For some lengths, however, only messages indicating some kind of incorrect syntax are issued. For still other lengths, the assembler incurs an access violation.
- No diagnostic message is given if a keyword actual argument is associated with a created local label. To avoid this problem, do not specify a keyword actual argument for a formal argument that is a created local label. (Refer to the “Created Local Labels” section of the “Macro Arguments and String Operators” chapter in the *VAX MACRO and Instruction Set Reference Manual*.)
- For branch instructions located in the body of a macro, the assembler does not always generate a diagnostic for a branch out of range.
- The assembler may abort with “%SYSTEM-F-RADRMOD, reserved addressing fault . . .” if the register mask operator used with the .ENTRY directive is incorrectly typed as “M^” or “M” instead of “^M”.
- When evaluating a large expression, the assembler may incorrectly generate a “store word” instruction instead of the correct “store longword” instruction—causing the linker to generate a truncation error.
- The assembler does not correctly compute the required size of an absolute offset unless it is defined in the same PSECT as the code being generated.
- The %LENGTH operator does not work within .REPEAT blocks.
- If the .IIF directive is not contained within one line of source code, then the continuation lines are assembled even if the condition is not satisfied. To avoid this problem, express the condition to be tested and the conditional assembly block completely within the line containing the .IIF directive. If the use of a continuation line is necessary, use the .IF directive instead of .IIF.

Programmer Release Notes

4.35 VAX MACRO Notes

- When `.DISABLE` is specified with multiple items, some of the items may be ignored. To avoid this problem, use individual `.DISABLE` directives for all the items.
- The assembler does not correctly evaluate expressions containing the arithmetic shift operator. For example, the expression `<1@30+1@31-2>` is evaluated as `<<<1@<30+1>>@31>-2>`.

Because there is no operator precedence in VAX MACRO, the first arithmetic shift operation should occur before the add operation. To avoid this problem, you can force the order of evaluation to be correct by placing angle brackets around subexpressions containing the arithmetic shift operator.

- A “branch to subroutine” instruction immediately followed by a directive which computes an expression containing the ASCII operator (^A) may yield a “MACRO-W-DATATRUNC, Data truncation error” diagnostic. To avoid this problem, place a NOP (or any other) instruction immediately after the “branch to subroutine” instruction.
- The assembler may not handle quadword literals (or any literal larger than 32 bits) correctly. For example, the instruction `MOVQ #<5@30>,R0` does not carry the high bit of the number 5 into R1; the bit is lost with no diagnostic message.
- If the last character in an `.IRP` argument list is a comma, the assembler expands the body of the `.IRP` loop $n-1$ times, where n is the number of elements (including null elements) in the list. To avoid this problem, always terminate the `.IRP` argument list with an element that is not null (that is, do not terminate the list with a comma).
- Attempting to assemble a VAX MACRO program using a command line in excess of 512 characters results in a corrupt object file.
- The assembler cannot display listing line numbers greater than 64K. To avoid this problem, do not assemble modules containing more than 64K source records. (Note the total size of a module when concatenating source files.)

V5.4

- When concatenated source files are assembled using the `/ANALYSIS_DATA` command qualifier, incorrect source record numbers are reported to VAX SCA for all but the first concatenated file. These incorrect source record numbers are exactly one less than the correct numbers.
- The following source-line correlation problem affects the debugging of MACRO programs that invoke either the `$FAO` or `$FAO_S` system service macro or an application-defined macro that contains any of the following directives:

```
.IRP
.IRPC
.REPT
.REPEAT
```

In such cases, the debug symbol table information about source lines and line numbers might be unreliable. During a debugging session, the problem occurs as soon as the macro is invoked and persists as long as execution is within the module in which the macro is invoked.

The screen-mode source display (SRC) is not updated correctly when the debugger suspends execution. The pointer in the source display is typically one or two lines off the line at which execution is actually suspended.

To determine the exact point at which execution is suspended, use the screen-mode instruction display INST. The arrow in display INST correctly points to the instruction at which execution is suspended. Pressing keypad key 7 puts displays SRC and INST side by side so that you can quickly compare the MACRO source code and the decoded instruction stream. Note that (as in SRC), the line number information in display INST might be unreliable.

The problem also arises when the debugger issues a “stepped to...”, “break at...”, or “trace at...” message, which might indicate the wrong source line.

Specifying a line number in a command might also give incorrect results. For example, do not specify SET BREAK %LINE 34. When setting breakpoints or tracepoints, specify a routine name, a label, or a memory address instead of a line number.

4.35.5 VAX MACRO Product Requirements

V5.0

VAX MACRO requires that your system run the following software versions:

- VMS Version 4.4 or later
- VAX LSE Version 2.2 or later
- SCA Version 1.1 or later
- VAX DEBUG Version 5.0 or later (for enhanced VAX DEBUG support only)

4.36 VAX Pascal Run-Time Library Enhancements

V5.2

The VAX Pascal Run-Time Library has been enhanced in VMS Version 5.2. These enhancements will become active when used in conjunction with a future release of VAX Pascal, and will be described in detail in connection with that release.

Running a VAX Pascal application linked on VMS Version V5.2 on a system running a previous version of VMS is not supported.

4.37 VAX PL/I Run-Time Library Corrections and Enhancements

V5.2

The following corrections and enhancements have been made to the VAX PL/I Run-Time Library (RTL):

- The COL and DCOL key types are now supported as datatypes for keys in indexed files. This allows PL/I programs to read indexed files containing keys with user-defined collating sequences. You can use the National Character Set utility to create these collating sequences.

Programmer Release Notes

4.37 VAX PL/I Run-Time Library Corrections and Enhancements

- When converting from PICTURE to a FIXED or FLOAT datatype, the pictured value is first converted to an intermediate FIXED DECIMAL value with the same precision and scale as the PICTURE variable. This intermediate value is then converted to the desired FIXED or FLOAT datatype. Previously the picture value was converted to a FIXED DECIMAL (31,0) intermediate value, which resulted in the loss of all fractional digits.
- It is now possible to specify a Boolean value with the record-locking options of the READ statement. The format for the options is NOLOCK(Boolean), LOCK_ON_READ(Boolean), LOCK_ON_WRITE(Boolean), MANUAL_UNLOCKING(Boolean), NONEXISTENT_RECORD(Boolean), READ REGARDLESS(Boolean), and WAIT_FOR_RECORD(Boolean), where Boolean is a BIT(1) expression. Either a constant or a run-time variable expression is acceptable. The record-locking option is either enabled or disabled for the duration of the READ statement. For example:

```
DECLARE BOOL BIT(1) ALIGNED;  
READ FILE(F) INTO(BUFF)  
    OPTIONS(NOLOCK(BOOL), LOCK_ON_READ(^BOOL));
```

- The correct status RMS\$_BUSY is now returned when you attempt to start a new I/O operation while another one is under way. Previously, the RMS status from the earlier I/O operation was returned rather than RMS\$_BUSY.
- When a file is implicitly opened in response to an I/O request, the RTL now uses the declared file attributes rather than the default file attributes. This helps avoid conflicts between declared and default attributes. For example, the following READ statement now implicitly opens the file REL1 with the UPDATE attribute rather than the default attribute INPUT.

```
DCL REL1 FILE KEYED DIRECT UPDATE;  
READ FILE(REL1) INTO(INREC) KEY(PART0);
```

- The ENCODE built-in function has been fixed to return the string "0" when the first parameter to ENCODE is 0. Previously the null string was returned.

4.38 XDELTA Notes

The release notes in this section describe procedures for bootstrapping the XDELTA debugger and requesting an interrupt for XDELTA for all computers available with VMS Version 5.4.

4.38.1 Invoking XDELTA

V5.4

To invoke the XDELTA debugger, perform the following steps:

- 1 Bootstrap the system using a console command or a command procedure that includes XDELTA.

Programmer Release Notes

4.38 XDELTA Notes

- 2 An initial XDELTA breakpoint is taken to allow setting of additional breakpoints or examining and changing of locations in memory. XDELTA displays the following breakpoint message:

```
1 BRK at 8000EB63
  8000EB63/NOP
```

- 3 Proceed from the initial breakpoint using the following command:

```
;P Return
```

The procedure for bootstrapping the system with XDELTA differs depending on your computer. Each procedure uses commands that include XDELTA in memory and cause the execution of a breakpoint in VMS initialization routines. Execution of the breakpoint instruction transfers control of the program to a fault handler that is located in XDELTA. The following sections describe the procedures for bootstrapping the computers, requesting an interrupt, and setting breakpoints in program code.

Some bootstrap procedures require use of the /R5 qualifier with the BOOT command. The /R5 qualifier enters a value for a flag that controls the way XDELTA is loaded. The flag is a 32-bit hexadecimal integer that is loaded into R5 as input to VMB.EXE, the primary bootstrap program. Table 4-4 contains a description of valid values for the flag.

Table 4-4 Valid XDELTA Values for the BOOT Command Qualifier

Value	Description
0	Normal, nonstop bootstrap (default)
1	Stop in SYSBOOT (equivalent to @DxyGEN on VAX-11/780 computers)
2	Include XDELTA with the system but do not take the initial breakpoint
6	Include XDELTA with the system and take the initial breakpoint
7	Include XDELTA with the system, stop in SYSBOOT, and take the initial breakpoint at system initialization (equivalent to @DxyXDT on VAX-11/750 computers)

Note: When you deposit a BOOT command qualifier value in R5, make sure that any other values you would normally deposit are included. For example, if you were depositing the number of the system root directory from which you were booting, as well as an XDELTA value, R5 would contain both values. If the system root directory value were 40000000, and the XDELTA value were 00000005, the final R5 value would be 40000005.

4.38.1.1 Bootstrapping a VAX-11/730 or VAX-11/750 Computer Using the VMS Console TU58

V5.4

In addition to the normal system bootstrap command files, the VMS console TU58 for a VAX-11/730 computer contains the following command files that bootstrap the system with XDELTA:

- DQAXDT
- DQ0XDT
- DL0XDT

Programmer Release Notes

4.38 XDELTA Notes

- DUAXDT
- DU0XDT

To bootstrap a VAX-11/730 computer with XDELTA, follow the procedures in the upgrade and installation supplement for your VAX computer, but specify one of the command files in the list given here.

For example, to bootstrap the VAX-11/730 computer from DQA1, enter the following commands at the console prompt:

```
>>> D/G/L 3 1
>>> @DQAXDT
```

The first command (D) deposits the unit number (1) in R3. The second command (@DQAXDT) invokes the DQAXDT command procedure.

If the boot device is DQA0, invoke the DQ0XDT command procedure, as follows. You do not have to specify the unit number.

```
>>> @DQ0XDT
```

Either of these procedures boots the processor and prompts you from SYSBOOT. When the SYSBOOT> prompt appears, enter any SYSBOOT command.

To continue the bootstrapping process, enter CONTINUE.

To bootstrap a VAX-11/750 computer with the console TU58, refer to the upgrade and installation supplement for the VAX-11/750 computer. The console TU58 contains the command files DUAXDT, DMAAXDT, and DBAXDT, which contain the command procedures that boot the system from DU, DM, and DB devices, respectively.

4.38.1.2 Bootstrapping XDELTA on a MicroVAX 2000, VAXstation 2000, MicroVAX 3400-Series, VAXstation 3520, VAXstation 3540, MicroVAX/VAXstation 3600-Series, MicroVAX 3900, VAX 4000 Model 300, MicroVAX II, or VAX-11/750 Computer

V5.4

To bootstrap VMS with XDELTA on all of these computers, enter the following command to specify the boot device:

```
>>> B/n devname
```

The B command is the console's BOOT command.

The *devname* parameter is the name of the device from which to bootstrap the system. Specify the device name using the format *ddcu*. (See the *Guide to Maintaining a VMS System* for a complete description of the format of device names.) You must specify identifiers for both the controller and the unit identifiers; there are no defaults.

The /n qualifier loads the value *n* into R5. The contents of R5 are passed as input to VMB.EXE. The value of *n* must be one of the 32-bit hexadecimal numbers described in Table 4-4.

For example, the following command bootstraps the VMS operating system on a VAX-11/750 computer from DUA0 with XDELTA included, stops at XDELTA's initial breakpoint, and stops in SYSBOOT to allow setting of system parameters:

```
>>> B/7 DUA0
```

The /7 qualifier includes XDELTA in the system and stops the booting process in SYSBOOT, which issues a prompt. It also stops at the breakpoint in the system initialization routine.

You can enter SYSBOOT commands when you see the SYSBOOT> prompt.

To continue the bootstrapping operation, enter CONTINUE.

See the upgrade and installation supplement for your computer for more information on the B command.

V5.4

4.38.1.3 Bootstrapping XDELTA on a VAX-11/780 or a VAX-11/785 Computer

In addition to the normal system bootstrap command files, the VMS console RX01 for a VAX-11/780 or VAX-11/785 computer contains the following command files that bootstrap the system with XDELTA:

- DUAXDT.CMD
- DMAXDT.CMD
- DBAXDT.CMD

To bootstrap the system with XDELTA, follow the procedures in the upgrade and installation supplement for your VAX computer, with the following exceptions:

- In R3, deposit the unit number of the drive that holds the system disk.
- Specify one of the command files given here.

For example, if the unit number of the drive that holds the system disk is zero, enter the following command:

```
>>> DEPOSIT R3 0
```

Then specify the command file that corresponds to the drive that holds the system disk. For example, if the system disk is on an RA80 drive with a controller designation of A, enter the following command:

```
>>> @DUAXDT
```

The command procedure boots the processor and prompts you from SYSBOOT. When the SYSBOOT> prompt appears, enter any SYSBOOT command.

To continue the bootstrapping operation, enter CONTINUE.

V5.4

4.38.1.4 Bootstrapping XDELTA on a VAX 6000-Series Computer

To boot a VAX 6000-series computer with XDELTA, use the following procedure:

- 1 If you are booting a VAX 6000-500 computer, skip this step.

If you have a CIBCA-A adapter and are booting over the CI, insert the console tape cartridge in the console drive.

- 2 Type Ctrl/P to put the system in console mode.

- 3 Enter the BOOT command in the following format:

```
>>> BOOT /R5:a /XMI:b /BI:c [/R3:d] [/NODE:e] DUu
```

Programmer Release Notes

4.38 XDELTA Notes

Table 4-5 lists the qualifiers to the BOOT command.

Table 4-5 XDELTA BOOT Command Qualifiers for VAX 6000-Series Computers

Qualifier	Function
/R5:a	Deposits a value (a) into R5. For valid values for the flag, refer to Table 4-4.
/XMI:b	Specifies the XMI node number (b) of the node being accessed. Defaults to the lowest numbered I/O device.
/BI:c	Specifies the BI node number (c) of the node being accessed and must be used with the /XMI qualifier. Defaults to zero.
/R3:d	Not required unless you are using volume shadowing. For more information on /R3:d, refer to the <i>VAX Volume Shadowing Manual</i> .
/NODE:e	Specifies the HSC node number of the node being accessed. The HSC node number is hexadecimal. You can specify a maximum of two HSC node numbers (if two HSCs are available). Refer to the upgrade and installation supplement for the VAX 6000-series computer.
u	Specifies the unit number of the drive that holds the system disk.

For example, use the following command to boot a VAX 6000 computer from the boot disk at VAXBI node 4, through the DWMBDA adapter at XMI node E, load XDELTA, stop in SYSBOOT, and take the initial breakpoint:

```
>>> BOOT /R5:7 /XMI:E /BI:4 DU1
SYSBOOT>
```

V5.4

4.38.1.5 Bootstrapping XDELTA on a VAX 8200, 8250, 8300, or 8350 Computer

To bootstrap a VAX 8200, 8250, 8300, or 8350 computer with XDELTA, use the B command (the console BOOT command) as follows:

```
>>> B/R5:f devname
```

The boot command qualifier (/R5:f) enters a value for a flag that controls how to load XDELTA. The flag is a 32-bit hexadecimal integer that is loaded into R5 as input to VMB.EXE, the primary bootstrap program. Refer to Table 4-4 for a description of the valid values for this flag. To use this qualifier, you must first modify the boot command procedure to remove (or comment out) the DEPOSIT R5 command.

The boot command procedure is specified by *devname* in the BOOT command. The *devname* format to use is *ddnu*, where *n* is the number of the VAXBI node to which the boot device unit is attached. If you do not specify a value for *devname*, the default boot device is used.

If in R5 you specified the flag to load SYSBOOT, the SYSBOOT> prompt appears. Enter any SYSBOOT command.

For example, use the following commands to boot a VAX 8200 computer from the boot disk at VAXBI node 4, load XDELTA, stop in SYSBOOT, and take the initial breakpoint (that is, R5 contains 7):

```
>>> B/R5:7 DU40
SYSBOOT> CONTINUE
```

V5.4

4.38.1.6 Bootstrapping XDELTA on a VAX 8530, 8550, 8810 (8700), 8820, 8820-N (8800), 8830, or 8840 Computer

To boot a VAX 8530, 8550, 8810 (8700), 8820, 8820-N (8800), 8830, or 8840 computer with XDELTA, use the BOOT command in the following format:

```
B dddn /R5:f
```

Substitute BCI, BDA, or UDA for *ddd*. Substitute the unit number of the drive that holds the system disk for *n*. Refer to Table 4-4 for a description of the valid values for this flag.

For example, if you have a BCI-controlled system disk with a unit number of 2, use the following command to load XDELTA and take the initial breakpoint:

```
>>> B BCI2 /R5:6
```

This command boots the system with BCIBOO.COM, deposits 2 in R3, and deposits 6 in R5.

You can also boot with XDELTA by editing the appropriate *ddd*GEN.COM procedure so that the unit number of the drive is deposited in R3. Then you can enter the BOOT command in the following format:

```
>>> @dddGEN
```

Substitute BCI, BDA, or UDA for *ddd*. For example, if the system disk is on a BCI-controlled drive, edit BCIGEN.COM so that the unit number of the drive is deposited in R3. At the console-mode prompt, enter the following command:

```
>>> @BCIGEN
```

V5.4

4.38.1.7 Bootstrapping XDELTA on a VAX 8600 or 8650 Computer

There are two ways to boot a VAX 8600 or VAX 8650 with XDELTA, depending on whether the console RL02 includes a boot command file in the *dd*0XDT format, where *dd* is the device code of the system disk.

If DU0XDT is present, follow the standard boot procedure except in the following two steps:

- When you specify the bootstrap device, enter the following command:

```
>>> DEPOSIT R3 u
```

This command deposits the unit number of the drive that holds the system disk, *u*, from which to boot.

- Then enter the following command to invoke DU0XDT:

```
>>> @DU0XDT
```

The command procedure boots the processor and prompts you from SYSBOOT. When the SYSBOOT> prompt appears, enter any SYSBOOT command.

To continue the bootstrapping operation, enter CONTINUE.

Programmer Release Notes

4.38 XDELTA Notes

If the console media does not have the DU0XDT file, perform a normal bootstrap procedure using an available *dduGEN.COM*, *dduBOO.COM*, or *DEFBOO.COM* procedure, including the following steps:

- 1 Include the */NOSTART* qualifier to the *BOOT* command to cause the processor to pause and prompt for console commands prior to starting the VMB initialization routines.
- 2 Select a value from Table 4-4 for the boot flag to control the loading of XDELTA.
- 3 Examine the value of the boot flag in R5. If it is not the value you want, deposit the correct value.

For example, use the following procedure to boot a VAX 8600 computer to include XDELTA, stop in *SYSBOOT*, take the initial breakpoint (flag value of 7), and continue the boot procedure:

```
>>> BOOT/NOSTART
>>> EXAMINE R5 40000000
>>> DEPOSIT R5 2 40000007
>>> CONTINUE
```

4.38.1.8 Bootstrapping XDELTA on a VAX 9000 Computer

V5.4

To boot a VAX 9000 computer with XDELTA, use the *BOOT* command in the following format:

```
BOOT [/NOSTART] [/R5:boot_flags] [R3:shadow] [/BI:vaxbi_info] [/XMI:xmi_info] [/NODE:hsc_info] xxxunit_number
```

The parameter *xxxunit_number* refers to the abbreviation of the boot command procedure you are using (*xxx*) and the unit number of the drive. For example, if you are booting from an HSC system disk with unit number 30, use *CI* (the abbreviation for *CIBOO.COM*) and the unit number 30. Use the parameter *CI30*.

Table 4-6 lists the qualifiers to the *BOOT* command. Please note that values you specify for *BOOT* command qualifiers can be overridden by *DEPOSIT* commands in a boot command procedure. For example, you could set up *DEFBOO.COM* to boot from the *[SYS0]* directory. If you enter the following command, you expect the system to boot from the *[SYSC]* directory; however, a *DEPOSIT R5* command in *DEFBOO* overrides the value you specify on the command line, and the system boots from the *[SYS0]* directory:

```
>>> B /R5:C0000000
```

To avoid this override condition, boot with the */NOSTART* qualifier as follows:

```
>>> B /NOSTART
>>> D R5 C0000000
>>> CONTINUE
```


Programmer Release Notes

4.38 XDELTA Notes

Table 4–6 BOOT Command Qualifiers for VAX 9000 Computers

Qualifier	Function
/NOSTART	Stops the boot operation after the boot command procedure executes. Lets you deposit values in registers before transferring control to the primary boot program with the START command.
/R5:boot_flags	Deposits a value (in hexadecimal) into R5. The value affects the execution of VMB9AQ.EXE. You can perform a conversational boot or boot from a different system root. Use to boot XDELTA. See Table 4–4 for a description of the valid values for the flag.
/R3:shadow	Specifies shadowing information. Only required if you are using Volume Shadowing phase I. If you are using Volume Shadowing phase II, you need to set several SYSGEN parameters to use volume shadowing. For more information on /R3:shadow, see the <i>VAX Volume Shadowing Manual</i> and the <i>VMS Volume Shadowing Manual</i> .
/BI:vaxbi_info	Specifies the BI node number (in hexadecimal) of the node being accessed. Defaults to zero. If you do not access an XBI or XBI-Plus adapter, you do not need to specify /BI:vaxbi_info.
/XMI:xmi_info	Specifies the XMI number and the XMI node number (both in hexadecimal) of the node being accessed. Defaults to zero. The hexadecimal number that you specify must be in the format xy, where x is the XMI number and y is the XMI node number.
/NODE:hsc_info	Specifies the CI node number (in hexadecimal) of the HSC being accessed. You can specify a maximum of two CI node numbers (if two HSCs are available). If you deposit two CI node numbers, put the greater number in hexadecimal digits 3 and 2. Put the smaller number in hexadecimal digits 1 and 0. For more information on booting when two HSCs are available, refer to the upgrade and installation supplement for the VAX 9000.

