
VMS Version 5.5 Release Notes

Order Number: AA-LB22D-TE

November 1991

This document describes changed software features, software problems and restrictions, and changes to documentation for Version 5.5 of the VMS operating system.

Revision/Update Information: This manual supersedes previous versions of VMS release notes.

Software Version: VMS Version 5.5

**Digital Equipment Corporation
Maynard, Massachusetts**

November 1991

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

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.

© Digital Equipment Corporation 1991.

All Rights Reserved.

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, DDIF, DEC, DECdtm, DECnet, DECram, DECUS, DECwindows, DECwriter, DEQNA, Desktop-VMS, DIGITAL, GIGI, HSC, LiveLink, LN03, MASSBUSS, MicroVAX, PrintServer 40, Q-bus, ReGIS, ULTRIX, UNIBUS, VAX, VAXcluster, VAX RMS, VAXserver, VAXstation, VMS, VT, XUI, and the DIGITAL logo.

Adobe, Display PostScript, and PostScript are registered trademarks of Adobe Systems Incorporated.

Motif, OSF, and OSF/Motif are registered trademarks of Open Software Foundation, Inc.

UNIX is a registered trademark of UNIX System Laboratories, Inc.

ZK5782

This document was prepared using VAX DOCUMENT, Version 1.2

Contents

Preface	xvii
---------------	------

1 General User Release Notes

1.1	VMS Version 5.5 Specific Release Notes for General Users	1-1
1.2	Characters Changed from Nonprintable to Printable	1-1
1.3	Batch and Print Queuing System Notes	1-2
1.3.1	\$ENTRY—New Symbol	1-2
1.3.2	DEFINE/FORM Command /SHEET_FEED Qualifier—Restriction ...	1-2
1.3.3	DELETE/ENTRY Command—Problem Corrected	1-2
1.3.4	Invalid Qualifiers Removed from Queue Commands	1-3
1.3.5	Print Accounting Record Page Count—Problem Corrected	1-3
1.3.6	SUBMIT/DELETE Command—Modifications	1-3
1.3.7	SUBMIT or PRINT /DELETE Qualifier—Problem Corrected	1-4
1.4	DECwindows Notes	1-4
1.4.1	Cutting and Pasting Between XUI Windows and DECwindows Motif Windows—Restriction	1-4
1.4.2	DECterm Terminal Emulator	1-4
1.4.2.1	Conformance Level Change for Support of Terminal State Reports	1-4
1.4.2.2	DECterm Graphics	1-5
1.4.2.3	DECterm—Default Server ID Specification Problem	1-6
1.4.2.4	Initializing DECterm	1-6
1.4.2.5	Memory—How to Reduce DECterm Use	1-7
1.4.2.6	Memory—DECterm Resize Request Problem	1-7
1.4.2.7	PC Interoperability Restrictions for DECterm	1-7
1.4.2.8	DECterm Text	1-8
1.4.2.9	VAXstation 3520 and 3540 Screens Blank Out	1-8
1.4.2.10	DECterm Window Scrolling Problem	1-8
1.4.3	DECW\$COLOR Guidelines for Changes in Achromatic Color (GrayScale) Rendering	1-9
1.4.4	DECwindows Startup Restriction	1-9
1.4.5	Desktop Applications	1-9
1.4.5.1	Calendar—Corrected Problems	1-9
1.4.5.2	Calendar—Restrictions and Limitations	1-9
1.4.5.3	Cardfiler—Scroll Bar	1-10
1.4.5.4	Clock—Changing the System Time	1-10
1.4.5.5	CDA Viewer—Corrections	1-10
1.4.5.6	CDA Viewer—Problems and Unexpected Behavior	1-10
1.4.5.7	CDA Viewer—Restrictions	1-11
1.4.5.8	DECwindows Mail—Problems and Restrictions	1-12
1.4.5.9	Paint—Corrected Problems	1-12
1.4.5.10	Print Screen—Corrections	1-12
1.4.5.11	Print Screen—Problems and Unexpected Behavior	1-12