For example, to boot from the default system disk with XDELTA and stop in SYSBOOT for input, enter the following command:

```
>>> B/R5:7
```

4.38.1.9 Bootstrapping XDELTA on a VAXft 3000 Computer

V5.4

To boot a VAXft 3000 computer with XDELTA, use the BOOT command in the following format:

```
$  Break  
>>> BOOT/R5:7
```

This format of the BOOT command invokes XDELTA and causes SYSBOOT to prompt for input. If Break is not enabled, press F5.

Programmer Release Notes

4.38 XDELTA Notes

4.38.1.10 Bootstrapping XDELTA on a VAXstation 3100 or MicroVAX 3100-Series Computer

V5.4

To boot XDELTA on a VAXstation 3100 or MicroVAX 3100-series computer, use the following command:

```
>>> B/f dev-name
```

For *dev-name*, enter the device name of the system disk. For *f*, enter a value from Table 4-4.

For example, to boot XDELTA and stop at the breakpoint from a drive with a device name of DKA400, enter the following command:

```
>>> B/6 DKA400
```

4.38.2 Requesting an Interrupt

V5.4

If you set the boot control flag in R5 to 7, as described in Section 4.38.1, XDELTA stops at an initial breakpoint during the system bootstrap process. You can then set other breakpoints or examine locations in memory.

Your program can also call the routine INI\$BRK, which in turn executes the first XDELTA breakpoint. Note that INI\$BRK is defined as XDELTA's breakpoint 1. Never clear breakpoint 1 from any code being debugged in XDELTA.

Once XDELTA is loaded into memory, you can also invoke it at any time from the console by requesting a software interrupt. For example, you might need to use a software interrupt to enter XDELTA if your program is in an infinite loop or if no INI\$BRK call had been made.

To request a software interrupt for all computers, deposit the value E₁₆ into IPR 14₁₆.

For a VAX 8530, 8550, 8600, 8650, 8810 (8700), 8820, 8820-N (8800), 8830, 8840, 9000, VAX-11/780, or VAX-11/785 computer, enter the following commands at the console terminal to request the interrupt:

```
$ Ctrl/P  
>>> HALT  
>>> D/I 14 E  
>>> C
```

On a VAX 6000-series, 8200, 8250, 8300, 8350, VAX-11/730, or VAX-11/750 computer, enter the following commands at the console terminal:

```
$ Ctrl/P  
>>> D/I 14 E  
>>> C
```

For a VAXstation 3520 or 3540 computer, perform the following steps:

- Press and release the HALT button on the CPU control panel. When you release the HALT button, make sure it is popped out, or the system remains halted. Alternatively, you can press Break (if enabled) on the console terminal.

Programmer Release Notes

4.38 XDELTA Notes

- Enter the following commands at the console terminal:

```
>>> D/I 14 E
>>> C/ALL
```

For a VAXft 3000 computer, enter the following commands at the console terminal to request the interrupt:

```
$ Break
```

or

```
$ F5
>>> HALT
>>> D/I 14 E
>>> CONT
>>> PIO
```

For the VAXstation 2000, MicroVAX 2000, MicroVAX 3400-series, MicroVAX/VAXstation 3600-series, MicroVAX 3900-series, VAX 4000 Model 300, or MicroVAX II computer, perform the following steps:

- Press and release the HALT button on the CPU control panel. When you release the HALT button, make sure it is in the out position, or the system remains halted. Alternatively, you can press Break (if enabled) on the console terminal.
- Enter the following commands at the console terminal:

```
>>> D/I 14 E
>>> C
```


5

Documentation Release Notes

This chapter contains additions and corrections to the VMS documentation set.

5.1 *Guide to DECnet-VAX Networking*

V5.2 On page 2-26 near the end of Example 2-1 of the *Guide to DECnet-VAX Networking*, the field **ast_ret_status** replaces the field **return_status** in the **ast_routine**, on the lines beginning with **switch (ast_param.type)** and ending with **exit (return_status)**. The replacement code is as follows:

```
{
  switch ( ast_param.type ) {
    case NET_RD   : ast_ret_status = insque_buffer ( LIVE_QUE,
                                                    lct[ast_param.ndx].cur_buff );   break;
    case NET_WRT  : ast_ret_status = insque_buffer ( LIVE_QUE,
                                                    lct[ast_param.ndx].cur_buff );   break;
    case NET_CMD  : ast_ret_status = que_and_reissue ();                               break;
    default      : ast_ret_status = SS$_BADPARAM;                                   break;
  }
  if (ast_ret_status & STS$M_SUCCESS) {
    ast_ret_status = sys$wake ( 0, 0 );
    if ( ast_ret_status != SS$_NORMAL )
      exit ( ast_ret_status );
  }
  else
    exit ( ast_ret_status );
}
```

V5.4 On page 3-27, replace the example in the last bullet of step 8b with the following corrected example:

```
$ RUN SYS$SYSTEM:NCP
NCP> SET LINE TT-0-0 RECEIVE BUFFERS 4 LINE SPEED 2400 STATE ON
NCP> EXIT
```

5.2 *Guide to Setting Up a VMS System*

V5.4 Make the following corrections to the *Guide to Setting Up a VMS System*:

- Make the following format changes to page 6-9:
 - In step 1, the first pass of AUTOGEN, the end phase to the **@SYS\$UPDATE:AUTOGEN** command is listed as **GENPARAMS**. Although the **GENPARAMS** phase is not incorrect, Digital recommends that you specify **TESTFILES** as the end phase.

The recommended format is:

```
$ @SYS$UPDATE:AUTOGEN SAVPARAMS TESTFILES
```

Documentation Release Notes

5.2 Guide to Setting Up a VMS System

- In step 2, the second pass of AUTOGEN, the start phase to the @SYS\$UPDATE:AUTOGEN command is listed as SETPARAMS. Although the SETPARAMS phase is not incorrect, Digital recommends you specify GETDATA as the start phase.

The recommended format is:

```
$ @SYS$UPDATE:AUTOGEN GETDATA REBOOT
```

- Table 6-2 incorrectly lists the names of the SYSGEN parameters for the PQL default and minimum working set sizes, extents, and quotas.

In the following table, the parameters are shown as they are currently named in Table 6-2, followed by their correct names:

Name Currently Listed	Correct Name
PQL_DWDEFAULT	PQL_DWSDEFAULT
PQL_DWEXTENT	PQL_DWSEXTENT
PQL_DWQUOTA	PQL_DWSQUOTA
PQL_MWDEFAULT	PQL_MWSDEFAULT
PQL_MWEXTENT	PQL_MWSEXTENT
PQL_MWQUOTA	PQL_MWSQUOTA

- The section entitled Building and Copying a VMS System Disk contains an incorrect description of VMSKITBLD.COM. Replace this incorrect description with Appendix C of this manual, which contains the correct VMSKITBLD.COM description.
- Add the following information to Section 3.2.3:

If the workstation is running DECwindows, the system manager should enter the following commands during the conversational bootstrap:

```
SYSBOOT> SET WINDOW_SYSTEM 0  
SYSBOOT> SET UAFALTERNATE 1  
SYSBOOT> CONTINUE
```

Follow the directions as documented in steps 1 through 5. After logging in to the console terminal, enter the following command lines:

```
$ DEFINE/SYSTEM/EXECUTIVE_MODE SYSUAF SYS$SYSTEM:SYSUAF.DAT  
$ SET DEFAULT SYS$SYSTEM  
$ RUN AUTHORIZE
```

Correct the problem that caused you to be locked out of the system. That is, make the necessary changes to the UAF. Using SYSGEN as follows, set the WINDOW_SYSTEM parameter and clear the UAFALTERNATE parameter:

```
$ RUN SYS$SYSTEM:SYSGEN  
SYSGEN> SET WINDOW_SYSTEM 1  
SYSGEN> SET UAFALTERNATE 0  
SYSGEN> WRITE CURRENT  
SYSGEN> EXIT
```

Documentation Release Notes

5.2 Guide to Setting Up a VMS System

Shut down and reboot the system.

With VMS Version 5.4, the sequence of operations for the command procedure STARTUP.COM has changed. For the new sequence of operations, see Section 3.41.

5.3 **Guide to VMS File Applications**

V5.4 Section 3.2.1 in the *Guide to VMS File Applications* contains insufficient information. Replace Section 3.2.1 with Appendix D of this manual, which contains updated information.

5.4 **Guide to VMS Programming Resources**

V5.4 In Sections 2.1.3, 2.1.3.3, and 4.6.1 of the *Guide to VMS Programming Resources*, replace all references to the PPL\$CREATE_PROCESS routine with its correct name: PPL\$SPAWN. Note also that the syntax given for the PPL\$CREATE_PROCESS routine is not the correct syntax for the PPL\$SPAWN routine. For the correct syntax of the PPL\$SPAWN routine, see the *VMS RTL Parallel Processing (PPL\$) Manual*.

In Section 4.6.1, replace the reference to the PPL\$DELETE_PROCESS routine with its correct name: PPL\$STOP.

In Section 4.6.3, replace the reference to the PPL\$RETURN_SEMAPHORE_VALUES routine with its correct name: PPL\$READ_SEMAPHORE.

On page 3-30, add the following new section after Section 3.4.4.

Storing Resources Names in a Network

Use the DIGITAL Distributed Name Service (DECdns) system service, \$DNS, to store the names of resources, such as files, disks, nodes, queues, and mailboxes, in your network. For more information about DECdns, see the *VMS Version 5.4 New Features Manual*.

On page 4-19, insert the following new section before Section 4.8.

Ensuring Atomicity in Distributed Transactions

Use the DECdtm services to ensure the atomicity of transactions in a distribute application. For more information about DECdtm services, see the *VMS Version 5.4 New Features Manual*.

5.5 **Introduction to VMS System Services**

V5.2 On page 3-33 of the *Introduction to VMS System Services*, the decision box ACL ON THIS OBJECT in the flowchart of \$CHKPRO operation should not be connected to the box ACCESS DENIED. If the answer is YES, the arrow should point to the CHECK ACCESSOR FOR PRIVILEGES box. The box ACCESS DENIED at the right of the page should be deleted.

Documentation Release Notes

5.5 Introduction to VMS System Services

On page 11-14, there is an error in the \$CRMPSC example. The UPI option must be specified if the file is to be write-shared.

Incorrect example:

```
SECFAB: $FAB FNM=<SECTION.TST>, -  
        FOP=UFO, -  
        FAC=PUT, -  
        SHR=<GET,PUT>  
        .  
        .  
        .
```

Corrected example:

```
SECFAB: $FAB FNM=<SECTION.TST>, -  
        FOP=UFO, -  
        FAC=PUT, -  
        SHR=<GET,PUT,UPI>  
        .  
        .  
        .
```

V5.4

The documentation that explains the format of the system service call can be misleading to programmers who use high-level languages. On pages 1-4 and 1-5, the documentation states that arguments and argument placeholders within brackets ([]) are optional and can be omitted.

The following example is used to show the syntax for the system service call.

```
ENTRY-POINT-NAME    arg1 ,arg2 ,[arg3] ,nullarg [,arg4] [,arg5]
```

The explanation of this syntax is as follows:

Square brackets surround optional arguments; **arg3**, **arg4**, and **arg5** are optional arguments because they are surrounded by brackets. Note that the commas can also be optional.

Between arguments, the comma always follows the space. If the argument is optional, the comma may appear inside or outside the brackets, depending on the position of the argument in the list and on whether surrounding arguments are optional or required.

For example, **arg3** in the sample format is an optional argument, but because other required arguments follow **arg3** in the list, the comma itself is not optional (because it marks the place of **arg3**). Therefore, the comma is outside the brackets.

The arguments **arg4** and **arg5** are optional. Because no required arguments follow **arg4** and **arg5** in the list, the commas in front of **arg4** and **arg5** are themselves optional; that is, you would not specify the commas in the call if you did not specify **arg4** and **arg5**. Therefore, the commas in front of **arg4** and **arg5** are inside the brackets.

Documentation Release Notes

5.5 Introduction to VMS System Services

Optional Arguments Cannot Always Be Omitted

Languages like MACRO and Bliss allow you to omit trailing arguments and their placeholders. Optional arguments are handled differently by other languages. If you are coding in VAX/BASIC, VAX/C, VAX/COBOL, VAX/FORTRAN, or VAX/Pascal, you cannot omit trailing arguments; the compilers for these languages expect all arguments to be present.

For example, if you code a system service call from a C program and replace the **arg4** and **arg5** arguments with commas, the program will not compile; if you omit the arguments and the commas, the program compiles but you get a run-time error. The C language requires a zero in place of omitted arguments.

The following table describes the way some languages treat optional arguments.

Language	How to Omit Optional Arguments
BASIC	Replace the argument with a comma
Bliss-32	Omit the keyword or replace the argument with zero
C	Replace the argument with zero
COBOL	Replace the argument with the keyword OMITTED
FORTRAN	Replace the argument with a comma
MACRO	Omit the keyword or replace the argument with zero
Pascal	Replace the argument with a comma
PL/1	Replace the argument with a comma

For a more detailed explanation of how to handle placeholders for optional arguments, see the applicable language reference manual.

V5.4

Section 1.1.3.4 discusses passing values by descriptor and includes a long list of possible descriptor classes. In fact, the only descriptor class that system services accepts is class S, also known as fixed length. (The services actually assume that a class S descriptor is passed.) If a non-S descriptor is passed, incorrect execution may result.

The list of passing mechanisms has been revised to include only fixed-length descriptors, as follows:

- By value
- By reference
- By reference, array reference
- By descriptor, fixed-length

V5.4

On page 2-14, the text under the System Service Failure Exception Mode heading has been moved to the *VMS Obsolete Features Manual*.

Documentation Release Notes

5.5 Introduction to VMS System Services

V5.4 On page 6-20, the sample program did not link. The corrected example is as follows:

```

PROGRAM CALC
! Status variable and system routines
    INCLUDE '($LNMDEF)'
    INCLUDE '($SYSSRVNAM)'
    INTEGER*4 STATUS

    INTEGER*2 NAME_LEN,
2      NAME_CODE
    INTEGER*4 NAME_ADDR,
2      RET_ADDR /0/,
2      END_LIST /0/

    COMMON /LIST/ NAME_LEN,
2      NAME_CODE,
2      NAME_ADDR,
2      RET_ADDR,
2      END_LIST

    CHARACTER*3 REPETITIONS_STR
    INTEGER REPETITIONS

    EXTERNAL CLI$M NOLOGNAM,
2      CLI$M NOCLISYM,
2      CLI$M NOKEYPAD,
2      CLI$M NOWAIT

    NAME_LEN = 3
    NAME_CODE = (LNM$ STRING)
    NAME_ADDR = %LOC(REPETITIONS_STR)
    STATUS = SYS$CRELNM (,'LNM$JOB','REP_NUMBER',,NAME_LEN)
    IF (.NOT. STATUS) CALL LIB$SIGNAL (%VAL(STATUS))

    MASK = %LOC (CLI$M NOLOGNAM) .OR.
2      %LOC (CLI$M NOCLISYM) .OR.
2      %LOC (CLI$M NOKEYPAD) .OR.
2      %LOC (CLI$M NOWAIT)
    STATUS = LIB$GET_EF (FLAG)
    IF (.NOT. STATUS) CALL LIB$SIGNAL (%VAL(STATUS))
    STATUS = LIB$SPAWN ('RUN REPEAT',,,MASK,,,,FLAG)
    IF (.NOT. STATUS) CALL LIB$SIGNAL (%VAL(STATUS))

    END

PROGRAM REPEAT
    INTEGER STATUS,
2      SYS$TRNLNM, SYS$DELLNM
    INTEGER*4 REITERATE,
2      REPEAT_STR_LEN
    CHARACTER*3 REPEAT_STR
! Item list for SYS$TRNLNM
    INTEGER*2 NAME_LEN,
2      NAME_CODE
    INTEGER*4 NAME_ADR,
2      RET_ADR,
2      END_LIST /0/
    COMMON /LIST/ NAME_LEN,
2      NAME_CODE,
2      NAME_ADDR,
2      RET_ADDR,
2      END_LIST

```

Documentation Release Notes

5.5 Introduction to VMS System Services

```

NAME_LEN = 3
NAME_CODE = (LNM$ STRING)
NAME_ADDR = %LOC(REPEAT_STR)
RET_ADDR = %LOC(REPEAT_STR_LEN)
STATUS = SYS$STRNLNM (,
2         'LNM$JOB',      ! Logical name table
2         'REP_NUMBER',  ! Logical name
2         NAME_LEN)      ! List requesting equivalence string
IF (.NOT. STATUS) CALL LIB$SIGNAL (%VAL(STATUS))

READ (UNIT = REPEAT_STR,
2     FMT = '(I3)') REITERATE

DO I = 1, REITERATE
END DO

STATUS = SYS$DELLNM ('LNM$JOB',      ! Logical name table
2                  'REP_NUMBER',) ! Logical name
IF (.NOT. STATUS) CALL LIB$SIGNAL (%VAL(STATUS))

END

```

V5.4

On page 9-11, the timer service example has the following errors:

- The program assumes that PRC\$V_HIBER is available as a global symbol.
- The program does not declare the system services as INTEGER.
- The program does not declare STATUS as INTEGER.

It has been corrected as follows:

```

INCLUDE '($SYSSRVNAM)' ! Declare system services
INCLUDE '($PRCDEF)'    ! Declare process options

! SYS$CREPRC options and values
INTEGER OPTIONS,
2 STATUS

! ID of created subprocess
INTEGER CR_ID

! Binary times
INTEGER TIME(2),
2     INTERVAL(2)

! Set the PRC$V_HIBER bit in the OPTIONS mask and
! create the process.
STATUS = SYS$CREPRC (CR_ID,      ! PID of created process
2                   'CHECK',    ! image
2                   'SLEEP',    ! Process name
2                   %VAL(4),    ! Priority
2                   %VAL(PRC$M_HIBER)) ! Hibernate
IF (.NOT. STATUS) CALL LIB$SIGNAL (%VAL(STATUS))

! Translate 6:00 a.m. (absolute time) to binary
STATUS = SYS$BINTIM ('23-- 06:00:00.00', ! 6:00 a.m.
2                   TIME)
IF (.NOT. STATUS) CALL LIB$SIGNAL (%VAL(STATUS))

! Translate 10 minutes (delta time) to binary
STATUS = SYS$BINTIM ('0 :10:00.00', ! 10 minutes
2                   INTERVAL)
IF (.NOT. STATUS) CALL LIB$SIGNAL (%VAL(STATUS))

```

Documentation Release Notes

5.5 Introduction to VMS System Services

```
! Schedule wakeup calls
STATUS = SYS$SCHEDWK (CR_ID,      ! ID of created process
2
2          TIME,      ! Initial wakeup time
2          INTERVAL) ! Repeat wakeup time
IF (.NOT. STATUS) CALL LIB$SIGNAL (%VAL(STATUS))
END
```

5.6 VAX MACRO and Instruction Set Reference Manual

V5.4 On pages 9-191 and 9-192 of the *VAX MACRO and Instruction Set Reference Manual*, the MTPR and MFPR instructions are documented incorrectly as setting the condition codes in a predictable way. In fact, MTPR and MFPR leave the condition codes in the UNPREDICTABLE state.

5.7 VMS Authorize Utility Manual

V5.4 In Table AUTH-2 in the *VMS Authorize Utility Manual*, the /EXPIRATION qualifier and function should read as follows:

/[NO]EXPIRATION	Specifies the expiration date and time of the account. Default is 180 days for nonprivileged users. Use the /NOEXPIRATION qualifier to reset the expiration time for expired accounts. With /NOEXPIRATION, an account does not expire.
-----------------	--

5.8 VMS DECnet Test Sender/DECnet Test Receiver Utility Manual

V5.0 On page DTS-8, the following qualifiers to the DATA command are listed. These qualifiers are no longer supported:

- FLOW=flow-control
- NOFLOW
- RQUEUE=number
- SQUEUE=number
- NAK=number
- NONAK
- BACK=number
- NOBACK

On page DTS-9, the example contains the unsupported qualifier /FLOW=MESSAGE. The correct command line in the example should be as follows:

```
_TEST: DATA/PRINT/TYPE=SEQ/SIZE=128/SECONDS=10
```

On page DTS-13, the following qualifiers to the INTERRUPT command are listed. These qualifiers are no longer supported:

- RQUEUE=number

5.8 *VMS DECnet Test Sender/DECnet Test Receiver Utility Manual*

- SQUEUE=number

5.9 ***VMS DECwindows Desktop Applications Guide***

V5.4

This section contains corrections to the Calendar chapter of the *VMS DECwindows Desktop Applications Guide*.

On page 4-10, under the section "Creating Overlapping Entries," the manual states the following:

With Show Text enabled, even if you have two entries starting at the same time, Calendar leaves a gap to display at least some of the text associated with the hidden entry.

Currently, Calendar does not display any text in hidden entries, when the entries start at the same time. To view a hidden entry, click on its icon and the entry moves to the top of the stack. When the entries partially overlap, Calendar tries to stack them so that the first line of text for each entry is visible.

On page 4-12, under the section "Sending and Receiving Calendar Entries Through Mail," the manual incorrectly describes how to select and paste a mail message into Calendar. Replace the third paragraph of the section with the following:

To receive an entry through electronic mail, first open the day view and use MB1 to select the text from an existing mail message. Then use MB1 to give Calendar input focus. Finally, use MB3 to paste the mail message into Calendar.

On pages 4-14 and 4-15, under the sections "Creating a New Calendar" and "Opening an Existing Calendar," step 2 is incorrect. In both cases, the paragraphs beginning with "A dialog box..." and ending with "at the same time." should be deleted, as no dialog box appears. The remainder of the procedures are correct as described.

5.10 ***VMS DECwindows Guide to Application Programming***

V5.4

In the *VMS DECwindows Guide to Application Programming*, corrections have been made to memory usage patterns in both the Hello World! and the DECburger sample applications. The following sections describe these changes. The corrected versions of both the Hello World! and DECburger sample applications can be found in the DECW\$EXAMPLES directory.

Freeing Memory in the Hello World! and DECburger Sample Applications

In the Hello World! and DECburger sample applications, code has been added to free memory allocated by the compound string creation routine. If you do not explicitly free this memory, it accumulates and is unavailable for other uses. For example, after using a compound string to specify a widget label, you should free the compound string because the widget has made its own copy of it. Use the FREE intrinsic routine to free memory obtained by compound string creation routines.

Documentation Release Notes

5.10 VMS DECwindows Guide to Application Programming

Example 5-1 Freeing Compound Strings in the Hello World! Sample Application

```
.
.
.
/*****      Main Input Loop      *****/
    XtMainLoop();
/*****      Callback Routine      *****/
static void helloworld_button_activate( widget, tag, callback_data )
    Widget      widget;
    char        *tag;
    DwtAnyCallbackStruct *callback_data;
{
    Arg arglist[2];
    ❶ DwtCompString newlabel;
    static int call_count = 0;
    call_count += 1 ;
    switch ( call_count )
    {
        case 1:
            ❷ newlabel = DwtLatin1String("Goodbye\nWorld!");
            ❸ XtSetArg( arglist[0], DwtNlabel, newlabel);
            XtSetArg( arglist[1], DwtNx, 11 );
            XtSetValues( widget, arglist, 2 );
            ❹ XtFree( newlabel);
            break ;
        case 2:
            exit(1);
            break ;
    }
}
```

Example 5-1 is the new callback routine for the Hello World! sample application. Replace the existing Example 2-18 with Example 5-1.

- ❶ The variable *newlabel* holds the address of the compound string used as the new push button label.
- ❷ The new push button label, "Goodbye World!", is created using the compound string creation routine LATIN1 STRING.
- ❸ The *newlabel* variable is passed as an argument to the SET VALUES intrinsic routine to change the push button label. The original version did not use a temporary variable. Instead, the LATIN1 STRING routine was placed in line as an argument to the SET VALUES routine, as in the following example:

```
XtSetArg( arglist[0], DwtNlabel, DwtLatin1String("Goodbye\nWorld!");
```

This method did not provide any way to free the memory allocated by the compound string routine. When you use a temporary variable to hold the address of the memory allocated by the compound string routine, you can pass this address to the FREE intrinsic routine.

- ❹ The memory allocated by LATIN1 STRING is freed using the FREE intrinsic routine.

Documentation Release Notes

5.10 VMS DECwindows Guide to Application Programming

In addition, code has been added to the Ada and FORTRAN versions of the Hello World! sample application to free the memory allocated by the VMS SET ARG routine. (Note that this change has also been made to the Ada and FORTRAN versions of the DECBurger sample application.)

The intrinsic routines expect the argument name field in an argument list entry to be the address of a null-terminated string. In languages other than C, null-terminated strings are inconvenient to use; therefore, the Ada and FORTRAN versions of the Hello World! and DECBurger examples use the VMS SET ARG routine to convert character constants to null-terminated strings and place the address of the dynamically allocated null-terminated string in the argument list. As with compound string creation routines, the dynamic memory allocated by the VMS SET ARG routine for the null-terminated string must be freed explicitly when the string is no longer needed. Use the VMS FREE ARGUMENTS routine to free these dynamically allocated argument-name strings, as illustrated in Example 5-2.

Example 5-2 Freeing Memory Allocated by the VMS SET ARG Routine

```
.
.
.
① DWT.VMS_SET_ARG (
    ARG      => CSTRING,
    ARGLIST  => ARG_LIST,
    ARGNUMBER => 0,
    ARGNAME  => DWT.C_NLABEL);
.
.
.
② DWT.XT_SET_VALUES (
    WIDGET   => WIDGET,
    ARGLIST  => ARG_LIST,
    ARGCOUNT => ARG_LIST'LENGTH);
.
③ DWT.VMS_FREE_ARGUMENTS (
    ARGLIST  => ARG_LIST,
    ARGCOUNT => ARG_LIST'LENGTH);
.
.
.
```

-
- ① The VMS SET ARG routine converts the argument name, *DWT.C_NLABEL*, into a null-terminated string and puts it in the argument list.
 - ② The SET VALUES intrinsic routine accepts the argument list as one of its arguments and makes the change to the label of the push button.
 - ③ The VMS FREE ARGUMENTS routine frees the memory allocated by the VMS SET ARG routine.

Documentation Release Notes

5.10 VMS DECwindows Guide to Application Programming

Memory Errors Corrected in the DECburger Sample Application

A set of errors relating to the improper use of memory was corrected in the DECburger sample application in all three language versions. In several places in the application, the GET VALUES intrinsic routine was called to obtain a compound string value of a widget, such as a list box. After using the compound string value, the application called the FREE intrinsic routine to free the compound string. Because the value was not a copy of the widget's compound string value but rather the original value from the widget's data structures, this improper freeing of the string caused abnormal behavior in the application, such as an access violation occurring after you clicked on the Apply button several times.

These errors have been corrected. A copy is now made of any compound string values that are to be kept for later use, and compound strings that have not been allocated by the application are no longer freed.

5.11 **VMS DECwindows Programming Documentation Supplement**

V5.4

In Section 5.3, Using the DECTERM PORT Routine, of the *VMS DECwindows Programming Documentation Supplement*, the routine name is listed as DWT\$DECTERM_PORT. The correct routine name is DECW\$TERM_PORT.

This documentation change has been incorporated into the *VMS DECwindows Guide to Application Programming*, which supersedes the *VMS DECwindows Programming Supplement*.

5.12 **VMS DECwindows User Interface Language Reference Manual**

V5.4

This section describes an addition to the widget attribute descriptions in the *VMS DECwindows User Interface Language Reference Manual*.

A number of widget attributes are allowed in the Toolkit routines but cannot be handled by the UIL compiler. Table 5-1 lists the arguments and widgets that are supported by the Toolkit routines but are not supported by the UIL compiler. Because these widgets and arguments are not supported in UIL, they are not listed in Appendix B of the *VMS DECwindows User Interface Language Reference Manual*; the only way you can use them is to use the ARGUMENT function to create a user-defined argument.

5.12 VMS DECwindows User Interface Language Reference Manual

Table 5-1 Arguments Unsupported by the UIL Compiler

Argument	Widget
grab_key_syms	Attached dialog box, Caution box, Color mix, Command window, Dialog box, File selection, Help box, Label widget, List box, Main window, Menu bar, Message box, Option menu, Popup attached dialog box, Popup dialog box, Popup menu, Pulldown menu, Push-button widget, Radio box, Scale, Scroll bar, Scroll window, Selection, Separator, Toggle button, Window, Work-in-progress box
no_resize	Color mix
take_focus	Color mix
menu_extend_last_row	Option menu

In addition, there is one widget argument that the UIL compiler supports but the Toolkit does not: the menu_extend_last_row argument for the radio box widget.

5.13 VMS DECwindows Xlib Programming Volume

V5.4

In Table 8-5 (page 8-17) of the *VMS DECwindows Guide to Xlib Programming: VAX Binding*, the information for the last three atoms has been updated, including a new name for atom X\$C_XA_FONT_NAME.

Replace the existing text for atoms X\$C_XA_FONT_NAME, X\$C_XA_FAMILY_NAME, and X\$C_XA_FULL_NAME with the following updated text:

Atom	Data Type	Description of the Property
X\$C_XA_FONT	Atom	Font name. For example: <i>-Adobe-Helvetica-Bold-R-Normal-10-100-75-75-P-60-ISO8859-1</i>
X\$C_XA_FAMILY_NAME	Atom	Name of the font family. For example: <i>Helvetica</i>
X\$C_XA_FULL_NAME	Atom	Full name of the font. For example: <i>Helvetica Bold</i>

In Table 8-5 (page 8-17) of the *VMS DECwindows Guide to Xlib Programming: MIT C Binding*, the information for the last three atoms has been updated, including a new name for atom XA_FONT_NAME.

Replace the existing text for atoms XA_FONT_NAME, XA_FAMILY_NAME, and XA_FULL_NAME with the following updated text:

Atom	Data Type	Description of the Property
XA_FONT	Atom	Font name. For example: <i>-Adobe-Helvetica-Bold-R-Normal-10-100-75-75-P-60-ISO8859-1</i>
XA_FAMILY_NAME	Atom	Name of the font family. For example: <i>Helvetica</i>
XA_FULL_NAME	Atom	Full name of the font. For example: <i>Helvetica Bold</i>

Documentation Release Notes

5.14 VMS I/O User's Reference Manual: Part I

5.14 VMS I/O User's Reference Manual: Part I

V5.4 On page 1-17 of the *VMS I/O User's Reference Manual: Part I*, in the attribute list description, 0 is given as a legal value in the ATR\$W_SIZE field. Previously, the I/O subsystem terminated the ACP attribute list processing when it encountered 0 in the ATR\$W_SIZE field. Now, the I/O subsystem checks both the ATR\$W_TYPE and ATR\$W_SIZE fields before terminating the attribute list processing.

If you are developing an ACP, you should be aware that it is now possible for an ACP to receive an attribute descriptor with 0 length.

In Section 8.2.1.5, the last sentence in the next-to-last paragraph incorrectly states:

TTY_ALTALARM specifies the threshold at which a Ctrl/S is sent.

Replace the incorrect text with the following corrected text:

A Ctrl/S is sent if the type-ahead buffer contains less than TTY_ALTALARM bytes

5.15 VMS License Management Utility Manual

V5.4 Sections 11.1, 11.2, and 11.3 (pages 34 and 36) of the *VMS License Management Utility Manual* Version 5.2 describe a VMSLICENSE.COM parameter named *NOSHARE_NODE*. Although the description is correct, the parameter name is incorrect. The correct parameter name is *INCLUDE_NODE*.

5.16 VMS Linker Utility Manual

V5.0 The *VMS Linker Utility Manual* noted that a linker produces a debugger symbol table (DST) only if the /DEBUG qualifier is specified at link time. In fact, the linker produces a DST when either the /DEBUG or the /TRACEBACK qualifier is specified at link time.

V5.3 In Section 3.3 (Link Options), IDENTIFICATION=id-name subsection, of the *VMS Linker Utility Manual*, the maximum length of the image-id field is incorrectly stated as 39 characters. The correct maximum length of the image-id field is 15 characters.

5.17 VMS Monitor Utility Manual

V5.2 On page MON-42 of the *VMS Monitor Utility Manual*, the Network Control Program (NCP) example is no longer valid. Refer to the following example instead:

Documentation Release Notes

5.17 VMS Monitor Utility Manual

```
$ SET PROCESS/PRIVILEGE=SYSPRV
$ RUN SYS$SYSTEM:NCP
NCP> DEFINE OBJECT VPM NUMBER 51 -
- FILE SYS$SYSTEM:VPM.EXE -
- PROXY NONE -
- ACCOUNT account -
- USER user-id -
- PASSWORD password
NCP> SET OBJECT VPM NUMBER 51 -
- FILE SYS$SYSTEM:VPM.EXE -
- PROXY NONE -
- ACCOUNT account -
- USERNAME user-id -
- PASSWORD password
NCP> EXIT
$ SET PROCESS/PRIVILEGE=NOSYSPRV
```

V5.0

On page MON-97, the contents of a file named SUBMON.COM are listed. This file is shown as part of an example of a MONITOR data collection procedure and is included in the SYS\$EXAMPLES directory. The following lines of that file establish working set values of 100:

```
/WORKING_SET=100 -
/MAXIMUM_WORKING_SET=100 -
```

These values are too low. They should be changed to a higher value such as 512, in order to avoid excessive paging by the detached MONITOR process. You may also want to consider raising the working set extent from 512 to 1024 or higher.

5.18 VMS Network Control Program Manual

V5.0

On page NCP-10 of the *VMS Network Control Program Manual*, the description of the value of an object-name should read as follows:

A string of up to 12 characters, consisting of alphanumeric characters, the dollar sign (\$), or the underscore (_).

5.19 VMS Networking Manual

V5.2

On page 3-58 of the *VMS Networking Manual*, the following description of correcting asynchronous buffer problems is added.

An insufficient number of receive buffers on asynchronous DDCMP lines can cause such network problems as timeouts and loss of packets. If these problems occur, you can enter the Network Control Program (NCP) command SHOW CIRCUIT to confirm whether an insufficient number of receive buffers is the cause:

```
$ RUN SYS$SYSTEM:NCP
NCP> SHOW CIRCUIT TT-0-0 COUNTERS
```

Check the Remote Buffer Errors listed for the circuit. If the counters show any Remote Buffer Errors that include the words "buffer unavailable," you should increase the number of receive buffers for the line. First, use the NCP command SHOW LINE *line-id* CHARACTERISTICS to find out the current number of receive buffers for the line, as in the following command:

Documentation Release Notes

5.19 VMS Networking Manual

```
NCP> SHOW LINE TT-0-0 CHARACTERISTICS
```

The resulting display lists the characteristics for the line, including the number of receive buffers. For example:

```
Line Volatile Characteristics as of 5-APR-1989 9:32:50
```

```
Line = TT-0-0
Receive Buffers      = 4
Controller           = normal
Duplex               = full
Protocol             = DDCMP point
Retransmit timer     = 3000
Line speed           = 9600
Switch               = disabled
Hangup               = disabled
```

Then use the NCP command SET LINE to change the number of receive buffers in the volatile database. In the following example, the number of receive buffers shown in the previous example (four buffers) is increased to six:

```
NCP> SET LINE TT-0-0 STATE OFF
NCP> SET LINE TT-0-0 RECEIVE BUFFERS 6
NCP> SET LINE TT-0-0 STATE ON
```

To change the number of receive buffers in the permanent database as well, use the NCP command DEFINE LINE.

V5.0

On page 3-59, Section 3.6.2.2, the last paragraph should read as follows:

This feature can be used to maximize performance over high-speed links such as Ethernet and the CI. To maximize performance on the Ethernet, one would use a large value for the BUFFER SIZE parameter, which would cause all logical links between adjacent nodes on the Ethernet to use that larger message size. This maximization would also work for a CI. However, on a CI the BUFFER SIZE parameter must be less than or equal to the SYSGEN parameter SCSMAXDG. Failure to do this will result in an unusable CI circuit.

5.20 VMS Record Management Services Manual

V5.0

The *VMS Record Management Services Manual* contains the following error in the second example for the \$PUTMSG system service:

```
NEWSIGARGS (ELEMENT) = 10
```

The correct example is as follows:

```
NEWSIGARGS (ELEMENT) = MIN (SIGARGS (1) - 2, 10)
```

V5.4

Because VMS RMS now supports the use of the Asynchronous option for process-permanent files, make the following changes:

- In Section 5.16, delete the following text under the heading **FAB\$V_ ASY**:

... and is ignored for process-permanent files ...

- In Section 7.19, delete the following text under the heading **RAB\$V_**
ASY:
... ; it is ignored for process-permanent files ...

5.21 VMS RTL Library (LIB\$) Manual

The release notes in this section pertain to the *VMS RTL Library (LIB\$) Manual*.

5.21.1 LIB\$ADAWI Routine

V5.0

On page LIB-3, the LIB\$ADAWI routine description should read “Add Aligned Word with Interlock” instead of “Add Adjacent Word with Interlock”.

The **result** argument description incorrectly implies that it is the sum. **Result** is actually -1, 0, or 1, denoting the sign of the **sum** argument.

5.21.2 LIB\$CREATE_VM_ZONE Routine

V5.0

On pages LIB-44 and LIB-47, the description of the LIB\$CREATE_VM_ZONE routine lists the argument *number-of-areas*; however, *number-of-areas* does not exist. LIB\$CREATE_VM_ZONE has 13, not 14, arguments.

5.21.3 LIB\$SPAWN Routine

V5.0

On page LIB-382, the last argument to LIB\$SPAWN is omitted. The last argument, **table**, is described in the following section.

table

VMS usage: **char_string**
type: **character string**
access: **read only**
mechanism: **by descriptor**

The **table** argument is the address of this file specification string's descriptor. The table specified must reside in SYS\$SHARE with a file type of EXE, and it must be installed.

If omitted, the subprocess uses the same table as the parent process.

The following are general notes about LIB\$SPAWN omitted from the *VMS RTL LIB\$ (Library) Manual*.

Though the subprocess inherits the caller's process privileges as its own process privileges, the set of authorized privileges in the subprocess is inherited from the caller's current privileges.

If the calling image is installed with elevated privileges, it should disable those privileges around the call to LIB\$SPAWN unless the environment of the subprocess is strictly controlled. Otherwise, there is a possibility of a security breach due to elevated privileges accidentally being made available to the user in the spawned process.

Documentation Release Notes

5.21 VMS RTL Library (LIB\$) Manual

The `cli` argument must be specified in all uppercase characters.

5.21.4 **LIB\$SYS_TRNLOG Routine**

V5.1

The RTL routine `LIB$SYS_TRNLOG` was removed from the VMS Version 5.0 *VMS RTL Library (LIB\$) Manual* because the system service that it calls, `$TRNLOG`, is obsolete. However, the RTL routine `LIB$SYS_TRNLOG` is not obsolete. Following is the routine description for `LIB$SYS_TRNLOG` that should have appeared in the *VMS RTL Library (LIB\$) Manual* for VMS Version 5.0. See the *VMS Obsolete Features Manual* for a description of the system service `$TRNLOG`.

LIB\$SYS_TRNLOG

destination-string

VMS usage: **char_string**
type: **character string**
access: **write only**
mechanism: **by descriptor**

Destination string into which LIB\$SYS_TRNLOG writes the logical name translation. The **destination-string** argument contains the address of a descriptor pointing to this destination string.

byte-integer-table

VMS usage: **byte_signed**
type: **byte integer (signed)**
access: **write only**
mechanism: **by reference**

Logical name table number. The **byte-integer-table** argument contains the address of a signed byte integer that is this table number.

access-mode

VMS usage: **access_mode**
type: **byte integer (unsigned)**
access: **write only**
mechanism: **by reference**

Access mode of entry (process table only). The **access-mode** argument contains the address of a signed byte integer that is this access mode.

The access modes, their numeric values, and symbolic names are as follows:

Mode	Value	Symbolic Name
Kernel	0	PSL\$C_KERNEL
Executive	1	PSL\$C_EXEC
Supervisor	2	PSL\$C_SUPER
User	3	PSL\$C_USER

byte-integer-disable-mask

VMS usage: **mask_byte**
type: **byte (unsigned)**
access: **read only**
mechanism: **by reference**

Table search disable mask. The **byte-integer-disable-mask** argument contains the address of a mask byte that is this mask.

The argument **byte-integer-disable-mask** is passed to this routine by reference and is changed to value for use by \$TRNLOG.

The mask-bit settings and their resultant actions are as follows:

Bit	Action
0	Do not search system logical name table
1	Do not search group logical name table
2	Do not search process logical name table

DESCRIPTION See the *VMS System Services Reference Manual* for a complete description of \$TRNLOG.

**CONDITION
VALUES
RETURNED**

SS\$_NORMAL	Routine successfully completed.
SS\$_NOTRAN	Successfully completed, but the input logical name string was placed in the output buffer because no equivalence name was found.
LIB\$_STRTRU	Successfully completed, but the source string was truncated on copy.
SS\$_ACCVIO	The caller cannot read the logical name string or the string descriptor, or cannot write the output length, output buffer, table, or access mode field.
SS\$_IVLOGNAM	The specified logical name string has a length of zero or is greater than 255 characters. String was placed in the destination string buffer because no equivalence name was found.
SS\$_RESULTOVF	The destination string buffer has a length of zero, or it is smaller than the resultant string.
LIB\$_FATERRLIB	Fatal internal error. An internal consistency check has failed. This usually indicates an internal error in the Run-Time Library and should be reported to Digital in a Software Performance Report (SPR).
LIB\$_INVSTRDES	Invalid string descriptor. A string descriptor has an invalid value in its DSC\$_CLASS field.
LIB\$_INSVIRMEM	Insufficient virtual memory. A call to LIB\$GET_VM has failed because your program has exceeded the image quota for virtual memory.

Documentation Release Notes

5.22 VMS RTL Screen Management (SMG\$) Manual

5.22 VMS RTL Screen Management (SMG\$) Manual

V5.4 On page 1-4 of the *VMS RTL Screen Management (SMG\$) Manual*, the first paragraph in Pasteboards, Section 1.1, incorrectly states that a pasteboard can be larger or smaller than the physical screen. In fact, a pasteboard cannot be larger than the physical screen as set in the DCL command SET TERMINAL/WIDTH=*x*.

5.23 VMS System Manager's Manual

V5.4 Add the following information to Section 2.9.1 of the *VMS System Manager's Manual*:

If the workstation is running DECwindows, the system manager should enter the following commands during the conversational bootstrap:

```
SYSBOOT> SET WINDOW_SYSTEM 0
SYSBOOT> SET UAFALTERNATE 1
SYSBOOT> CONTINUE
```

Follow the directions as documented in steps 1 through 5. After logging in to the console terminal, enter the following command lines:

```
$ DEFINE/SYSTEM/EXECUTIVE_MODE SYSUAF SYS$$SYSTEM:SYSUAF.DAT
$ SET DEFAULT SYS$$SYSTEM
$ RUN AUTHORIZE
```

Correct the problem that caused you to be locked out of the system. That is, make the necessary changes to the UAF. Using SYSGEN as follows, set the WINDOW_SYSTEM parameter and clear the UAFALTERNATE parameter:

```
$ RUN SYS$$SYSTEM:SYSGEN
SYSGEN> SET WINDOW_SYSTEM 1
SYSGEN> SET UAFALTERNATE 0
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
```

Shut down and reboot the system.

5.24 VMS System Services Reference Manual

The following sections describe corrections in the *VMS System Services Reference Manual*.

5.24.1 \$ASCEF Service

V5.4 On page SYS-17, the following error messages do not refer to memory that is shared by the multiple processors of an SMP system but to memory coupling of otherwise independent VAX 11/780 systems through shared memory (MA780):

- SS\$_NOTCREATOR
- SS\$_NOSHMBLOCK

Documentation Release Notes

5.24 VMS System Services Reference Manual

- SS\$_SHMNOTCNT

These error messages might refer to mailboxes, event flag clusters, and global sections created in MA780 shared memory.

The SS\$_SHMNOTCNT message has been revised to read as follows:

The shared memory named in the **name** argument is not known to the system. This error can be caused by a spelling error in the string, an improperly assigned logical name, or failure to identify the multiport memory as shared at system generation time.