1.4.5.12	Print Screen—VAXstation 3520 and 3540 Restriction	1-13
1.4.6	Server Pseudomouse Key Combination Change	1-13
1.4.7	Session Manager	1-13
1.4.7.1	Corrected Problems	1-13
1.4.7.2	User-Defined Logo Support Added	1-13
1.4.8	SYSGEN Parameter PQL_MPRCLM and Captive Accounts	1-14
1.4.9	ULTRIX—Authorization Guidelines for DECwindows Applications	1-14
1.4.10	Window Manager—Enhancements	1-14
1.4.11	Window Manager (Icon Box)—Corrected Problems	1-15
1.4.12	Workstations with Dual-Head Support—Restrictions	1-15
1.5	DIGITAL Command Language (DCL) Notes	1-16
1.5.1	CONVERT/DOCUMENT Command—New /MESSAGE_FILE Qualifier	1-16
1.5.2	DCL Command Verb and Qualifier Length	1-16
1.5.3	DIRECTORY Command	1-16
1.5.4	ENDSUBROUTINE Command—Correct Usage	1-16
1.5.5	OPEN Command	1-17
1.5.5.1	Negating Problem Corrected	1-17
1.5.5.2	VFC Record Format Used	1-17
1.5.6	RUN (Process) Command—Modifications	1-17
1.5.7	SET HOST/DTE Command—Modifications	1-17
1.5.8	SHOW MAGTAPE Command Now Obsolete	1-18
1.5.9	Symbol Names—Use Caution When Making Symbol Name Assignments	1-18
1.5.10	SYNCHRONIZE Command—Restriction	1-18
1.6	DUMP Command on KZQSA Tape Drives	1-18
1.7	Ethernet/802 Controllers—New Support	1-18
1.8	F\$CONTEXT Lexical Function Problem	1-18
1.9	Forced Password Changes at Login	1-19
1.10	Image Data Verification in Batch Mode—Problem Corrected	1-19
1.11	Mail Utility—Changes to PRINT/QUEUE Command	1-19
1.12	Process Identification (PID)—All Significant Digits Must Be Specified	1-19
1.13	Tape Support—Acceptance of ANSI Initialized Magnetic Tapes	1-20
1.14	TLZ04 Tape Drive Performance Considerations	1-20
1.15	VAXstation 3100, Model 76 Computer Release Notes	1-20
1.16	VAX Text Processing Utility (VAXTPU) Notes	1-20
1.16.1	/NOWORK Qualifier Problem	1-21
1.16.2	/WORK Qualifier Problem	1-21
1.16.3	Display Manager Definition Restriction	1-21
1.17	VMS Mail Utility	1-21
1.17.1	Folder Name Parameter Now Supports Mixed Case	1-21
1.17.2	Multiple Copies of a Message Delivered to the Same Recipient—Problem Corrected	1-21
1.17.3	Use Quotes in Address When Forwarding Mail to ULTRIX Users	1-22

2 System Manager Release Notes

2.1	VMS Version 5.5 Specific Release Notes for System Managers	2-1
2.2	Automatic Attempt to Load Continuation Volume of Magnetic Tape	2-2
2.3	\$CREATE_RDB System Service—New Location for Rights Database	2-3
2.4	Authorize Utility Notes	2-3
2.4.1	Adding Proxy Accounts	2-3
2.4.2	Minimum Lengths for Generated Passwords	2-3
2.4.3	Modifications to the RESTRICTED Flag	2-3