5.24.2 \$ASCTIM Service

V5.4 On page SYS-18, the **timadr** argument is described as an unsigned quadword. The **timadr** argument is a signed quadword.

5.24.3 \$BINTIM Service

V5.4 On page SYS-27, the **timadr** argument is described as an unsigned quadword. The **timadr** argument is a signed quadword.

5.24.4 \$CANCEL Service

V5.4 On page SYS-39, the overview description for \$CANCEL implies that a channel can be shared by more than one process. The overview now reads as follows:

To cancel I/O on a channel, the access mode of the calling process must be equal to or more privileged than the access mode that the process had when it originally made the channel assignment.

On page SYS-39, the description for the **chan** argument erroneously states that the **chan** argument is a longword. The **chan** argument is a word, not a longword.

5.24.5 \$CHANGE_ACL Service

V5.4 On pages SYS-47 through SYS-49, the **itmlst** description does not explain the interrelationship of ACE operations. \$CHANGE_ACL returns an ACE as the result of read, grant, and find operations. In subsequent read, grant, and find types of operations, the service returns not the same ACE, but rather the next ACE meeting the desired criteria. With a find ACL operation, however, the behavior is slightly different. A read or grant following a FNDACLENT uses the ACE located with the FNDACLENT to perform the read or grant operation.

Documentation Release Notes

5.24 VMS System Services Reference Manual

On page SYS-48, please note in the `ACL$C_ADDACLNT` description that `$CHANGE_ACL` returns an error for a `READACE`, `FNDACETYP`, or `GRANT_ACE` operation in which the buffer is too small to hold the entire ACE. The operation attempts to move as much of the ACE as possible, truncating where necessary, and returns the status `SS$_IVBUFLN`. A subsequent read ACE, find ACE type, or grant ACE operation returns not the same ACE, but the next one that meets the desired criteria.

Also on page SYS-48, please note that `$CHANGE_ACL` returns an error in a `READACL` operation when a buffer is too small to hold at least one ACE. The following read or find operation starts with the ACE following the last one moved to the buffer. As long as `$CHANGE_ACL` moves one ACE, the operation returns success status. However, when the first ACE does not fit in the buffer, `$CHANGE_ACL` truncates the ACE and returns the status `SS$_IVBUFLN`. The subsequent read operation returns the next ACE.

5.24.6 `$CHKPRO` Service

V5.2

On page SYS-57, it is not clear that, when item codes are not specified, the routine uses default values. In all cases except the item code `CHP$_ACMODE`, the routine uses the value of the current process. If the `CHP$_ACMODE` item code is not specified, the routine uses the kernel mode value, which is 0. Therefore, if `CHP$_ACMODE` is not specified, the routine compares two default values, which are both 0, and the check succeeds.

On page SYS-61, the explanation of the item code `CHP$_MATCHDACE` states that the Access Control Element (ACE) returned from the object's Access Control List (ACL) allows the accessor to access the object. It should state that the ACE returned from the object's ACL can either allow or deny access to the object.

On page SYS-61, in the table explaining the item code `CHP$_PRIVUSED`, a V, which indicates a field offset symbol, was omitted from the following symbols:

- `CHP$_V_SYSPRV`
- `CHP$_V_GRPPRV`
- `CHP$_V_BYPASS`
- `CHP$_V_READALL`

On page SYS-62, there are two values to be added to the list of Condition Values Returned. `$CHKPRO` can also return:

<code>SS\$_ACLFULL</code>	More than 20 <code>CHP\$_ACL</code> items were given.
<code>SS\$_RIGHTSFULL</code>	More than 11 <code>CHP\$_ADDRIGHTS</code> items were given.

5.24.7 **\$CMKRNL Service**

V5.2

On page SYS-66, the description of \$CMKRNL has changed. The VMS Version 5.2 documentation notes that programs should not use registers R2 through R11 to pass context between the calling and called procedures. Prior to VMS Version 5.0, the system service dispatcher modified only R4. The system service dispatcher now modifies R0, R1, R2 and R4 before entry into the target routine.

The VAX calling standard states that registers R2 through R11 must be preserved across procedure calls. Called procedures can modify these registers as long as they are saved and restored using the procedure entry mask mechanism. VMS system services, including the \$CMKRNL system service, are invoked as called procedures, and may choose to use R2 through R11 as permitted by the calling standard.

Because the system service dispatcher did not previously use some registers that are reserved for its use, some privileged user programs have used these registers to pass context to called procedures. Those programs that used the R2 register no longer execute correctly. Programs that use registers between R3 and R11 may also produce errors in the future.

Application programs should pass context by using argument blocks rather than by using registers. See the *Introduction to VMS System Routines* for more information.

5.24.8 **\$CRELNM Service**

V5.2

On page SYS-70, the description of the **itmlst** argument erroneously states that the **itmlst** argument is required; it is not.

5.24.9 **\$CREMBX Service**

V5.4

On page SYS-83, the description for the **bufquo** argument erroneously states that the **bufquo** argument is a longword. The **bufquo** argument is a word, not a longword.

On page SYS-83, the description of the **promsk** argument erroneously states that the logical access bit must be clear for all categories of user. This statement has been changed to the following:

The logical access bit must be clear for the class of user requiring access to the mailbox.

On page SYS-84, the last sentence in the description of the **acmode** argument states that the most privileged access mode used is the access mode of the caller. This explanation has been expanded to the following:

The specified access mode and the access mode of the caller are compared. The less privileged (but the higher numeric valued) of the two access modes becomes the access mode associated with the assigned channel. I/O operations on the channel can be performed only from equal and more privileged access modes.

Documentation Release Notes

5.24 VMS System Services Reference Manual

On page SYS-86, the following error messages do not refer to memory that is shared by the multiple processors of an SMP system. They refer to memory coupling of otherwise independent VAX 11/780 systems through shared memory (MA780):

- SS\$_NOTCREATOR
- SS\$_NOSHMBLOCK
- SS\$_SHMNOTCNT

These error messages might refer to mailboxes, event flag clusters, and global sections created in MA780 shared memory.

The SS\$_SHMNOTCNT message has been revised to read as follows:

The shared memory named in the **name** argument is not known to the system. This error can be caused by a spelling error in the string, an improperly assigned logical name, or failure to identify the multiport memory as shared at system generation time.

5.24.10 \$CREPRC Service

V5.4

On page SYS-89, the description of the **prvadr** argument does not mention the default action if the parameter is not specified. If the **prvadr** argument is not specified, the current privileges are used.

On page SYS-95, the description of the **baspri** argument does not specify that, if the **baspri** argument is omitted when using the BLISS or MACRO predefined system service macros, a default of 2 is used. That paragraph of the **baspri** argument description now reads as follows:

If the **baspri** argument is not specified, the priority defaults to 2 for VAX MACRO and VAX BLISS-32 and to 0 for all other languages. If you want a subprocess to have a higher priority than its creator, you must have the ALTPRI privilege to raise the priority level. If the caller does not have this privilege, the specified base priority is compared with the caller's priority and the lower of the two values is used.

5.24.11 \$CRMPSC Service

V5.2

On page SYS-107, the last entry in the flag table that describes the flag SEC\$_NO_OVERMAP is misleading because it implies that setting the flag SEC\$_NO_OVERMAP allows sections to overmap existing address space. In fact, the opposite is true. The description of the flag SEC\$_NO_OVERMAP should be:

Pages cannot overmap existing address space. Note that, by default, pages can overmap existing address space.

Documentation Release Notes

5.24 VMS System Services Reference Manual

V5.4

On page SYS-115, the following status value has been added to the list of Condition Values Returned:

SS\$_VA_IN_USE A page in the specified input address range is already mapped and the flag SEC\$_NO_OVERMAP is set.

On page SYS-113, the following error messages do not refer to memory that is shared by the multiple processors of an SMP system, but to memory coupling of otherwise independent VAX 11/780 systems through shared memory (MA780).

- SS\$_NOTCREATOR
- SS\$_NOSHMBLOCK
- SS\$_SHMNOTCNT

These error messages might refer to mailboxes, event flag clusters, and global sections created in MA780 shared memory.

The SS\$_SHMNOTCNT message has been revised to read as follows:

The shared memory named in the **name** argument is not known to the system. This error can be caused by a spelling error in the string, an improperly assigned logical name, or failure to identify the multiport memory as shared at system generation time.

5.24.12 \$DCLCMH Service

V5.2

On page SYS-123, the VMS usage, type and access of the **address** argument are listed as follows:

address
VMS usage: **procedure**
type: **procedure entry mask**
access: **call without stack unwinding**
mechanism: **by reference**

The VMS usage, type and access of the **address** argument have been changed as follows:

address
VMS usage: **address**
type: **longword (unsigned)**
access: **read only**
mechanism: **by reference**

5.24.13 \$DEQ Service

V5.4

The following explanation has been added to the description of the LCK\$_CANCEL flag:

When you specify this flag, \$DEQ attempts to cancel a lock request that was queued by \$ENQ. You can only cancel a waiting request. When the request is canceled, \$DEQ returns the condition value SS\$_NORMAL.

Documentation Release Notes

5.24 VMS System Services Reference Manual

If you attempt to cancel a granted lock, the request fails and \$DEQ returns the condition value SS\$_CANCELGRANT. Two types of waiting requests can be canceled:

- 1 A request for a new lock
- 2 A request to convert an existing lock

When canceling a new lock request, the following actions occur:

- If a completion AST was requested, the AST is queued for delivery and SS\$_ABORT is stored in the lock status block.

When canceling a request to convert an existing lock, the conversion request is canceled. The existing granted lock remains unchanged. The following specific actions occur:

- The blocking AST address specified for the existing granted lock is queued for delivery if the granted mode of the existing lock is blocking other waiting requests.
- If a completion AST was specified by the conversion request, the completion AST is queued for delivery with SS\$_ABORT status stored in the lock status block that was specified by the conversion request.

If you specify the LCK\$_DEQALL flag, the LCK\$_CANCEL flag is ignored.

5.24.14 \$DGBLSC Service

V5.4

On page SYS-142, the following error messages do not refer to memory that is shared by the multiple processors of an SMP system. They refer to memory coupling of otherwise independent VAX 11/780 systems through shared memory (MA780).

- SS\$_NOTCREATOR
- SS\$_NOSHMBLOCK
- SS\$_SHMNOTCNT

These error messages might refer to mailboxes, event flag clusters, and global sections created in MA780 shared memory.

The SS\$_SHMNOTCNT message has been revised to read as follows:

The shared memory named in the **name** argument is not known to the system. This error can be caused by a spelling error in the string, an improperly assigned logical name, or failure to identify the multiport memory as shared at system generation time.

5.24.15 \$ENQ Service

V5.2

On page SYS-148, the description of the **efn** argument states that the event flag is set when the lock request has been granted. The event flag is also set if the lock request is canceled. The description of the **efn** argument has been changed to the following:

Number of the event flag to be set when the request has been granted or canceled. Cancellation occurs if you use \$DEQ with the cancel modifier or if the waiting request is chosen to break a deadlock.

On page SYS-154, the description of the **astadr** argument states that this parameter is an "AST service routine to be executed when the lock is either granted or converted." This AST routine is also called when the \$ENQ request is aborted because of deadlock (and the lock status block contains the condition SS\$_DEADLOCK). The description of the **astadr** argument has been changed to the following:

The AST service routine is to be executed when the lock is either granted or converted. The **astadr** argument is the address of the entry mask of this routine.

The AST is also delivered when the lock or conversion request is canceled. Cancellation occurs if you use \$DEQ with the cancel modifier or if the waiting request is chosen to break a deadlock.

5.24.16 \$FAO Service

V5.2

On page SYS-167, the first sentence of the description of \$FAO incorrectly states that "The \$FAO_S macro form uses a PUSHHL instruction for all parameters (**p1** through **pn**)." The \$FAO_S macro actually accepts arguments **P1** to **P17**.

On page SYS-177, on lines 5 and 31 of Example 8, **/5>** should be **/5*>**.

The incorrect example appears as follows:

```
.ASCID /!5> NOW IS: !%D/
```

The corrected example should appear as follows:

```
.ASCID /!5*> NOW IS: !%D/
```

On page SYS-177, on line 5 of Example 9, **/#_** should be **/5*_**. The incorrect example appears as follows:

```
.ASCID /DATE: !11%D!#_TIME: !5%T/
```

The corrected example should appear as follows:

```
.ASCID /DATE: !11%D!5*_TIME: !5%T/
```

V5.4

The documentation of the !AS FAO edit item has been revised to state that \$FAO assumes the descriptor is a CLASS_S (static) string descriptor. Other descriptor types may give improper results.

Documentation Release Notes

5.24 VMS System Services Reference Manual

5.24.17 \$FORMAT_ACL Service

- V5.2 On page SYS-195, the description of the alarm ACE states that ACE\$_AUDITNAME contains a counted ASCII string. There is no requirement for the string to be counted.
- On page SYS-200, in the description of the last table item, ACE\$_L_KEY, the *identifier* field incorrectly states that the number of longwords is implied by ACE\$_B_LENGTH. The number of longwords is implied by ACE\$_B_SIZE.
- V5.4 On page SYS-193, the format line includes two null arguments. The second **nullarg** argument has been eliminated.

5.24.18 \$GETDVI Service

- V5.4 On page SYS-219, the description for SS\$_BADPARAM in the Condition Values Returned section, states that the **return length address** field in an item descriptor specifies less than four bytes for the return length information. The field referred to is the **buffer address** field, not the **return length address** field. The statement should read as follows,
- The **buffer address** field in an item descriptor specifies less than four bytes for the return length information.
- On page SYS-217, the following item codes have been added to the list of item codes that return information about terminal characteristics:
- DVI\$_TT_DECCRT3
 - DVI\$_TT_DECCRT4

5.24.19 \$GETQUI Service

- V5.2 On pages SYS-259 through SYS-282, the following output item codes should be listed as valid for the QUI\$_DISPLAY_ENTRY function of the \$GETQUI system service:
- QUI\$_ASSIGNED_QUEUE_NAME
 - QUI\$_PROCESSOR
 - QUI\$_QUEUE_FLAGS
 - QUI\$_QUEUE_STATUS
 - QUI\$_RESTART_QUEUE_NAME
- The QUI\$_INTERVENING_BLOCKS and QUI\$_INTERVENING_JOBS output item codes are incorrectly listed as valid for the QUI\$_DISPLAY_ENTRY function; they are only supported for the QUI\$_DISPLAY_JOB function.
- On page SYS-284, the QUI\$_SEARCH_NAME input is valid for the QUI\$_TRANSLATE_QUEUE function.

Documentation Release Notes

5.24 VMS System Services Reference Manual

V5.4

On pages SYS-272 and SYS-273, the explanation of the item codes QUI\$_INTERVENING_BLOCKS and QUI\$_INTERVENING_JOBS has been changed to read as follows:

QUI\$_INTERVENING_BLOCKS

When you specify QUI\$_INTERVENING_BLOCKS, \$GETQUI returns, as a longword integer value, the size (in blocks) of files associated with pending jobs in the queue that were skipped during the current call to \$GETQUI. These jobs were not reported because they did not match the selection criterion in effect for the call to \$GETQUI.

QUI\$_INTERVENING_BLOCKS is zero when

- The job is not a pending job
- The job that matches the selection criterion is the first pending job in the queue
- The preceding pending job in the queue was reported in the previous call to \$GETQUI

QUI\$_INTERVENING_JOBS

When you specify QUI\$_INTERVENING_JOBS, \$GETQUI returns, as a longword integer value, the number of pending jobs in the queue that were skipped during the current call to \$GETQUI. These jobs were not reported because they did not match the selection criterion in effect for the call to \$GETQUI.

QUI\$_INTERVENING_JOBS is zero when

- The job is not a pending job
- The job that matches the selection criterion is the first pending job in the queue
- The preceding pending job in the queue was reported in the previous call to \$GETQUI

On page SYS-283, the first flag in the \$QUI\$_SEARCH_FLAGS table, is incorrectly named QUI\$_V_FREEZE_CONTEXT. The correct name is QUI\$_V_SEARCH_FREEZE_CONTEXT.

5.24.20 \$GETSYI Service

V5.2

On page SYS-301, the description of the item codes SYI\$_ACTIVECPU_CNT and SYI\$_AVAILCPU_CNT fails to note that the \$GETSYI service returns this information only for the local VAX node.

The new wording for these item codes is as follows:

\$GETSYI Item Codes

SYI\$_ACTIVECPU_CNT

When you specify SYI\$_ACTIVECPU_CNT, \$GETSYI returns a count of CPUs actively participating in the current boot of the Symmetric Multiprocessing (SMP) system. The \$GETSYI service returns this information for the local node only.

Documentation Release Notes

5.24 VMS System Services Reference Manual

Because this number is a longword, the **buffer length** field in the item descriptor should specify 4 bytes.

SYI\$_AVAILCPU_CNT

When you specify SYI\$_AVAILCPU_CNT, \$GETSYI returns the number of CPUs available in the current boot of the SMP system. The \$GETSYI service returns this information for the local node only.

Because this number is a longword, the **buffer length** field in the item descriptor should specify 4 bytes.

V5.4

On page SYS-303, the table in the description of the SYI\$_CPU item descriptor field of the \$GETSYI system service has been updated to include several corrections and new computers. Replace the existing table with the following updated table:

Processor	Symbol
VAX-11/730	PR\$_SID_TYP730
VAX-11/750	PR\$_SID_TYP750
VAX-11/780, 785	PR\$_SID_TYP780
VAXstation II, II/GPX, and MicroVAX II	PR\$_SID_TYPUV2
VAXstation 2000/MicroVAX 2000	PR\$_SID_TYP410
VAX 8200, 8250, 8300, 8350	PR\$_SID_TYP8SS
VAX 8530, 8550, 8810 (8700), and 8820-N (8800)	PR\$_SID_TYP8NN
VAX 8600, 8650	PR\$_SID_TYP790
VAX 8820, 8830, 8840	PR\$_SID_TYP8PS
VAXft 3000 Model 310	PR\$_SID_TYP520
VAXstation, MicroVAX 3100 series	PR\$_SID_TYP420
MicroVAX 3300,3400,3500,3600,3800,3900	PR\$_SID_TYP650
VAXstation 3520, 3540	PR\$_SID_TYP60
VAX 4000-300	PR\$_SID_TYP670
VAX 6000-200, 6000-300 series	PR\$_SID_TYP9CC
VAX 6000-400 series	PR\$_SID_TYP9RR
VAX 9000-200, 9000-400 series	PR\$_SID_TYP9AQ

On pages SYS-305 and SYS-306, the table in the description of the SYI\$_HW_NAME item descriptor field of the \$GETSYI system service has been updated to include several corrections and new computers. Replace the existing table with the following updated table:

VAX Model Processor Name	VAX Model Type
VAX-11/730	VAX\$K_V730
VAX-11/750	VAX\$K_V750
VAX-11/780	VAX\$K_V780

Documentation Release Notes

5.24 VMS System Services Reference Manual

VAX Model Processor Name	VAX Model Type
VAX-11/785	VAX\$K_V785
MicroVAX II	VAX\$K_VUV2
VAXstation II	VAX\$K_VWS2
VAXstation II/GPX	VAX\$K_VWSD
VAXstation 2000	VAX\$K_VWS2000
MicroVAX 2000	VAX\$K_VUV2000
VAXstation 2000/GPX	VAX\$K_VWSD2000
VAX 8200	VAX\$K_V8200
VAX 8250	VAX\$K_V8250
VAX 8300	VAX\$K_V8300
VAX 8350	VAX\$K_V8350
VAX 8530	VAX\$K_V8500
VAX 8550	VAX\$K_V8550
VAX 8600	VAX\$K_V8600
VAX 8650	VAX\$K_V8650
VAX 8810 (8700)	VAX\$K_V8700
VAX 8820-N (8800)	VAX\$K_V8800
VAX 8820, 8830, or 8840 with one CPU enabled	VAX\$K_V8810
VAX 8820	VAX\$K_V8820
VAX 8830	VAX\$K_V8830
VAX 8840	VAX\$K_V8840
VAXft 3000 Model 310	VAX\$K_V520FT
VAXstation 3520	VAX\$K_V3520L
VAXstation 3540	VAX\$K_V3540L
VAX 4000-300 timeshare	VAX\$K_V670
VAX 4000-300 server	VAX\$K_V670_S
VAX 6000-210 timeshare	VAX\$K_V6210_T
VAX 6000-220 timeshare	VAX\$K_V6220_T
VAX 6000-230 timeshare	VAX\$K_V6230_T
VAX 6000-240 timeshare	VAX\$K_V6240_T
VAX 6000-250 timeshare	VAX\$K_V6250_T
VAX 6000-260 timeshare	VAX\$K_V6260_T
VAX 6000-210 server	VAX\$K_V6210_S
VAX 6000-220 server	VAX\$K_V6220_S
VAX 6000-310 timeshare	VAX\$K_V6310_T
VAX 6000-320 timeshare	VAX\$K_V6320_T
VAX 6000-330 timeshare	VAX\$K_V6330_T
VAX 6000-340 timeshare	VAX\$K_V6340_T

Documentation Release Notes

5.24 VMS System Services Reference Manual

VAX Model Processor Name	VAX Model Type
VAX 6000-350 timeshare	VAX\$K_V6350_T
VAX 6000-360 timeshare	VAX\$K_V6360_T
VAX 6000-310 server	VAX\$K_V6310_S
VAX 6000-320 server	VAX\$K_V6320_S
VAX 6000-410 timeshare	VAX\$K_V9RR10_T
VAX 6000-420 timeshare	VAX\$K_V9RR20_T
VAX 6000-430 timeshare	VAX\$K_V9RR30_T
VAX 6000-440 timeshare	VAX\$K_V9RR40_T
VAX 6000-450 timeshare	VAX\$K_V9RR50_T
VAX 6000-460 timeshare	VAX\$K_V9RR60_T
VAX 6000-410 server	VAX\$K_V9RR10_S
VAX 6000-420 server	VAX\$K_V9RR20_S
VAX 9000-210	VAX\$K_V9AR10
VAX 9000-410	VAX\$K_V9AQ10
VAX 9000-420	VAX\$K_V9AQ20
VAX 9000-430	VAX\$K_V9AQ30
VAX 9000-440	VAX\$K_V9AQ40

On page SYS-309, the table in the description of the SYI\$_XCPU item descriptor field to the \$GETSYI system service has been updated to include several corrections and new computers. Replace the existing table with the following updated table:

VAX Processor Type Symbol	Extended Processor Type	Extended Processor Symbol
PR\$_SID_TYPUV	MicroVAX II VAXstation II	PR\$_XSID_UV_UV2
	MicroVAX 2000 VAXstation 2000	PR\$_XSID_UV_410
PR\$_SID_TYPCV	MicroVAX 3300, 3400, 3500, 3600, 3800, 3900 series	PR\$_XSID_CV_650
	VAX 6000-200, 6000- 300 series	PR\$_XSID_CV_9CC
	VAXstation 3520, 3540	PR\$_XSID_CV_60
	VAXstation 3100 series	PR\$_XSID_CV_420
	VAXft 3000 Model 310	PR\$_XSID_CV_520
PR\$_SID_TYP8NN	VAX 8530	PR\$_XSID_N8500

Documentation Release Notes

5.24 VMS System Services Reference Manual

VAX Processor Type Symbol	Extended Processor Type	Extended Processor Symbol
	VAX 8550	PRS\$_XSID_N8550
	VAX 8810 (8700)	PRS\$_XSID_N8700
	VAX 8820-N (8800)	PRS\$_XSID_N8800
PRS\$_SID_TYPRV	VAX 4000-300	PRS\$_XSID_RV_670
	VAX 6000-400 series	PRS\$_XSID_RV_9RR

5.24.21 \$GETUAI Service

V5.2

On page SYS-316, the description of UAI\$_ACCOUNT states that the buffer length field in the item descriptor should specify 9 bytes. It should specify 32 bytes. That description has been changed to the following:

When you specify UAI\$_ACCOUNT, \$GETUAI sets, as a blank-filled 32-character string, the account name of the user. An account name can include up to 8 characters. Because the account name is a blank-filled string, however, the buffer length field of the item descriptor should specify 32 bytes.

On page SYS-324, UAI\$_USERNAME, a \$GETUAI item code, has been eliminated. UAI\$_USERNAME cannot return the username of the owner of a specified job. UAI\$_USERNAME returns only the username that you enter as an argument. Use \$GETJPI to return job information.

5.24.22 \$MOUNT Service

V5.2

On page SYS-355, the description of the MNT\$_FLAGS option, MNT\$_NOMNTVER, erroneously states that MNT\$_M_NOMNTVER applies only to disks. Since VMS Version 5.0, mount verification has applied to tapes as well as disks.

V5.3

The following text for the /NOREBUILD flag for the \$MOUNT system service was omitted:

The volume to be mounted should be returned to active use immediately, without performing a rebuild operation. A rebuild operation can consume a considerable amount of time, depending on the number of files on the volume and, if quotas are in use, on the number of different file owners. The volume can be rebuilt later with the DCL command SET VOLUME/REBUILD to recover the free space; for more information, see the *VMS DCL Dictionary*.

If a disk volume is dismounted improperly, for example, during a system failure, it must be rebuilt to recover any caching limits that were enabled on the volume at the time of the dismount. By default, \$MOUNT attempts to rebuild the disk volume. A successful rebuild operation includes

Documentation Release Notes

5.24 VMS System Services Reference Manual

reclaiming all the available free space. Therefore, you must mount all of the volume set members.

MNT\$M_NOREBUILD applies only to disks.

On page SYS-355, the description of the \$MOUNT flag, MNT\$M_NOMNTVER, incorrectly states that the flag applies only to disks. As of Version 5.0, mount verification applies to tapes as well as disks.

5.24.23 \$NUMTIM Service

V5.4 On page SYS-367, the **timadr** argument is described as an unsigned quadword. The **timadr** argument is a signed quadword.

5.24.24 \$PARSE_ACL Service

V5.4 On page SYS-368, the format line includes two null arguments. The second **nullarg** argument has been eliminated.

5.24.25 \$PUTMSG Service

V5.2 On page SYS-371, the description of \$PUTMSG is incomplete and has been changed to the following:

The Put Message service is a generalized message-formatting and output routine used by VMS to write informational and error messages to processes. These messages are written to SYS\$ERROR and SYS\$OUTPUT. Informational messages are written to SYS\$OUTPUT only; error messages are written to SYS\$ERROR. Error messages are also written to SYS\$OUTPUT if it has a different definition from SYS\$ERROR.

V5.4 On page SYS-372, the description of the **msgvec** argument is misleading. While describing default message options, two references are made to default message "flags" rather than "options." The description should read as follows:

The \$PUTMSG service passes the **default message options** field to \$GETMSG as the **flags** argument.

If you do not specify the **default message options** field, the default message options for the process are used; you can set the process default message options by using the DCL command SET MESSAGE.

If you specify a value of 0 for the default message options field, the default message options for the process are used.

On page SYS-378, the second example contains an error. This is the incorrect code:

Documentation Release Notes

5.24 VMS System Services Reference Manual

```
ELEMENT = 1
NEWSIGARGS (ELEMENT) = 10

DO I = 1, SIGARGS (1) - 2
  ELEMENT = ELEMENT + 1
  NEWSIGARGS (ELEMENT) = SIGARGS (ELEMENT)
END DO
```

The incorrect code has been replaced by the following corrected code:

```
ELEMENT = 1
  ! NEWSIGARGS (ELEMENT) = 10
NEWSIGARGS (ELEMENT) = MIN (SIGARGS (1) - 2, 10)
```

The entire corrected example, which will appear in the next release of the *VMS System Services Reference Manual*, is as follows:

```
INTEGER STATUS,
2     OLDHND

CHARACTER*5 NUM

INCLUDE ' ($SSDEF) '
INCLUDE ' ($LIBDEF) '

INTEGER LIB$GET_INPUT,
2     LIB$ESTABLISH,
2     SYS$GETJPI
EXTERNAL ERR

OPEN (UNIT = 1,
2     TYPE = 'NEW',
2     CARRIAGECONTROL = 'LIST',
2     FILE = 'ERROR.LOG')

OLDHND = LIB$ESTABLISH (ERR)

! This routine executes successfully
STATUS = LIB$GET_INPUT (NUM, 'NUM: ')
IF (.NOT. STATUS) CALL LIB$SIGNAL (%VAL (STATUS))
! This routine fails with insufficient arguments
STATUS = SYS$GETJPI (,)
IF (.NOT. STATUS) CALL LIB$SIGNAL (%VAL (STATUS))

END

INTEGER FUNCTION ERR (SIGARGS,
2     MECHARGS)

INTEGER SIGARGS (*),
2     MECHARGS (*)
INTEGER NEWSIGARGS (10), ! Must specify a length for
                        ! array so choose one large enough
                        ! to cover any eventuality
2     ELEMENT

INCLUDE ' ($SSDEF) '

EXTERNAL PUT_LINE
INTEGER PUT_LINE

! Get rid of last two elements in SIGARGS (the PC and PSL),
! then pad NEWSIGARGS with zeros.

ELEMENT = 1
  ! NEWSIGARGS (ELEMENT) = 10
NEWSIGARGS (ELEMENT) = MIN (SIGARGS (1) - 2, 10)
```

Documentation Release Notes

5.24 VMS System Services Reference Manual

```
DO I = 1, SIGARGS(1) - 2
  ELEMENT = ELEMENT + 1
  NEWSIGARGS (ELEMENT) = SIGARGS (ELEMENT)
END DO
DO I = ELEMENT + 1, 10
  ELEMENT = ELEMENT + 1
  NEWSIGARGS (ELEMENT) = 0
END DO

CALL SYS$PUTMSG (NEWSIGARGS, PUT_LINE,)
ERR = SS$_RESIGNAL
                                ! Could use CONTINUE and let $PUTMSG
                                ! write the message

END
INTEGER FUNCTION PUT_LINE (LINE)
CHARACTER*(*) LINE

PUT_LINE = 0                    ! Since you're resignalling, don't let
                                ! SYS$PUTMSG write the error.

WRITE (UNIT = 1,
       2      FMT = '(A)') LINE
END
```

5.24.26 \$QIO Service

- V5.2** On page SYS-379, the **func** argument is described as both a word and a longword. The **func** argument is a word, not a longword, value.
- V5.4** On page SYS-380, the **chan** argument is described as a longword value. The **chan** argument is a word, not a longword.

5.24.27 \$SETDIR Service

- V5.2** On page SYS-516, the following format statement implies that **length-addr** and **cur-dir-addr** are optional; they are not.

```
[new-dir-addr] [,length-addr] [,cur-dir-addr]
```

The correct format statement is as follows:

```
[new-dir-addr] [,length-addr] [,cur-dir-addr]
```

5.24.28 \$SETEXV Service

- V5.2** The description of the **PRVHND** argument for the **\$SETEXV** system service uses the service name **\$SETEF** rather than **\$SETEXV**. The description should read as follows:

Previous condition handler address contained by the specified exception vector. The **prvhnd** argument is the address of a longword into which **\$SETEXV** writes the handler address.

5.24.29 \$SETIMR Service

V5.4 On page SYS-406, the **daytim** argument is described as an unsigned quadword. The **daytim** argument is a signed quadword.

5.24.30 \$SETPRV Service

V5.2 On page SYS-419, the **prmflg** argument to the \$SETPRV system service is incorrectly labeled **get jobprmflg**.

The arguments **enbflg** and **prmflg** are longword values, not byte values.

5.24.31 \$SETSFM Service

V5.4 Beginning with VMS Version 5.4, the \$SETSFM system service is obsolete. It is not being replaced by another system service. Although \$SETSFM is supported, Digital does not recommend use of this service, as it may cause unexpected failures, particularly with various language support routines and run-time libraries.

5.24.32 \$SETSSF Service

V5.4 Beginning with VMS Version 5.4, the \$SETSSF system service is obsolete. It is not being replaced by another system service. Although \$SETSSF is supported, Digital does not recommend use of this service, as it may cause unexpected failures, particularly with various language support routines and run-time libraries.

5.24.33 \$SETUAI Service

V5.2 On page SYS-432, the description of UAI\$_ACCOUNT states that the buffer length field in the item descriptor should specify 9 bytes. It should specify 32 bytes. That description has been changed to the following:

When you specify UAI\$_ACCOUNT, \$SETUAI sets, as a blank-filled 32-character string, the account name of the user. An account name can include up to 8 characters. Because the account name is a blank-filled string, however, the buffer length field of the item descriptor should specify 32 bytes.

On page SYS-439, UAI\$_USERNAME, a \$SETUAI item code, has been eliminated. The item code UAI\$_USERNAME cannot be used to set the username of the owner of a specified job.

5.24.34 \$SNDJBC Service

V5.2 On page SYS-469, the SJC\$_CREATE_JOB function code has the following new item codes:

- SJC\$_FIRST_PAGE
- SJC\$_NO_FIRST_PAGE

Documentation Release Notes

5.24 VMS System Services Reference Manual

- SJC\$_LAST_PAGE
- SJC\$_NO_LAST_PAGE

On page SYS-491, JBC\$_NOSUCHENT should be included as a valid Condition Value Returned in the I/O Status Block as follows:

JBC\$_NOSUCHENT There is no job with the specified entry number.

V5.4

On page SYS-490, SS\$_IVLOGNAM has been added to the list of Condition Values Returned:

5.24.35 \$SNDOPR Service

V5.2

The following status value has been added to the list of Condition Values Returned:

OPC\$_NOOPERATOR The Operator Communication Manager (OPCOM) is not running; the message will not be sent.

5.24.36 \$UNWIND Service

V5.4

On page SYS-530, the documentation incorrectly states that the passing mechanism for the **newpc** argument is by reference. The **newpc** argument is passed by value.

The correct wording is as follows:

newpc
VMS usage: **address**
type: **longword (unsigned)**
access: **read only**
mechanism: **by value**

5.24.37 \$UPDSEC Service

V5.4

On page SYS-535, the following error messages do not refer to memory that is shared by the multiple processors of an SMP system. They refer to memory coupling of otherwise independent VAX 11/780 systems through shared memory (MA780).

- SS\$_NOTCREATOR
- SS\$_NOSHMBLOCK
- SS\$_SHMNOTCNT

These error messages might refer to mailboxes, event flag clusters, and global sections created in MA780 shared memory.

The SS\$_SHMNOTCNT message has been revised to read as follows:

The shared memory named in the **name** argument is not known to the system. This error can be caused by a spelling error in the string, an improperly assigned logical name, or failure to identify the multiport memory as shared at system generation time.

5.24.38 \$WFLOR Service

V5.2

On page SYS-543, the description of \$WFLOR states that, if all the specified event flags have been set, the process resumes execution. All event flags do not have to be set; if any of the specified event flags have been set, the process resumes execution.

5.24.39 Error Messages Referring to Shared Memory

V5.4

In several system services, the following error messages can be misleading:

SS\$_NOTCREATOR
SS\$_NOSHMBLOCK
SS\$_SHMNOTCNT

These error messages do not refer to memory that is shared by the multiple processors of a Symmetric Multiprocessing system. They refer to memory coupling of otherwise independent VAX 11/780 systems through shared memory (MA780). These error messages might refer to mailboxes, event flag clusters, and global sections created in MA780 shared memory.

The following system services contain these error messages.

\$ASCEFC
\$CREMBX
\$CRMPSC
\$DGBLSC
\$UPDSEC

5.24.40 Self-Modifying Item Lists

V5.4

A problem can occur if you use self-modifying item lists with the following services:

\$GETDVI
\$GETDVIW
\$GETJPI
\$GETJPIW
\$GETLKI
\$GETLKIW
\$GETMSG
\$GETQUI
\$GETQUIW
\$GETSYI
\$GETSYIW
\$GETTIM
\$GETUAI

When any one of these services collects data, it makes multiple passes through the item list. The number of passes needed depends on which item codes are referenced and the state of the target process. If the item list is self-modifying, that is, if the addresses for the output buffers in the item list point back to the item list, the service replaces the item-list information with the collected data. Therefore, incorrect data may be

Documentation Release Notes

5.24 VMS System Services Reference Manual

returned or unexpected errors may occur when the service reads the item list again.

A program using self-modifying item lists that appears to work normally can fail when a system has processes that are swapped out of memory or when a process is on a remote node. A heavy system load or the order of the item list entries can also cause such a program to fail.

To prevent confusing errors, Digital recommends that you not use self-modifying item lists.

5.25 VMS Upgrade and Installation Supplement: VAXstation 2000, MicroVAX 2000

V5.4

The table in Chapter 2 of the *VMS Upgrade and Installation Supplement: VAXstation 2000, MicroVAX 2000* contains incorrect information about the label of the tape cartridge that contains standalone BACKUP and VMS DECwindows software. Replace the existing table with the following new table, which contains the correct information:

Paper Label ¹	Volume Label ¹
VAX/VMS V5.4 BIN TK50 2/2 DECWINDOWS & S/A BKUP	DECW54

¹A volume label is the name the VMS operating system uses to refer to the tape cartridge. During the installation, the procedure displays the volume label, not the paper label, in messages.

Section 1.3.2 describes how to attach a diagnostic console terminal to the system. If you use a hardcopy terminal as your diagnostic console terminal, you can get a printout of the installation or upgrade.

If you use a video terminal as your diagnostic console terminal, you should attach a printer to the video terminal using the second procedure described in Section 1.3.2. Note that this procedure refers to a BCC05 cable. Depending on the type of video terminal you have, you might need a different cable. For more information, refer to the documentation for your video terminal.

5.26 VMS Upgrade and Installation Supplement: VAX 8600, 8650

V5.4

On page 4-6 of the *VMS Upgrade and Installation Supplement: VAX 8600, 8650*, the command for booting standalone BACKUP is incorrect. To boot standalone BACKUP from an RL02 disk, enter the following command line:

```
>>> B CS1
```

5.27 VMS Upgrade and Installation Supplements

V5.4 Several of the VMS installation and upgrade supplements contain references to the *VAX Volume Shadowing Manual* for information about depositing a value in register 3 (R3).

If you are using Volume Shadowing phase I, you must deposit the physical and virtual unit numbers of the device in R3 (with the physical unit number in the low-order word and the virtual unit number in the high-order word). For more information, see the *VAX Volume Shadowing Manual*.

If you are using Volume Shadowing phase II, you must deposit the physical unit number of the drive in R3. Also, you must specify the virtual unit number by setting the SYSGEN parameters SHADOW_SYS_DISK and SHADOW_SYS_UNIT. For more information, see the *VMS Volume Shadowing Manual*.

Note: With VMS Version 5.4, the name of the product has changed from VAX Volume Shadowing to VMS Volume Shadowing. For details about the name change, see Section 1.11.

5.28 VMS Utility Routines Manual

V5.4 On page ACL-4 of the *VMS Utility Routines Manual*, the table of item identifiers for the ACL Editor Routine has been extended to include the following identifier:

ACLEDIT\$C_JOURNAL_NAME	Allows the caller to supply a character string with the name of the journal file.
-------------------------	---

5.29 VMS Version 5.3 New Features Manual

V5.4 In the *VMS Version 5.3 New Features Manual*, add the following entry to the list of Condition Values Returned for SYS\$DNS:

SS\$_BADPARAM Bad parameter value

This entry has been included in the *VMS Version 5.4 New Features Manual*.

5.30 VMS Version 5.4 New Features Manual

V5.4 In Chapter 12 of the *VMS Version 5.4 New Features Manual*, on the System Generation Utility (SYSGEN), the description incorrectly states that the new SYSGEN parameter SCSI_NOAUTO replaces the SYSGEN parameter VM\$D1. In fact, the SCSI_NOAUTO parameter replaces the SYSGEN parameter VM\$D2.

A

New and Modified System Messages for VMS Version 5.4

The following VMS Facilities have new or modified system message information:

- BUGCHECK (System Bugcheck)
- DISMOUNT (DISMOUNT command)
- INIT (INITIALIZE command)
- LMCP (Log Manager Control Program)
- MOUNT (Mount Utility)
- OPCOM (Operator Communication Process)

A.1 VMS System Error Messages

BAD_SIZE, permitted minimum is 100 blocks

Facility: LMCP, Log Manager Control Program

Explanation: An attempt was made to create a log file that was too small to use.

User Action: Re-create a log file with a larger file-size specification.

INCSHAMEM, system disk membership inconsistency

Facility: INIT, INITIALIZE Command

Explanation: The boot device is not currently a source member of the shadow set. One or more of the shadow set members named in the Storage Control Block (SCB) of the boot device is inaccessible.

User Action: None.

MSNGENDS, missing or misspelled ENDSUBROUTINE statement detected while scanning for label

Facility: DCL, DCL Dictionary

Explanation: A SUBROUTINE command without an ending ENDSUBROUTINE command or a misspelled ENDSUBROUTINE command was detected while performing a CALL command. This condition may have caused the CALL to be unable to locate a destination label even though the label existed. For correct usage of the ENDSUBROUTINE command, see Section 2.5.6.

User Action: Check the command procedure for one or more missing or misspelled ENDSUBROUTINE commands; correct as necessary.

New and Modified System Messages for VMS Version 5.4

A.1 VMS System Error Messages

NOALOCCLASS, allocation class not allowed with shadowing phase II virtual unit name

Facility: MOUNT, Mount Utility

Explanation: An allocation class was specified in the name of the virtual unit. Unlike shadowing phase one, an allocation class is not allowed in the name of the virtual unit when using shadowing phase II.

User Action: Reenter the command without specifying an allocation class on the virtual unit. The virtual unit must be specified in the form DSA or DSAnnnn, where nnnn represents a unique number from 0 to 9999.

NOINFO, no information in database

Facility: NCP, Network Control Program

Explanation: An NCP command (usually a SET command) was executed when there was no data to act upon in the database. This error commonly occurs during system startup when a SET KNOWN *component* ALL command is executed and there is no data to be copied into the volatile database from the permanent database. If the error occurs during system operation, a command has attempted to manipulate data that does not exist; for example, a command specifies a nonexistent component.

User Action: Ignore this error if it occurs during system startup. If you receive this error during system operation, reissue the command specifying an existing system component.

NOREMBROAD, no VAXcluster terminals were notified because OPCOM is not available

Facility: OPCOM, Operator Communication

Explanation: A REPLY command was issued that wants to send a message to terminals on other nodes within a VAXcluster, but the OPCOM process is not available to satisfy this request. This message is only set to terminals on the local node.

User Action: Restart the Operator Communications Process (OPCOM) with the following command:

```
$ @SYSS$SYSTEM:STARTUP OPCOM
```

NOREMWAIT, /WAIT requested, therefore no VAXcluster terminals notified

Facility: OPCOM, Operator Communication

Explanation: A REPLY command was issued that wants to send a message to terminals on other nodes within a VAXcluster, but the /WAIT qualifier was specified, requesting that the message be sent synchronously.

User Action: If the message must be delivered on terminals on other VAXcluster nodes, reissue the command without the /WAIT qualifier.

New and Modified System Messages for VMS Version 5.4

A.1 VMS System Error Messages

SHADBOOTFAIL, shadowing failed to boot from system disk shadow set

Facility: BUGCHECK, System Bugcheck

Explanation: The following conditions can cause this error:

- Failed to allocate memory.
- One or more critical devices are inaccessible.
- Boot device is a target of a full copy operation.
- Boot device is not a source member of the existing shadow set

User Action: Perform one or more of these user actions:

- If the boot device is the target of a full copy operation or is not a source member of the existing shadow set, change the device name in VMB to a source member and reboot the node.
- If the boot device is a source member of the existing shadow set, check the booting device's connections to all other shadow set members.
- If all device and system connections are fine, check the SYSGEN parameter settings for inappropriate memory configurations.

SHADDETINCON, SHADOWING detects inconsistent state.

Facility: BUGCHECK, System Bugcheck

Explanation: The shadowing software reached an unrecoverable or inconsistent situation because shadowing software failed an internal inconsistency check.

User Action: Submit a Software Performance Report (SPR) that describes the conditions leading to the error. If the system is configured to take a memory dump, include the dump file with the SPR.

SHASINGMBR, single member system shadow set formed

Facility: INIT, INITIALIZE Command

Explanation: The shadow set membership is changing to form a single member shadow set consisting of the boot device only.

User Action: None.

SRCMEM, only source member of shadow set cannot be dismounted

Facility: DISMOUNT, DISMOUNT Command

Explanation: An attempt was made to dismount a shadow set member that was the only valid source member of the set.

User Action: If there is only one shadow set member, it cannot be dismounted. If you want to dissolve the shadow set, dismount the virtual unit. If there is more than one member, remove a full member and wait for copy operations to complete before dismounting a member.