2.4.4	Setting the DISCTLY Flag	2-4
2.4.5	Submitting Batch Jobs	2-4
2.5	AUTOGEN Command Procedure Notes	2-4
2.5.1	OPER Privilege Requirement	2-4
2.5.2	Switching Window Systems	2-5
2.6	Backup Utility Notes	2-6
2.6.1	Backing Up Files Marked for Recovery Unit Journaling	2-6
2.6.2	BACKUP ACL Behavior—Change	2-6
2.6.3	BACKUP/RECORD Copy Operations Problem	2-6
2.6.4	Backup Utility—Problem Corrected	2-6
2.6.5	Image Save Operation Restriction	2-7
2.6.6	Using BACKUP with Compound Document Files	2-7
2.7	Batch and Print Queuing System Notes	2-7
2.7.1	Supported Versions for the VMS Batch and Print Queuing System ...	2-7
2.7.2	DECdtm Files Required for the Batch and Print Queuing System ...	2-8
2.7.3	Deleting the DEFAULT Print Form	2-8
2.7.4	Changing Print Forms and Characteristics	2-9
2.7.5	Generic Queue Restriction Lifted	2-9
2.7.6	Job Scheduling Priority	2-9
2.7.7	Print Symbiont Working Set Purge Less Frequent	2-10
2.7.8	Queue Manager Notes	2-10
2.7.8.1	Change in Behavior	2-10
2.7.8.2	Change in the /NEW_VERSION Qualifier	2-10
2.7.8.3	Mounting the Queue Database Disk	2-10
2.7.8.4	Obsolete Qualifiers	2-11
2.7.9	SHOW QUEUE and SHOW ENTRY Displays	2-11
2.7.10	STOP/QUEUE/RESET Command	2-11
2.7.11	SYSGEN Parameters for Queuing	2-12
2.7.12	SET TIME/CLUSTER Command Synchronizes Cluster Time	2-12
2.7.13	Using Banner Pages for Jobs Printed from MAIL—Change in Behavior	2-12
2.7.14	VAX Distributed Queuing System (DQS)—Recommended Version ...	2-13
2.8	Buffered I/O Byte Limit (BYTLM)—Increase Required	2-13
2.9	COBRTL Separate Installation Requirement Removed	2-13
2.10	Debugger Notes	2-13
2.10.1	System Management Considerations	2-13
2.10.2	System Resources	2-14
2.10.3	User Quotas	2-14
2.11	DECnet-VAX Notes	2-14
2.11.1	DECnet Links Dropping—Problem Corrected	2-14
2.11.2	NCP/NML Requires OPER Privilege to Obtain Service Passwords ...	2-14
2.11.3	NETACP\$BUFFER_LIMIT Logical Name—To Override Default BYTLM Quota	2-15
2.11.4	NETDRIVER Acknowledges Delay Time	2-15
2.11.5	NML Checks for Illegal Address Configurations	2-15
2.11.6	Node-Level Access Control Problem Corrected	2-15
2.11.7	Startup Restriction with Ethernet-Based Applications in a Local Area VAXcluster	2-15
2.11.8	SYS\$CLUSTER_NODE Logical Name—Correction	2-16
2.11.9	UNA Circuit Default Cost Changed	2-16
2.12	DECwindows Notes	2-16
2.12.1	DECwindows Server Logical Switch for Server Connect/Disconnect Messages	2-16
2.12.2	DECwindows Startup	2-16