New and Modified System Messages for VMS Version 5.4

A.1 VMS System Error Messages

virtual-unit: does not contain the member named to VMB. System may not reboot.

Facility: OPCOM, Operator Communication

Explanation:

- The boot device is dismounted or failed out of the system disk shadow set.
- Shadowing finds the boot device missing from the system disk shadow set membership during any dismount operations on the system disk.

User Action:

- Mount the boot device back into the shadow set as soon as possible.
- If you cannot mount the boot device back into the shadow set, change the device name in VMB so the system can reboot.

B

VMS DECwindows Performance Considerations

V5.3

VMS DECwindows allows the DECwindows server and applications to run on different nodes in a network. By running one or more applications on a remote node, you can minimize the amount of memory required on the workstation node. This feature can be beneficial to a workstation with a limited memory configuration. Workstations with an insufficient amount of memory exhibit response-time delays due to excessive paging.

B.1

Recommended Minimum Memory Configurations for DECwindows

Workstation memory configurations of 4 megabytes are supported in a nonclustered DECwindows environment. Digital recommends that your workstation be configured with at least 6 megabytes of memory for nonclustered use and at least 8 megabytes for use in a VAXcluster.

You are encouraged to review the memory management guidelines presented in the *Guide to VMS Performance Management*. This guide provides information about the establishment of appropriate working set quota values and other issues related to memory management.

B.2

Running VMS DECwindows Applications Remotely

If you have access to a node with enough memory to accommodate VMS DECwindows applications and DECwindows has been installed on that node, you can run your application there. An application running remotely appears identical to one running locally; the DECwindows server running on the workstation continues to handle screen output and to accept input from both the keyboard and the mouse. You need to customize the Session Manager to authorize the use of your workstation by a remote client. This procedure is described in the *VMS DECwindows User's Guide*.

When you run an application remotely, most of the memory required by the application is located on the remote node. Because more than one workstation can run the same application on a particular remote node, the application pages that are shareable can be shared by all workstations running that application. To do this, the system manager must install the application on the remote node with the **shared** attribute.

A relatively small component of an application's memory is still located on the workstation node in the form of data structures used by the DECwindows server. The number of remote applications that can be run may ultimately be limited by the amount of workstation memory available for this purpose.

When an application runs on a remote node, many of its performance characteristics reflect those of the remote node. Your application performance depends to a degree on how much memory the remote node has and on how busy the remote CPU and the network are. For example, if the remote node is a relatively fast processor, phases of the

VMS DECwindows Performance Considerations

B.2 Running VMS DECwindows Applications Remotely

application that depend heavily on the CPU, such as application startup and computation, execute faster.

A very busy CPU or network can lead to unpredictable application performance. Conversely, the performance experienced by users logged directly into the remote node depends on the amount of DECwindows work demanded of it.

Applications that have minimal communication with the workstation server generally run very well from a remote node. Applications that communicate frequently with the server, such as applications that constantly update the display in response to pointing device movements, or that transmit very large blocks of information to the server, generally do not perform as well. Local execution with sufficient local memory provides the best and most predictable performance for these types of applications.

B.3 Suggestion for Running Applications Remotely

The simplest method for running applications remotely is to bring up the FileView application remotely and then start other applications from FileView. Applications initiated this way are run on the remote node.

For example, from a local DECterm window, set host to a remote node. Once you have logged in to the remote node, run the following command procedure:

```
$ SET DISPLAY/CREATE/NODE=display-node
$ RUN SYSSYSTEM:VUE$MASTER
```

Run the previous command procedure as a noninteractive detached process, using the following command:

```
$ RUN SYSSYSTEM:LOGINOUT/DETACHED/AUTHORIZE -
_ $ INPUT=command_procedure/OUTPUT=log_file
```

When FileView is displayed, you can run other remote applications by choosing them from the Applications menu. You can continue to enter commands in the local DECterm window.

B.4 Using AUTOGEN

The single most effective tuning technique for a DECwindows workstation is to use AUTOGEN with the feedback option. AUTOGEN will automatically be run during DECwindows startup if your system's parameters cannot support DECwindows. However, this only brings the system parameters to the minimum level to run DECwindows and may not be optimal for your environment.

AUTOGEN should be run after you have operated your system for a period of time to establish your workload resource profile. This time period is usually a few days. You should invoke AUTOGEN from the system manager's account by entering the following command:

```
$ @SYSSUPDATE:AUTOGEN SAVPARAMS REBOOT CHECK_FEEDBACK
```

See the *Guide to Setting Up a VMS System* for more details about AUTOGEN.

VMS DECwindows Performance Considerations

B.5 Improving DECwindows Memory Sharing

B.5 Improving DECwindows Memory Sharing

DECwindows startup installs the DECwindows shareable image libraries to allow a single copy of the code to be shared by multiple users. If a VMS system is supporting multiple DECwindows users, additional code sharing can be achieved by installing more DECwindows images. The following images provided with DECwindows are not installed as shareable images by default:

- SYS\$SHARE:CDA\$WRITE_ANALYSIS.EXE
- SYS\$SHARE:CDA\$DTIF_TO_DDIF.EXE
- SYS\$SHARE:DDIF\$VIEWSHR.EXE
- SYS\$SHARE:DDIF\$READ_TEXT.EXE
- SYS\$SHARE:DDIF\$WRITE_PS.EXE
- SYS\$SHARE:DDIF\$WRITE_TEXT.EXE
- SYS\$SHARE:DECW\$AILSHR.EXE
- SYS\$SHARE:DECW\$MAILSHR.EXE
- SYS\$SYSTEM:CDA\$CONVERT.EXE
- SYS\$SYSTEM:DDIF\$VIEW.EXE
- SYS\$SYSTEM:DECW\$BOOKREADER.EXE
- SYS\$SYSTEM:DECW\$CALC.EXE
- SYS\$SYSTEM:DECW\$CALENDAR.EXE
- SYS\$SYSTEM:DECW\$CARDFILER.EXE
- SYS\$SYSTEM:DECW\$CLOCK.EXE
- SYS\$SYSTEM:DECW\$MAIL.EXE
- SYS\$SYSTEM:DECW\$NOTEPAD.EXE
- SYS\$SYSTEM:DECW\$PAINT.EXE
- SYS\$SYSTEM:VUE\$MASTER.EXE

Generally, each image takes up two extra pages of physical memory when installed as a shareable image. It will be necessary to increase your SYSGEN parameters for GBLPAGES, GBLSECTIONS, or both, with these additional images installed as shareable images. See the *VMS Install Utility Manual* for further information.

C

Building and Copying a VMS System Disk

This appendix contains replacement text for the section entitled “Building and Copying a VMS System Disk” in the *Guide to Setting Up a VMS System*.

C.1 Introduction

The command procedure `SYS$UPDATE:VMSKITBLD.COM` is used for building and copying a VMS system disk. The procedure provides you with the following options:

- **BUILD** — Destroys all previous information on the target disk and then creates a cluster-common system disk.
- **COPY** — Copies the operating system files to a target disk without destroying files that are currently on the target disk.
- **ADD** — Adds a new system root directory on a VMS system disk (provided the system disk you are adding to is not in use).

Caution: Do not attempt to use **VMSKITBLD** with the current system disk as the target device.

The following sections describe how to use each option.

C.2 Building the Operating System on Another Disk

You might want to move your operating system files to another disk. For example, assume that your operating system is initially stored on an RA60 disk together with some of your user files. Suppose that you have purchased an RA81 disk, and that you want to move only the operating system files from the RA60 disk to the RA81 disk. You can build the operating system on the RA81 disk (which is called the target, or destination, disk) without affecting the user files on the RA60 disk (the source disk) by using the **VMSKITBLD** command procedure with the **BUILD** option.

Caution: The **VMSKITBLD BUILD** option initializes the target disk, deleting all its previous contents.

If you want to build your operating system on another disk and you are not concerned about losing the current contents of the target disk, use the **BUILD** option as described in the following procedure:

- 1 Boot the operating system from the current system disk (source disk).
- 2 Log in to the **SYSTEM** account.
- 3 Place the target disk (assuming you are using a removable disk) in an appropriate drive and put it on line.

Building and Copying a VMS System Disk

C.2 Building the Operating System on Another Disk

- 4 Enter the following command to invoke VMSKITBLD:

```
$ @SYS$UPDATE:VMSKITBLD
```

VMSKITBLD prompts you to choose one of the following options:

```
* Operation [BUILD,ADD,COPY]?
```

- 5 Type BUILD and press the Return key.

VMSKITBLD displays messages that either prompt you for information needed to complete the operation or to inform you of the procedure's status.

The following example shows a typical message sequence for a single-node system:

```
* Enter mounted SOURCE disk name (ddcu): SYS$SYSDEVICE:
* Enter SOURCE disk top level system directory [default = SYS0]: 
* Enter TARGET disk name (ddcu): DUA0:
* Enter the TARGET disk's label [default = VAXVMSRL5]: 
* Enter TARGET disk top level system directory [default = SYS0]: 
  The target disk will be initialized.
* Target disk, _DUA0:, ready to be initialized? (Y/N): y
  Target disk, _DUA0:, has been initialized.
%MOUNT-I-MOUNTED, VAXVMSRL5 mounted on _DUA0:
  Creating system specific directories ...
  Creating cluster common directories ...
  Creating SYSGEN files ...
%SYSGEN-I-CREATED, _DUA0:<SYS0.SYSEXE>SWAPFILE.SYS;1 created
%SYSGEN-I-CREATED, _DUA0:<SYS0.SYSEXE>PAGEFILE.SYS;1 created
%SYSGEN-I-CREATED, _DUA0:<SYS0.SYSEXE>SYSDUMP.DMP;1 created
  Copying files from source disk ...
  Copying DECwindows file from source disk ...
  Writing a boot block ...
  System disk complete.
$
```

- 6 When the system disk is built, VMSKITBLD automatically dismounts the target disk. At this point, the target disk contains all the required VMS files for a complete system. However, you must still configure the system with appropriate system parameters. Use the following procedure to boot the target disk and run AUTOGEN.

Perform a **conversational boot**, as described in the upgrade and installation supplement for your computer, until the system displays the following prompt:

```
SYSBOOT>
```

- 7 When the SYSBOOT> prompt appears, enter the following commands:

```
SYSBOOT> USE DEFAULT
SYSBOOT> CONTINUE
```

- 8 After the system boots, log in to the SYSTEM account and enter the following commands to create the rights database and the network proxy database:

```
$ SET DEFAULT SYS$COMMON:[SYSEXE]
$ RUN AUTHORIZE
UAF> CREATE/RIGHTS
UAF> CREATE/PROXY
UAF> EXIT
```

Building and Copying a VMS System Disk

C.2 Building the Operating System on Another Disk

- 9 Enter the following command to run the AUTOGEN procedure and set appropriate system parameters:

```
$ @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT CHECK_FEEDBACK
```

See *Guide to Setting Up a VMS System* for detailed information on AUTOGEN.

C.3 Copying the Operating System Files to Another Disk

You can also use VMSKITBLD to copy the operating system files to a target disk without deleting the files already on it. In order to do this, the VMS operating system must be running, and the source disk that you intend to copy from must be mounted.

When you copy the system disk using VMSKITBLD.COM, the user-modified files (including SYSUAF.DAT and site-specific command files) are *not* copied from the source disk; VMSKITBLD uses the unaltered TEMPLATE versions of these files. In addition, the procedure does not create the system-specific files SWAPFILE.SYS, PAGEFILE.SYS, SYSDUMP.DMP.

Before each new system file is copied, the older version of the file is deleted on the target disk.

To copy the operating system files from the source disk to a target disk, use the following procedure:

Caution: Do not attempt to use VMSKITBLD with the current system disk as the target device.

- 1 Log in to the SYSTEM account.
- 2 Place the target disk in an appropriate drive.
- 3 Note the device name of the target disk.
- 4 Enter the following command to invoke VMSKITBLD:

```
$ @SYS$UPDATE:VMSKITBLD
```

VMSKITBLD prompts you to choose one of the following options:

```
Operation [BUILD,ADD,COPY]?
```

- 5 Type COPY and press the Return key.

VMSKITBLD displays messages that either prompt you for information needed to complete the copy operation or inform you of the procedure's status.

The following example shows a typical message sequence for a single-node system:

Building and Copying a VMS System Disk

C.3 Copying the Operating System Files to Another Disk

```
* Enter mounted SOURCE disk name (ddcu:): SYS$SYSDEVICE:
* Enter SOURCE top level system directory [default = SYS0]: 
* Enter TARGET disk name (ddcu:): DUA0:
* Enter TARGET disk top level system directory [default = SYS0]: 
%DCL-I-ALLOC, _DUA0: allocated
%MOUNT-I-MOUNTED, VAXVMSRL5 mounted on _DUA0:
  Copying files from source disk ...
  Copying DECwindows files from source disk ...
  Writing a boot block ...
  System disk complete.
$
```

- 6 When the copy operation is finished, VMSKITBLD automatically dismounts the target disk.

C.4 Adding an Alternate System Root Directory

Use the ADD option to create an alternate system root directory on a target disk. You might use this option to create a test environment. For example, if you want to test software on the operating system without interfering with the current version of the system, you could create an alternate system root directory and create a boot command procedure to select that version for testing sessions.

Note: To create a system root to add a new system to a cluster, use the **SYS\$MANAGER:CLUSTER_CONFIG.COM** procedure.

The ADD option creates only new specific root directories. The current common directory is linked to the new root.

Use the following procedure to add a system root directory to an existing system disk.

- 1 Log in to the SYSTEM account.
- 2 Check the number of free blocks on the system disk to make sure you have adequate space for the new files, including SWAPFILE.SYS, PAGEFILE.SYS, and SYSDUMP.DMP. The sizes of these files are determined by the type of computer you use.
- 3 Make sure you have a backup copy of your system disk. For instructions on how to back up your system disk, see the upgrade and installation supplement for your VAX computer.
- 4 Make sure the target system disk (your backup copy) is dismounted and offline.
- 5 Enter the following command to invoke VMSKITBLD:

```
$ @SYS$UPDATE:VMSKITBLD
```

VMSKITBLD prompts you to choose one of the following options:

```
Operation [BUILD,ADD,COPY]?
```

- 6 Type ADD and press the Return key.

You will receive messages that either prompt you for information needed to complete the operation, or inform you of the procedure's status.

Building and Copying a VMS System Disk

C.4 Adding an Alternate System Root Directory

- 7 When you are prompted for the mounted source disk name, type `SYSS$SYSDEVICE:` and press Return.
- 8 When you are prompted for the source top-level directory, press Return to use the default value.
- 9 When you are prompted for the target disk name, type the device name of the target disk, for example `DUA5:`.
- 10 When you are prompted for the target top-level directory, type the new root directory specification, for example `SYS3`.

Note: System directories `SYSE` and `SYSF` are reserved for Digital use.

A typical example using the `VMSKITBLD ADD` option might look like this:

```
* Enter mounted SOURCE disk name (ddcu:): SYSS$SYSDEVICE:
* Enter SOURCE top level system directory [default = SYS0]: Return
* Enter TARGET disk name (ddcu:): S$HEMP$DUA5:
* Enter TARGET disk top level system directory [default = SYS0]:
%DCL-I-ALLOC, _S$HEMP$DUA5: allocated
%MOUNT-I-MOUNTED, VAXVMSRL5 mounted on _S$HEMP$DUA5:
  Creating system specific directories ...
  Creating SYSGEN files ...
%SYSGEN-I-CREATED, _S$HEMP$DUA5:<SYSA.SYSEXE>SWAPFILE.SYS;1 created
%SYSGEN-I-CREATED, _S$HEMP$DUA5:<SYSA.SYSEXE>PAGEFILE.SYS;1 created
%SYSGEN-I-CREATED, _S$HEMP$DUA5:<SYSA.SYSEXE>SYSDUMP.DMP;1 created
  System disk complete.
$
```

At this point, the target system directory contains the new system root directory. However, you must still configure the new system root. Use the following procedure to boot the target disk and run `AUTOGEN`.

- 11 Shut down the system and halt your VAX computer. For instructions on shutting down your system, see the upgrade and installation supplement for your computer.
- 12 Perform a **conversational boot**, as described in the upgrade and installation supplement for your computer, until the system displays the following prompt:

```
SYSBOOT>
```

- 13 When the `SYSBOOT>` prompt appears, enter the following commands:

```
SYSBOOT> USE DEFAULT
SYSBOOT> CONTINUE
```

- 14 After the system boots, log in to the `SYSTEM` account and enter the following command to run `AUTOGEN` to configure system parameters:

```
$ @SYSS$UPDATE:AUTOGEN SAVPARAMS REBOOT CHECK_FEEDBACK
```

See the *Guide to Setting Up a VMS System* for detailed information on `AUTOGEN`.

D File Design Attributes

This appendix contains replacement text for Section 3.2.1 of the *Guide to VMS File Applications*.

D.0.1 File Design Attributes

The following file design attributes control how the file is arranged on the disk and how much of the file is transferred to main memory when needed. These file design attributes generally apply to all three types of file organization; other file design attributes that specifically pertain to the various file organizations are described under the appropriate heading.

- Initial file allocation
- Contiguity
- File extend quantity
- Units of I/O
- The use of multiple areas (for indexed files)
- Bucket fill factor (for indexed files)

The following sections discuss how each file design attribute can maximize efficiency.

D.0.1.1 Initial File Allocation

When you create a file, you should allocate enough space to store it in one contiguous section of the disk. If the file is contiguous on the disk, it requires only one retrieval pointer in the header; this reduces disk head motion.

You should also consider allocating additional space in anticipation of file growth to reduce the number of required extensions.

You can allocate space either by using the FDL attribute `FILE ALLOCATION` or by using the file access control block field `FAB$ALQ`.

D.0.1.2 Contiguity

Use the `FILE` secondary attribute `CONTIGUOUS` to arrange the file contiguously on the disk, if you have sufficient space. If you assign the `CONTIGUOUS` attribute and there is not enough contiguous space on the disk, VMS RMS does not create the file. To avoid this, consider using the FDL attribute `BEST_TRY_CONTIGUOUS` instead of the `CONTIGUOUS` attribute. The `BEST_TRY_CONTIGUOUS` attribute arranges the file contiguously on the disk if there is sufficient space or noncontiguously if the space is not available for a contiguous file.

File Design Attributes

You can make this choice by accepting the FDL default values for both attributes — NO for CONTIGUOUS, YES for BEST_TRY_CONTIGUOUS — or by taking the VMS RMS option FAB\$V_CBT in the FAB\$L_FOP field.

D.0.1.3 Extending a File

An extend operation (file extend) adds unused disk blocks to a RMS file when the free space within a file is exhausted. If the unused disk blocks are not contiguous to the previously allocated disk blocks of the file, the file becomes fragmented. As a file becomes fragmented, access time increases and processing performance can degrade. Appropriate use of extends can minimize file fragmentation.

If you intend to add relatively large amounts of data to a file over a relatively short time, using large extends will minimize file fragmentation and the overhead of extend operations. Conversely, if you intend to add relatively small amounts of data to a file over a relatively long time, smaller file extends can avoid wasted disk space.

There are two methods for doing file extends. One method is for an application program to call the \$EXTEND service. (See the *VMS Record Management Services Manual* for details.) When it calls the \$EXTEND service, the application must specify an explicit extend size in disk blocks because no defaults are used to determine the extend size.

The other method is for VMS RMS to automatically extend (auto-extend) a file when free space is needed. You can specify the size of auto-extends using various default extension quantities, or you can have VMS RMS supply a default extend size. However, when VMS RMS supplies a default, it uses an algorithm that allocates a minimal extend. Repeated minimal extends may increase file fragmentation.

D.0.1.3.1 Auto-Extend Size Selection

This section describes the factors used to determine the size of auto-extends. These include:

- RMS file organization (sequential, relative and indexed)
- Type of access (record I/O or block I/O)
- Various default extension quantities

The remainder of this section describes the usage of various default extension quantities in the selection of the auto-extend size for all VMS RMS file organizations and access types. Manipulation of the various default extension quantities is described in Section D.0.1.3.2.

Sequential File and Block I/O Accessed File Extend Size

The auto-extend size used for sequential files is used also for all VMS RMS file organizations when accessed by block I/O. The extend size is selected from the following ordered list of default extension quantities. Generally if a default extension quantity does not exist, it will be set to zero. VMS RMS processes this list until it finds a nonzero value.

- File default extension quantity

File Design Attributes

- Process default extension quantity
- System default extension quantity
- VMS RMS supplies a minimal extend size that is the smaller of twice the buffer size or 256. The buffer size in this calculation depends on the type of file access. If the file is a sequential file that is opened for record I/O access, VMS RMS uses the multiblock count. If the file is opened for block I/O access (regardless of organization), VMS RMS uses the size of the user buffer supplied by the application to the \$WRITE service.

Note that if the selected value from this list is any value but the **file** default extension quantity, the selected value is maximized against the volume default extension quantity.

Relative File Extend Size

A relative file can be viewed as an accessible series of fixed-sized cells (or records) ranging from one to the maximum number of cells. Writing new cells that are located substantially beyond the allocated space of the relative file is permitted.

- The size of a relative file auto-extend is initially set to the minimum number of disk blocks that must be allocated to reference the new cell. The extend size is then rounded up to the next bucket boundary so that the entire bucket containing the new record can be accessed.
- This value is then maximized against the file default extension quantity.
- If no file default exists, then this value is maximized against the volume default extension quantity.

The process and system default extension quantities are not applicable to auto-extending a relative file.

Indexed File Extend Size

Indexed files are auto-extended by adding space to a particular area of the indexed file. The extend size is always rounded up to a multiple of the bucket size for the area being extended.

- If the area being auto-extended had an area default extension quantity specified when the indexed file was created, or converted using an FDL, that quantity is used for the extend size.
- If no area default extension quantity exists, the file default extension quantity is used for the extend size.
- If no area or file default extension quantities are specified, VMS RMS auto-extends the area by one bucket.

The process, system, and volume default extension quantities are not applicable to auto-extending an indexed file.

File Design Attributes

D.0.1.3.2 Establishing Auto-Extend Default Quantities

This section describes how to establish the auto-extend default quantities for an RMS file.

Area and File Default Extension Quantities

You can establish a file specific default called the file default extension quantity for all RMS organizations. In the case of an indexed file with multiple areas, you can also establish a separate area default extension quantity for each area of the indexed file.

The following list describes the methods for establishing file and, where applicable, area default extension quantities:

- The recommended method is to use the FDL editor (EDIT/FDL) to permanently establish file and area default extension quantities when you create or convert a file. EDIT/FDL calculates these quantities using your responses to the script questions, and it assigns the file default extension quantity using the FILE EXTENSION attribute. For indexed files with multiple areas, EDIT/FDL assigns a default extension quantity to each area using the AREA EXTENSION attribute. A subsequent \$CREATE or CONVERT using a FDL with these EXTENSION attributes permanently sets these defaults. For a description of how EDIT/FDL calculates default extension quantities, see Appendix A.
- One alternative to using EDIT/FDL is to establish the file and area default extension quantities permanently by specifying the appropriate values in the FAB (or XABALL) used as input to the \$CREATE service.

The FAB\$W_DEQ field defines the file default extension quantity. For indexed files with multiple areas, individual area XABALLs, with the XAB\$B_AID field and the related XAB\$W_DEQ field set appropriately, establish area default extension quantities.

If you use this method, you can determine the default extension quantities using file- and area-specific information like the average record size, the number of records to be added to the file during a given period of time, the number of records per bucket, and the bucket size.

When both a FAB and an XABALL are present on the open or creation of a RMS file, the XABALL fields override equivalent FAB fields. If the XABALL is present, then the file default extension quantity is set from the XAB\$W_DEQ, overriding any value in the FAB\$W_DEQ field. In the case of an indexed file with multiple areas where multiple XABALLs may exist, the file default extension quantity is set to the default extension quantity for Area 0.

A single allocation XAB (XABALL) can also be specified on the creation of a relative or sequential file. However, there is no separate area default extension quantity maintained for these files. The XABALL is used in this case to establish the file default extension quantity.

File Design Attributes

- After a file has been created, specifying the file default extension quantities (FAB\$W_DEQ) on input to a \$OPEN establishes a temporary file default extension quantity that overrides any permanent setting that may exist. This temporary default is used when you access the file until the file is closed.

Note that the area default extension quantities for an indexed file specified on a \$CREATE cannot be changed over the lifetime of the file nor can they be overridden at run time.

- Once a file has been created, you can change or establish the permanent file default extension quantity by using the DCL command SET FILE/EXTENSION=*n*, where *n* is the extension quantity in disk blocks for the file. The next open of the file uses the new default quantity.

In addition to the file and area default extension quantities, there are process, system, and volume default extension quantities.

Process Default Extension Quantity

The process default extension quantity is established by the DCL command SET RMS_DEFAULT/EXTEND_QUANTITY=*n*, where *n* is the extension quantity. This default applies only to the process issuing this DCL command and remains in effect only until the process is deleted.

System Default Extension Quantity

The system default extension quantity is established by the SET RMS_DEFAULT/SYSTEM/EXTEND_QUANTITY=*n* command. Note that you need the CMKRNL privilege to use the /SYSTEM qualifier. This default applies to all processes on a node in the cluster. When you use this DCL command to establish the system default extension quantity, it remains in effect until the node is rebooted.

You can also establish the system default extension quantity in a temporary or permanent fashion by appropriately setting the SYSGEN system parameter RMS_EXTEND_SIZE.

Volume Default Extension Quantity

The volume default extension quantity can be permanently established when the volume is initialized with the INITIALIZE/EXTENSION=*n* command. This default quantity is used whenever the volume is mounted. To permanently change the volume default extension quantity, you can issue the SET VOLUME/EXTENSION=*n* on a mounted disk. To temporarily establish a volume default extension quantity or temporarily override the permanent volume default extension quantity, use the MOUNT/EXTENSION=*n* command. The new default is in effect until the disk volume is dismounted. Unlike the other default quantities described that default to 0 if not specified, the volume default extension quantity defaults to 5 if not specified.

File Design Attributes

D.0.1.3.3 Placement and Contiguity of Extends

In addition to specifying the size of an extend, you can specify other characteristics that affect the placement and contiguity of the extend.

When an application extends a file by calling the \$EXTEND service, an Allocation XAB (XABALL) can be used to place an extend on a particular disk block or disk cylinder. If no allocation XAB is present on the \$EXTEND and the FAB contiguity options (described later in this section) are not selected, VMS RMS automatically places the extend near the last allocated disk block in the file. If the file being extended in this fashion is an indexed file opened for record I/O access, VMS RMS adds the new disk space as near as possible to the last allocated disk block in the area being extended. This technique groups disk blocks belonging to the same area of the indexed file.

When VMS RMS automatically extends a file, the application cannot control placement. However, VMS RMS uses placement controls appropriate to the file organization:

- When automatically extending an indexed file, VMS RMS uses placement control to allocate the new disk space as close as possible to the last allocated disk block of the indexed file area being extended.
- When automatically extending a relative file, VMS RMS uses placement control to allocate the new disk space as close as possible to the last allocated disk block of the file.
- No placement control is used when VMS RMS automatically extends a sequential file or any VMS RMS file organization accessed for block I/O.

An extend is considered contiguous if all the disk blocks of the extend are physically adjacent on the disk. Two types of contiguous extend requests can be made. The first, called a **contiguous request**, returns an error if contiguous disk blocks cannot be found to satisfy the request. The second, called a **contiguous best try request**, attempts to find contiguous disk blocks for the request. If it does not find sufficient contiguous space, it extends the file and does not return an error. The contiguity options can be input to an \$EXTEND service in the FAB (FAB\$V_CBT, FAB\$V_CTG) or in the Allocation XAB (XAB\$V_CBT, XAB\$V_CTG). The Allocation XAB settings override any FAB settings.

When RMS automatically extends a file, the application can only indirectly control contiguity by setting the FAB or XABALL contiguity bits on input to the \$CREATE service. Once set on file creation, these options are available for subsequent extends done automatically by RMS.

Note that setting the FAB\$V_CTG bit could cause an extend to fail on a sufficiently fragmented disk. Note too, that the FAB\$V_CBT option is disabled after several failures to allocate contiguous disk space, to avoid the expensive overhead of contiguous-best-try processing on a badly fragmented disk.

D.0.1.4 Truncating a File

Only RMS sequential disk files that have been opened for write access (FAB\$V_PUT, FAB\$V_UPD, FAB\$V_DEL or FAB\$V_TRN) can be truncated. This applies to unshared and shared sequential files.

Two types of truncation can occur on RMS sequential files: RMS truncation and ACP truncation.

RMS truncation involves resetting the End-of-File (EOF) pointer back to a previous position (possibly the beginning) of a sequential file, to reuse the allocated space in a file. RMS truncation is described in the *VMS Record Management Services Manual* under the \$TRUNCATE service.

ACP truncation occurs when VMS RMS closes a sequential file and requests that the ACP deallocate all disk blocks allocated beyond the EOF. The primary use of ACP truncation is to conserve disk space. The remainder of this section deals with ACP truncation.

You can also use ACP truncation in conjunction with large extend sizes to reduce disk fragmentation. If a file is growing slowly over time, the application can allocate the largest possible extend, and, when finished, it can use ACP truncation to deallocate any unused space at the end of the sequential file. However, if a sequential file is continually growing, excessive ACP truncation can lead to an increase in disk fragmentation, resulting in more CPU and I/O overhead.

ACP truncation can be requested directly by way of the VMS RMS programming interface by setting the FAB\$V_TEF bit on input to the \$OPEN, \$CREATE, or \$CLOSE service. The ACP truncation occurs on the close of the sequential file. Note that ACP truncation can occur on shared as well as unshared sequential files. If there are shared readers of the file, ACP truncation is postponed until the last reader of the file closes the file. If there are other writers of a shared sequential file, then ACP truncation requests are ignored. However, the ACP truncation request of the last writer to close the file will be honored.

ACP truncation of a sequential file may be automatically requested by RMS if an auto-extend has been done during this file access and no file default extend quantity exists to be used for the auto-extend. This avoids wasting space when auto-extending with a less precise extend quantity default, like the system default extend quantity.

D.0.1.5 Units of I/O

Another file design consideration is to reduce the number of times that VMS RMS must transfer data from disk to memory by making the I/O units as large as possible. Each time VMS RMS fetches data from the disk, it stores the data in an I/O memory buffer whose capacity is equal to the size of one I/O unit. A larger I/O unit makes more records immediately accessible to your program from the I/O buffers.

In general, using larger units of I/O for disk transfers improves performance, as long as the data does not have to be swapped out too frequently. However, as the total space used for I/O buffers in the system approaches a point that results in excessive paging and swapping, increasing I/O unit size degrades system performance.

File Design Attributes

VMS RMS performs I/O operations using one of the following I/O unit types:

- Blocks
- Multiblocks
- Buckets

A *block* is the basic unit of disk I/O. It consists of 512 contiguous bytes. Even if your program requests less than a block of data, VMS RMS automatically transfers an entire block.

The other I/O units—multiblocks and buckets—are made up of block multiples. A *multiblock* is an I/O unit that includes up to 127 blocks but whose use is restricted to sequential files. A *bucket* is the I/O unit for relative and indexed files. It may include up to 63 blocks.

D.0.1.6 Multiple Areas for Indexed Files

For indexed files, another design strategy is to separate the file into multiple *areas*. Each area has its own extension size, initial allocation size, contiguity options, and bucket size. You can minimize access times by precisely positioning each area on a specific disk volume, cylinder, or block.

For instance, you can place the data area on one volume of a volume set and place the indexed area on another volume of the volume set. If your application is I/O bound, this strategy could increase its throughput. You can also ensure that your data buckets are contiguous for sequential access by allocating extra space to the data area for future extensions.

D.0.1.7 Bucket Fill Factor for Indexed Files

Any attempt to insert a record into a filled bucket results in a *bucket split*. When a bucket split occurs, VMS RMS tries to keep half of the records (including the new record if applicable) in the original bucket and moves the remaining records to a newly created bucket.

Excessive bucket splitting can degrade system performance through wasted space, excessive processing overhead, and file fragmentation. For example, each record that moves to a new bucket must maintain a forward pointer in the original bucket. The forward pointer indicates the record's new location. At the new bucket, the record must maintain a backward pointer to its original bucket. VMS RMS uses the backward pointer to update the forward pointer in the original bucket if the record later moves to yet another bucket.

When a program attempts to access a record by alternate key or by RFA, it must first go to the bucket where the record originally resided, read the pointer to the record's current bucket residence, and then access the record.

To avoid bucket splits, you should permit buckets to be only partially filled when records are initially loaded. This provides your application with space to make additional random inserts without overfilling the affected bucket.

Index

A

ACL (access control list) Editor
 \$CHANGE_ACL lock correction • 4-1
 protected entries correction • 4-2

Address
 NML address check • 3-23

ADD_ prefix
 subtracting parameter values in MOD-
 PARAMS.DAT • 3-5

Alias directory entries • 2-29

Alternate root directory
 adding to an existing system disk • C-4

ANALDISK facility code (ANALYZE/DISK_
 STRUCTURE) • 3-2

Analyze/Disk_Structure Utility (ANALYZE/DISK_
 STRUCTURE)
 ANALDISK facility code change • 3-2

Asynchronous option
 VMS RMS support • 5-16

Audit Analysis Utility (ANALYZE/AUDIT)
 /SELECT=STATUS=FAILURE qualifier problem •
 3-3

Authorize Utility (AUTHORIZE)
 adding proxy accounts • 3-3
 ADD/PROXY and MODIFY/PROXY commands •
 3-3
 /NOEXPIRATION qualifier • 5-8
 restricted flag modifications • 3-3

AUTOGEN
 dump file size • 3-37
 end phase TESTFILES • 5-1
 FEEDBACK generated parameters
 ADD_ prefix • 3-5
 feedback mechanism • 3-4
 files not marked NOBACKUP • 3-6
 MSCP server mechanism • 3-5
 running DECW\$STARTUP.COM command • 3-26
 selective crash dump files • 3-38
 start phase GETDATA • 5-2
 swapping file size • 3-6
 switching window systems • 3-6
 volume shadowing adjustment required • 3-86

B

BACKUP
 saving a bound volume set • 3-11

BACKUP command
 with the TA90E tape drive • 3-75

Backup Utility (BACKUP)
 BACKUP command
 /DELETE and /RECORD qualifiers • 3-10
 corrected problems • 3-8
 files with active transactions • 3-7
 image save operation restriction • 3-9
 postprocessing problem • 3-10
 problems and restrictions • 3-8
 using with compound document files • 3-11

Batch/Print Facility • 2-1

Bootstrap procedures for XDELTA
 See entries for specific computers

/BUFFER_COUNT qualifier • 3-73

C

Calendar
 See DECwindows applications

CALENDAR.SPLIT improvement • 4-49

\$CANCEL system service • 3-64

Captive command procedures
 effect of Ctrl/Y • 3-65

CAPTIVE flag (UAF) • 3-60
 new interpretation • 3-59

CDA Toolkit
 base converters • 4-13, 4-14
 corrections and new functionality • 4-14, 4-16
 new messages to clarify errors • 4-16

CDA Viewer
 See DECwindows applications

Clock
 See DECwindows applications

CLOSE procedures (VAX Ada) • 4-49

Clusters
 adding proxy accounts • 3-3

CLUSTER_CONFIG.COM command procedure •
 3-15

VOLPRO privilege • 3-14

Index

Command procedures
 labels • 2-2
Command verb and qualifier length • 2-22
CTDRIVER
 effect of \$CANCEL on • 3-64
 effect of Ctrl/C • 3-64, 3-65
 effect of Ctrl/Y • 3-64, 3-65
 effect of out-of-band abort character • 3-64, 3-65
 enforces SETMODE/SENSEMODE buffer size • 3-64
 loading • 3-67
 output buffering • 3-64
CTERM protocol
 limitations on use of Ctrl/C • 3-67
Ctrl/C
 effect when using CTDRIVER • 3-64
 limitation of CTERM protocol • 3-67
Ctrl/Y
 effect in captive command procedure • 3-65
 effect when using CTDRIVER • 3-64
Cursor patterns • 4-19
Cursor screen boundaries • 4-19

D

DAP (DECnet file access protocol)
 extensions • 4-13
Data record compaction
 TA90E support • 3-75
DCL (DIGITAL Command Language)
 ANALYZE/ERROR_LOG command • 2-22
 BACKUP/REWIND command • 3-10
 CALL command • 4-2
 command verb and qualifier length • 2-22
 Debugger commands • 4-8
 DEFINE/FORM command • 2-23
 expiration of RMS disk files • 4-40
 F\$CONTEXT lexical function • 2-23
 IF-THEN-ELSE construct • 4-33
 label scoping in • 4-2
 MOUNT command
 OPCOM message • 2-24
 OPEN command • 2-26
 PRINT command • 2-26
 /PAGES qualifier • 2-23
 REPLY/LOG command • 3-50
 SET DISPLAY command • 4-22
 SET HOST/DTE command • 2-28, 4-48
 SHOW DEVICES/FULL command • 2-28
 SHOW LOGICAL/FULL command • 2-28
 DCL (DIGITAL Command Language) (Cont.)
 SHOW MAGTAPE command • 2-28
 SUBMIT command • 2-26
 SUBMIT/DELETE command • 2-29
 SUBROUTINE command • 4-2
 subroutine entry points • 4-2
 substring assignment • 4-3
 DEBNA Ethernet controller
 and tuning VMS operating system • 3-41
 DEBNI Ethernet/802 controller • 3-39
 Debugger
 commands disabled in DECwindows • 4-6
 commands used in DCL command procedures • 4-8
 corrected problems and restrictions • 4-4
 examining LABEL[*n*] • 4-6
 obsolete commands • 4-11
 problems with DECwindows interface • 4-9
 using ABORT key after SPAWN command • 4-8
 using concealed rooted-directory logical names • 4-8
 using on VAXstation workstation • 4-9
 using Stop button after SPAWN command • 4-8
 Debugging with POOLCHECK • 4-30
 DEC\$SYLOGIN.COM file • 3-27
 DEC\$SYLOGIN.TEMPLATE file • 3-27
 DECburger sample application
 corrections • 5-9
 DECdfs (DIGITAL Distributed File Service)
 required update • 3-78
 DECdqs (DIGITAL Distributed Queuing Service) • 3-28
 DECdtm services
 adjusting SYSGEN parameters for • 3-18
 defining logical names before starting • 3-18
 how to inhibit • 3-18
 image files for • 3-19
 start-up processes • 3-18
 DECnet file access protocol
 See DAP
 DECnet software
 default account • 3-54
 downline loading • 3-21
 executor
 ALIAS MAXIMUM LINKS parameter • 3-22
 MAXIMUM LINKS parameter • 3-22
 PIPELINE QUOTA parameter • 3-22
 incompatibility among Phase IV implementations • 3-24
 starting during DECwindows session • 2-8
 DECnet-VAX objects
 characteristic added • 3-20

DECnet-VAX objects (Cont.)

- outgoing connect privileges • 3-20
- preventing access to • 3-20

DECterm

See DECwindows

DECW\$IGNORE_DECWINDOWS logical name • 3-26

DECwindows • 2-3

applications

- Calendar restrictions • 2-8
- CDA Viewer restrictions • 2-11
- Mail • 2-11, 2-12
- performance • B-1
- running remotely • B-1

Debugger problems • 4-9

DECterm

- conformance level • 2-3
- corrected color table report problem • 4-17
- fonts • 4-18
- graphics • 2-3
- initializing • 2-4
- keyboards • 2-5
- languages • 2-5
- memory • 2-6
- negative values correction • 4-18
- ReGIS locator report • 4-18
- text • 2-7
- VT52-mode cursor addressing • 4-18
- window scrolling problem • 2-7

DECW\$COLOR guidelines • 2-7

DECW\$STARTUP.COM procedure
running AUTOGEN.COM command
procedure • 3-26

disabled Debugger commands • 4-6

FileView • 2-14

process quotas • 2-17

font properties • 5-13

limit on number of clients • 4-19

minimum memory • B-1

multihead system support • 3-27

Print Screen function • 2-14

server • 4-18

Session Manager • 2-19

startup • 3-26

startup problem • 2-8

tailoring on DR53 system disk • 3-26

terminal emulator

See DECterm

terminal fonts • 4-18

UIL built-in tables • 4-22

UIL corrections • 4-23

DECwindows (Cont.)

ULTRIX authorization requirements • 2-21

VT1000 terminal support • 2-21

Window Manager icon box • 2-22

DEFINE L!NE command • 5-15

Delta/XDelta Utility (DELTA/XDELTA)

bootstrapping the VAXft 3000 system • 3-83

invoking XDELTA • 4-60

DEMNA Ethernet/802 controller • 3-40

Department of Defense (DoD) erase pattern • 3-53

DEQNA Ethernet adapter

reducing corrupt data • 3-40

DEQTA Ethernet/802 controller • 3-40

Device driver debugging

with POOLCHECK parameter • 4-30

DEVICE_SCAN • 4-33

DIGITAL Distributed File Service

See DECdfs

DIGITAL Distributed Queuing Service

See DECdqs

DIGITAL Standard Runoff

See DSR

Directory processing

size limitations removed • 4-31

Disk

See also RA92 DSA disk

DSA drivers, alternate host information • 4-31

large-capacity disks, header space problem • 3-29

DISMOUNT command

processing open files • 3-31

DO command (SYSMAN Utility) • 3-72

DRM routines

unavailable VAX bindings for • 4-26

DSDRIVER disk class driver • 3-37

accounting size of lock ID • 3-35

DSR (DIGITAL Standard Runoff)

/VARIANT qualifier • 2-29

DSSI (DIGITAL Storage System Interconnect)

device naming • 3-35

DUDRIVER disk class driver • 3-37

E

EDT editor

delete access requirement • 2-30

Erase pattern

Department of Defense • 3-53

Error logging

new format • 2-22

Index

Error messages • 5–41

Ethernet

DEBNI controller • 3–39

DEMNA controller • 3–40

DEQTA controller • 3–40

SGEC controller • 3–42

/EXTEND_QUANTITY qualifier • 3–73

F

F\$GETDVI lexical function

VAXft 3000 device information • 3–83

FAB\$V_ASY option

documentation change • 5–16

File design attributes • D–1

FileView

See DECwindows

Flag

restricted modifications in Authorize Utility • 3–3

Folder name parameters

in Mail Utility • 2–35

G

GBBDriver output driver • 4–32

GETDATA

corrected AUTOGEN start phase • 5–2

H

Hello World! sample application

corrections • 5–9

Help widget • 2–18

I

I/O device drivers

ACP–QIO interface • 4–33

opening a sequential-media file • 4–32

I/O driver

logical end-of-volume detection • 4–32

Icon box • 2–22

IF-THEN-ELSE construct (DCL)

setting \$STATUS symbol • 4–33

INCN_TIME field (SHOW CLUSTER) • 3–69

Internal processor register

See IPR

Interrupt request for XDELTA • 4–68

See also entries for specific processors

INVALIDATE_TB macro • 4–30

IPCACP process (DECdtm), preventing startup of • 3–18

IPR (internal processor register)

definition symbols • 4–39

J

Job queue manager

/BUFFER_COUNT qualifier • 3–12

Joe queue manager

starting • 3–73

L

Label scoping in DCL • 4–2

Language

reinstalling • 4–33

Layered products

upgrade caution • 3–77

VAX Public Access Communications • 3–78

VAX TU70/72 Device Driver • 3–79

LIB\$DECODE_FAULT

use with vector processor • 4–44

LIB\$FREE_VM • 4–45

LIB\$GET_VM • 4–45

LIB\$SHOW_VM_ZONE • 4–45

LIB\$VERIFY_VM_ZONE • 4–45

LICENSE (License Management Utility)

activity definition • 1–16

codes for license types • 1–11

database location • 1–15

error messages • 1–3

example of registration • 1–2

for service customers • 1–8

license registration procedure • 1–1

LMF\$CONFIG.COM procedure • 1–8

modifying license units • 1–10

MOD_UNITS option • 1–10

types for VMS • 1–15

VMSLICENSE.COM procedure • 1–3

License Management Facility
 See LMF

License Management Utility
 See LICENSE

License Unit Requirement Table (LURT) • 1–11

Linker image-ID field • 5–14

Linking image
 against system table of different VMS version •
 4–2

LMF (License Management Facility)
 DECnet–VAX software • 1–17
 VAXcluster software • 1–16
 VAX RMS Journaling software • 1–17
 VAX Volume Shadowing software • 1–17

LNK\$OPEN_LIB logical name • 4–34

Lock ID
 accounting maximum size • 3–35

M

MA780 (multiport shared memory) • 3–43

Magnetic tape ACP
 correction to I/O • 3–43

Mail Utility (MAIL)
 folder name parameter • 2–35
 PRINT/QUEUE command changes • 3–44

Maintenance operations module (MOM)
 repeated operations • 3–22

Mass Storage Control Protocol
 See MSCP

Mass storage devices
 DSSI device naming change • 3–35

Memory
 minimum recommended for DECwindows • B–1

Message Router
 installation restriction • 4–34

MicroVAX/VAXstation 3600-series computer
 bootstrap procedure for XDELTA • 4–62
 requesting interrupt • 4–69

MicroVAX 2000 computer
 bootstrap procedure for XDELTA • 4–62
 requesting interrupt • 4–69

MicroVAX 3400-series computer
 bootstrap procedure for XDELTA • 4–62
 requesting interrupt • 4–69

MicroVAX 3900-series computer
 bootstrap procedure for XDELTA • 4–62
 requesting interrupt • 4–69

MicroVAX II computer

MicroVAX II computer (Cont.)
 bootstrap procedure for XDELTA • 4–62
 requesting interrupt • 4–69

Modified page list
 flushing • 3–44

Modified page writer • 3–44

MODPARAMS.DAT • 3–5

MODPARAMS.DAT file
 ADD_ prefix • 3–5

Monitor Utility (MONITOR) • 5–14
 MONITOR CLUSTER command • 3–48

MSCP (Mass Storage Control Protocol) • 3–44

Multiport shared memory
 See MA780

N

National Character Set Utility (NCS)
 Norwegian collating sequence change • 4–38

NCP (Network Control Program) • 3–23
 DEFINE LINE command • 5–15
 number of receive buffers • 5–15
 Remote Buffer Errors counter • 5–15
 SET/DEFINE CIRCUIT command • 3–22
 SET/DEFINE EXECUTOR command • 3–21
 SET/DEFINE OBJECT command • 3–20
 SET LINE command • 5–15
 SHOW CIRCUIT command • 5–15
 SHOW LINE command • 5–15

NETACP\$BUFFER_LIMIT logical name • 3–23

NETACP (network ancillary control process) • 3–23

NETCONFIG.COM command procedure
 security enhancements • 3–54

NETCONFIG_UPDATE.COM command procedure •
 3–54

Network ancillary control process
 See NETACP

Network Control Program
 See NCP

Network default access
 controlling access to your system • 3–54

Network management listener
 See NML

NEXT_AGGREGATE routine • 4–17

NML (network management listener)
 check for illegal address • 3–23
 object • 3–23

/NOEXPIRATION qualifier (AUTHORIZE) • 5–8

NOP instruction (VAX MACRO) • 4–55

Index

NUMERIC_ERROR exception (VAX Ada) • 4-49

O

ODS-1 disk structures
 backup corrections • 3-10
OPC\$LOGFILE_CLASSES logical name • 3-51
OPC\$LOGFILE_ENABLE logical name • 3-51
OPC\$LOGFILE_NAME logical name • 3-51
OPC\$OPA0_CLASSES logical name • 3-51
OPC\$OPA0_ENABLE logical name • 3-50
OPCOM (Operator Communication Manager)
 enabling OPA0: • 3-50
 log file operator classes • 3-50
 number of messages sent by • 3-49
 operator log file • 3-49
 operator request numbers • 3-49
 overriding defaults • 3-49, 3-50
 remove old overrides • 3-52
 SECURITY class messages • 3-56
OPCOM message
 with MOUNT command • 2-24
Open image library
 support for AIA in Linker • 4-34
Operating system
 copying files to another disk • C-3
Operator Communication Manager
 See OPCOM
Out-of-band abort characters
 effect when using CTDRIVER • 3-64

P

Page table entry
 modifying • 4-30
PARAMETERS SET/STARTUP command (SYSMAN
 Utility) • 3-73
Password
 service • 3-23
Passwords
 new security alarms • 3-55
Path split policy
 and INTERIM option • 3-21
 and NORMAL option • 3-21
PGFIPLHI crash • 3-72
PIOPAGES parameter • 4-43
POOLCHECK parameter
 enhancements • 4-30

PPL\$CREATE_APPLICATION • 4-46
PPL\$FIND_OBJECT_ID • 4-46
PPL\$FIND_SYNCH_ELEMENT_ID • 4-46
PPL\$INITIALIZE • 4-46
PPL\$TRIGGER_EVENT
 memory correction • 4-46
Primary directory entries • 2-29
Printer execution queue • 3-52
Printers
 LJ250 • 2-14
Print Screen function (DECwindows) • 2-14
Print symbiont
 purging working set • 3-11
PRIO=HIGH parameter (SPI\$MAP_BUFFER macro) •
 4-30
PRIORITY_OFFSET parameter (SYSGEN) • 3-71
Process-permanent files
 VMS RMS asynchronous support • 5-16
PROCESS_SCAN • 4-33
Proxy accounts
 changes • 3-3
Pseudoterminal driver • 3-52

Q

Q-bus
 DEQTA Ethernet/802 controller support for • 3-40
\$QIO call requirements
 setting multiscreen cursor pattern • 4-19
Queue
 execution • 3-11
 generic • 3-11
Queue file
 fragmentation • 3-74
Queue manager
 See Job queue manager

R

R3 intrinsics
 problem corrected • 4-26
RA92 DSA disk
 defining symbol DT\$_RA92 • 4-39
 support • 2-32
RAB\$V_ASY qualifier
 process-permanent file support • 4-40
Record Management Services
 See VMS RMS

- Reinstalling languages • 4–33
 - REM\$MAX_TERMINALS • 3–68
 - REMACP
 - loading remote drivers • 3–67
 - setting maximum remote-user value dynamically • 3–68
 - Remote buffer errors • 5–15
 - REMOVE_AGGREGATE routine • 4–17
 - RESTRICTED flag (UAF)
 - new flag • 3–59
 - Rights database • 3–1
 - RJOBLIM
 - setting dynamically • 3–68
 - RMS journaling
 - recovery-unit journaling • 4–42
 - RMS statistics
 - restrictions • 4–41
 - Root directory
 - adding to an existing system disk • C–4
 - RSX–11 compatibility mode
 - limitation on directory size • 4–31
 - RTL (Run-Time Library)
 - language support • 4–46
 - LIB\$CREATE_VM_ZONE routine, new flags added • 4–44
 - LIB\$FREE_VM routine • 4–45
 - LIB\$GET_VM routine • 4–45
 - LIB\$SHOW_VM_ZONE routine • 4–45
 - LIB\$SYS_TRNLOG routine • 5–18
 - LIB\$VERIFY_VM_ZONE routine • 4–45
 - SYSS\$SHARE:UVMTHRTL.EXE routine • 4–47
 - RTPAD
 - work-around for CTERM problem with Ctrl/C • 3–67
 - RTTDIVER • 3–67
 - RTTLOAD.COM command procedure • 3–67, 3–68
 - Run-Time Library
 - See RTL
 - RX33 diskette
 - VMS kits • 3–79
-
- S
-
- Save sets backup
 - reading from TU81–PLUS tape drive • 3–10
 - SCSI_NOAUTO parameter (SYSGEN) • 3–71
 - SDA
 - procedure to cause a VAXft 3000 system failure • 3–83
 - Second-Generation Ethernet Controller (SGEC) • 3–42
 - Security
 - auditing failure mode setting • 3–56
 - new alarms for passwords • 3–55
 - Security enhancements to NETCONFIG.COM
 - for new systems • 3–54
 - Service passwords • 3–23
 - Session Manager
 - See DECwindows
 - Set cursor pattern \$QIO call • 4–19
 - SET FILE/AI_JOURNAL command
 - errors when creating duplicate journals • 4–43
 - SET FILE/BI_JOURNAL command
 - errors when creating duplicate journals • 4–43
 - SET HOST facility • 3–64
 - dynamic failover • 3–67
 - extra read prompt • 3–66
 - SET LINE command • 5–15
 - SETMODE/SENSEMODE buffer size
 - enforced by CTDRIVER • 3–64
 - Shared memory
 - error messages • 5–41
 - SHOW CIRCUIT command • 5–15
 - Show Cluster Utility (SHOW CLUSTER)
 - INCN_TIME field obsolete • 3–69
 - SHOW LINE command • 5–15
 - SHUTDOWN.COM command procedure
 - change in disk dismount reporting • 3–69
 - SMP
 - See Symmetric Multiprocessing
 - Socket routines • 4–54
 - Sort/Merge Utility (SORT) • 2–32
 - SPI\$MAP_BUFFER macro
 - new parameter PRIO=HIGH • 4–30
 - Standalone BACKUP
 - command to boot from an RL02 disk • 5–42
 - STARLET library symbols • 3–60
 - STARTUP.COM procedure
 - new sequence of operations • 3–69
 - \$STATUS symbol
 - set by IF-THEN-ELSE construct • 4–33
 - Subroutine entry points in DCL • 4–2
 - Substring assignments • 4–3
 - Switched virtual circuit
 - local address implemented for • 3–22
 - SYLOGICALS.COM command procedure • 3–49, 3–50
 - Symbol names
 - making assignments • 2–29
 - Symmetric Multiprocessing (SMP) • 3–71

Index

- Synchronous device
 - repeated operations on • 3-22
 - SYS\$CHANGE_ACL • 5-23
 - lock correction • 4-1
 - SYS\$CHKPRO • 5-24
 - SYS\$CMKRNL • 4-1, 5-25
 - SYS\$CREATE_RDB
 - creation of rights database • 3-1
 - SYS\$CREMBX • 5-25
 - error message • 5-25
 - SYS\$CREPRC • 5-26
 - SYS\$CRMPSC • 5-26, 5-27
 - SYS\$DCLCHM • 5-27
 - SYS\$DEQ • 5-27
 - SYS\$ENQ • 5-29
 - SYS\$ERAPAT • 3-53
 - SYS\$FAO • 5-29
 - SYS\$FORMAT_ACL • 5-30
 - SYS\$GETDVI • 5-30
 - using to obtain FREEBLOCK count • 3-87
 - VAXft 3000 device information • 3-83
 - SYS\$GETQUI • 5-30
 - SYS\$GETSYI • 5-31, 5-32
 - SYS\$GETUAI • 5-35
 - SYS\$MOUNT • 5-35
 - SYS\$PUTMSG • 5-36
 - SYS\$QIO • 5-38
 - SYS\$SETDDIR • 5-38
 - SYS\$SETEXV • 5-38
 - SYS\$SETPRV • 5-39
 - SYS\$SETSFM • 5-39
 - SYS\$SETSSF • 5-39
 - SYS\$SETUAI • 5-39
 - SYS\$SHARE:UVMTHRTL.EXE routine • 4-47
 - SYS\$SNDJBC • 5-39, 5-40
 - SYS\$SNDOPR • 5-40
 - SYS\$UNWIND • 5-40
 - SYS\$UPDESC • 5-40
 - SYS\$WFLOR • 5-41
 - SYSGEN (System Generation Utility)
 - adjusting parameters for DECdtm services • 3-18
 - new UCB order • 4-31
 - parameters
 - correction to SCSI_NOAUTO description • 5-43
 - PIOPAGES parameter • 4-43
 - PQL_MPRCLM parameter • 2-20
 - PRIORITY_OFFSET parameter • 3-71
 - RJOBLIM parameter • 3-68
 - SCSI_NOAUTO parameter • 3-71
 - SYSLOA • 3-72
 - SYSMAN (System Management Utility)
 - changes to DO command • 3-72
 - PARAMETERS SET/STARTUP command
 - corrections • 3-73
 - SET PROFILE command • 3-73
 - SYSSTARTUP_V5.COM command procedure
 - removing REPLY commands from • 3-52
 - System disk
 - building and copying • C-1
 - System disk size, recommendation • 3-74
 - System Dump Analyzer Utility
 - See SDA
 - System dump file
 - size • 3-74
 - System Generation Utility
 - See SYSGEN
 - System Management Utility
 - See SYSMAN
 - System messages
 - for VAXft 3000 system • 2-34
 - new and modified for VMS Version 5.4 • A-1
 - System services • 5-22 to 5-41
 - See also specific services
 - arguments • 5-4
 - \$CANCEL • 3-64
 - self-modifying item lists • 5-41
 - System symbol table
 - linking against • 4-2
 - System User Authorization File
 - See UAF
 - SYSUAF (System User Authorization File)
 - See UAF
-
- ## T
-
- TA90E tape drive • 2-32
 - support for • 3-75
 - using BACKUP command • 3-76
 - using /MEDIA_FORMAT qualifier • 3-75
 - using SHOW DEVICE command • 3-75
 - Tape drive
 - TLZ04 • 4-48
 - Tape support, for ANSI initialized magnetic tapes • 2-33
 - TAPE_ALLOCLS parameter • 3-72
 - TDRIVER.MAR file • 3-76
 - Terminal driver
 - asynchronous \$CANCEL • 3-64

TESTFILES

- corrected AUTOGEN end phase • 5-1
- TIMEDWAIT macro (VAX MACRO) • 4-56
- TLZ04 tape drive
 - defining symbol DT\$_TLZ04 • 4-48
 - performance • 2-33
 - support • 2-33
- TP_SERVER process (DECdtm), preventing startup of • 3-18
- Translation buffer
 - invalidating • 4-30
- TU58 console bootstrap procedures • 4-61
- TU81-PLUS tape drive
 - reading backup save sets from • 3-10
- Tuning VMS operating system
 - for DEBNA controllers • 3-41

U

- UAF (User Authorization File)
 - flags • 3-59
- UAI\$V_CAPTIVE symbol (STARLET) • 3-60
- UAI\$V_RESTRICTED symbol (STARLET) • 3-60
- UIL compiler • 4-22
 - convenience translation files • 4-23
 - valid tables changes • 4-22
- ULTRIX applications • 2-21
- Upgrading VMS systems
 - VAX Public Access Communications requirements • 3-78
 - VAXstation 8000 unsupported • 3-79
 - VAX TU70/72 Device Driver requirements • 3-79
 - VAX Workstation Software (VWS) requirements • 3-79
- User Authorization File
 - See UAF
- User EOT mode, correction • 4-33
- User Interface Language Compiler
 - See UIL compiler

V

- VAX-11/730 computer
 - bootstrap procedure for XDELTA • 4-61
 - requesting interrupt • 4-68
- VAX-11/750 computer
 - bootstrap procedure for XDELTA • 4-62

- VAX-11/750 computer (Cont.)
 - bootstrap procedure for XDELTA with TU58 console • 4-62
 - requesting interrupt • 4-68
- VAX-11/780 computer
 - bootstrap procedure for XDELTA • 4-63
 - requesting interrupt • 4-68
- VAX-11/785 computer
 - bootstrap procedure for XDELTA • 4-63
 - requesting interrupt • 4-68
- VAX 6000 computer
 - bootstrap procedure for XDELTA • 4-63
 - requesting interrupt • 4-68
- VAXstation 3100-series computer
 - bootstrap procedure for XDELTA • 4-68
- MicroVAX 3100-series computer
 - bootstrap procedure for XDELTA • 4-68
- VAX 4000 Model 300 computer
 - bootstrap procedure for XDELTA • 4-62
 - requesting interrupt • 4-69
- VAX 8000-series systems
 - SET TIME/CLUSTER command • 3-80
 - SET TIME command • 3-80
 - VAXBI restriction • 3-81
- VAX 8200 computer
 - bootstrap procedure for XDELTA • 4-64
 - requesting interrupt • 4-68
- VAX 8250 computer
 - bootstrap procedure for XDELTA • 4-64
 - requesting interrupt • 4-68
- VAX 8300 computer
 - bootstrap procedure for XDELTA • 4-64
 - requesting interrupt • 4-68
- VAX 8350 computer
 - bootstrap procedure for XDELTA • 4-64
 - requesting interrupt • 4-68
- VAX 8530 computer
 - bootstrap procedure for XDELTA • 4-65
 - requesting interrupt • 4-68
- VAX 8550 computer
 - bootstrap procedure for XDELTA • 4-65
 - requesting interrupt • 4-68
- VAX 8600 computer
 - bootstrap procedure for XDELTA • 4-65
 - requesting interrupt • 4-68
- VAX 8650 computer
 - bootstrap procedure for XDELTA • 4-65
 - requesting interrupt • 4-68
- VAX 8700 computer
 - See VAX 8810

Index

- VAX 8800 computer
 - See also VAX 8820-N
 - deadlock situation • 3–80
- VAX 8810 computer
 - bootstrap procedure for XDELTA • 4–65
 - requesting interrupt • 4–68
- VAX 8820 computer
 - bootstrap procedure for XDELTA • 4–65
 - requesting interrupt • 4–68
- VAX 8820-N computer
 - bootstrap procedure for XDELTA • 4–65
 - requesting interrupt • 4–68
- VAX 8830 computer
 - bootstrap procedure for XDELTA • 4–65
 - requesting interrupt • 4–68
- VAX 8840 computer
 - bootstrap procedure for XDELTA • 4–65
 - requesting interrupt • 4–68
- VAX 9000 computer
 - AUTOGEN parameter calculations for • 3–81
 - BI device driver requirement • 4–48
 - bootstrap procedure for XDELTA • 4–66
 - requesting interrupt • 4–68
- VAX Ada
 - CLOSE procedures change • 4–49
 - restrictions • 4–52
- VAXBI bus
 - DEBNI Ethernet/802 controller support • 3–39
- VAXBI restriction • 3–81
- VAX C
 - Run-Time Library changes • 4–53
 - Run-Time Library error checking • 4–54
 - socket routines • 4–54
- VAXcluster
 - failover • 3–18
 - reconfiguration time reduction • 3–82
- VAX computers, VMS support for • 3–82
- VAXft 3000 computer
 - bootstrap procedure for XDELTA • 4–67
 - procedure to cause a VAXft 3000 system failure • 3–83
 - requesting an interrupt for VAXft 3000 • 4–69
 - using \$GETDVI and F\$GETDVI for device information • 3–83
- VAXft 3000 systems
 - system messages • 2–34
 - VMS HELP for • 2–34
- VAX MACRO
 - NOP instruction • 4–55
 - restrictions • 4–59
 - TIMEDWAIT macro • 4–56
- VAX PL/I
 - run-time library • 4–59
- VAX Procedure Calling Standard
 - registers • 4–1
- VAXstation 2000 computer
 - bootstrap procedure for XDELTA • 4–62
 - requesting interrupt • 4–69
- VAXstation 3520 and 3540 computers
 - Ctrl/F2 key sequence • 3–84
 - Print Screen restriction • 2–14
 - supported software products • 3–84
- VAXstation 3520 computer
 - bootstrap procedure for XDELTA • 4–62
 - requesting interrupt • 4–68
- VAXstation 3540 computer
 - bootstrap procedure for XDELTA • 4–62
 - requesting interrupt • 4–68
- VAXstation 8000 computer
 - upgrade information • 3–79
- VAXTPU (VAX Text Processing Utility)
 - /WORK and /NOWORK qualifiers • 2–34
- VAX Workstation Software (VWS)
 - upgrade requirements • 3–79
- VIRTUALPAGECNT parameter • 3–74
- VMSINSTAL
 - DCLTABLES not used • 3–85
- VMSKITBLD procedure • C–1
- VMS kits
 - on RX33 diskettes • 3–79
- VMSLICENSE data file
 - parameter name correction • 5–14
- VMS RMS
 - new local buffers default • 4–41
 - RAB\$_ASY qualifier • 4–40
 - restriction removed for deferred write option • 4–40
 - XAB\$_NUL option • 4–41
 - XAB\$_NORECORD • 4–40
- VMS versions
 - computer support • 3–82
- VOLPRO privilege • 3–14
- Volume Shadowing phase II
 - AUTOGEN adjustment required • 3–86
 - HSC revision levels required • 3–87
 - implications for batch and print jobs • 3–86
 - SHOW DEVICES command • 3–87
 - VMSD3 parameter (SYSGEN) • 3–88
- VWS
 - See VAX Workstation Software (VWS)

W

- Widgets
 - redrawing in XUI Toolkit • 4–29
 - Window Manager
 - See DECwindows
 - Window systems
 - switching with AUTOGEN • 3–6
 - WRTJNL_BIJ error message
 - returns incorrect completion status value • 4–44
-

X

- X\$DISPLAY_STRING • 4–25
 - X.25 Packetnet Switch Interface • 3–22
 - XAB\$V_NUL option • 4–41
 - XDELTA
 - invoking • 4–60
-

- Xlib
 - corrected sequence problem • 4–25
 - programming • 4–25
 - XMI bus
 - DEMNA Ethernet/802 controller support • 3–40
 - X servers
 - interoperability with other vendors' X servers • 3–28
 - XUI Toolkit
 - corrections • 4–26
 - redrawing widgets • 4–29
 - unavailable VAX bindings for DRM routines • 4–26
 - VMS V5.2 changes • 4–26
-

Y

- YFDRIVER terminal port driver • 3–88
-

Z

- Zone analysis • 4–45
-

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-DEC-DEMO (800-332-3366) using a 1200- or 2400-baud modem. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	_____	Local Digital subsidiary or approved distributor
Internal ¹	_____	USASSB Order Processing - WMO/E15 <i>or</i> U.S. Area Software Supply Business Digital Equipment Corporation Westminster, Massachusetts 01473

¹For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

VMS Version 5.4
Release Notes
AA-LB22C-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completeness (enough information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clarity (easy to understand)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization (structure of subject matter)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Figures (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Examples (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index (ability to find topic)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Page layout (easy to find information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

I would like to see more/less _____

What I like best about this manual is _____

What I like least about this manual is _____

I found the following errors in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

I am using **Version** _____ of the software this manual describes.

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

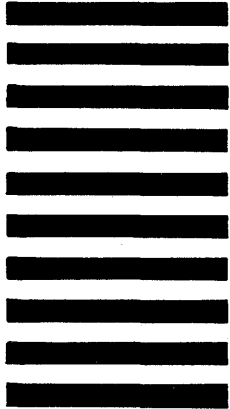
Phone _____

-- Do Not Tear - Fold Here and Tape --

digital™



No Postage
Necessary
if Mailed
in the
United States



BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
Corporate User Publications—Spit Brook
ZK01-3/J35 110 SPIT BROOK ROAD
NASHUA, NH 03062-9987



-- Do Not Tear - Fold Here --