2.12.3	Starting the ULTRIX Connection (UCX) Before DECnet—Problem . . .	2-16
2.12.4	Tailoring DECwindows	2-17
2.12.5	Template File—New DECW\$SYLOGIN.TEMPLATE	2-17
2.12.6	Template File—Support for Configuring Multihead Systems	2-17
2.12.7	X Servers—Interoperability Restriction	2-17
2.13	Disk Quota Cache Entries—Maximum Value	2-18
2.14	Disk Header Space Problem	2-18
2.15	DISMOUNT Command—Changes Regarding Open Files	2-20
2.15.1	Closing Files	2-21
2.15.2	Clusterwide Support for DISMOUNT	2-22
2.15.3	Restoring the Previous Behavior of the DISMOUNT Command	2-22
2.16	DNS RTL Routines DNS\$PARSE_USERNAME_STRING and DNS\$CVT_TO_USERNAME_STRING—Corrections	2-23
2.17	DSSI Device Naming No Longer Dependent on SYSGEN Parameter VMS5	2-23
2.18	Ethernet Notes	2-25
2.18.1	DEBNI Ethernet/802 Controller—New Support for the VAXBI Bus	2-25
2.18.1.1	I/O Interface	2-25
2.18.2	DEMNA Ethernet/802 Controller—New Support for the XMI Bus	2-26
2.18.3	DEQTA Ethernet/802 Controller—New Support for the Q-bus	2-26
2.18.4	Second-Generation Ethernet Controller (SGEC) Now Supported	2-27
2.19	Hierarchical Storage Controller (HSC) Revision Levels Required	2-27
2.20	ISL Support for VAX Systems—Extended	2-27
2.21	Multiple Backups Now Supported	2-28
2.22	LAT Release Notes	2-28
2.22.1	Setting Up LAT on the System	2-29
2.22.2	Installing and Setting Privileges for LATCP	2-29
2.22.3	LATCP Commands Replaced	2-29
2.22.4	Modifications to LATCP Command SET NODE	2-30
2.22.5	LATCP Qualifiers Ignored (Offering Services Over Specific Links Removed)	2-30
2.22.6	Additional LATCP Commands and Qualifiers Ignored	2-31
2.22.7	Using the LATCP Command SET NODE/STATE=OFF or /STATE=SHUT	2-31
2.22.8	LAT Print Symbiont (LATSYM)	2-31
2.22.9	READSYNC Not Supported by LAT Software	2-31
2.22.10	Setting the Large Request Packet (LRP) Value to Support X Terminals	2-31
2.22.11	Setting LTA MAX Units Through LATCP	2-32
2.22.12	Saving Nonpaged Pool Memory and Avoiding Duplicate Application Ports	2-32
2.22.13	CREATE PORT/LOGICAL Command and BYTCNT Quota	2-33
2.22.14	Using Dedicated Ports	2-33
2.22.15	LAT Support for FDDI Controller	2-33
2.23	Magnetic Tape ACP Notes	2-33
2.23.1	Correction to I/O Process	2-33
2.23.2	Undocumented Implementation Removed	2-33
2.24	Monitor Utility Notes	2-34
2.24.1	Error in Display	2-34
2.24.2	MONITOR CLUSTER Command	2-34
2.24.2.1	Error Messages	2-34
2.24.2.2	Network Load Problem	2-35
2.24.3	RMS Bucket and Multibucket Split Rates Invalid	2-35
2.24.4	VMS Volume Shadowing (Phase II) Disk Names Display Incorrect	2-35

2.25	NMLSHR Image Superseded	2-36
2.26	SCSI Notes	2-36
2.26.1	SCSI Concepts	2-36
2.26.2	Bad Block Support for SCSI Disks	2-37
2.26.2.1	BAD Utility	2-37
2.26.2.2	Console Disk Formatter	2-37
2.26.2.3	INITIALIZE Command and the File System	2-37
2.26.2.4	SCSI Disk Class Driver	2-37
2.26.3	Error Recovery Parameters	2-38
2.26.4	What the Disk Class Driver Does When It Detects an Error	2-38
2.26.5	Use of Disk Caching Products with SCSI Disks	2-39
2.26.6	Timing in MicroVAX 3100 SCSI Port Driver Fixed	2-39
2.27	Security Features—Notes	2-39
2.27.1	COMMSYNC Qualifier to the SET TERMINAL Command	2-39
2.27.2	Department of Defense (DoD) Erase Pattern Corrected	2-39
2.27.3	Modifying the System Password Dictionary	2-39
2.27.4	NETCONFIG.COM Security Enhancements	2-40
2.27.5	OPCOM on Nonclustered MicroVAX Systems	2-41
2.27.6	Reestablishing the Security Environment	2-41
2.27.7	Security Audit Alarm Settings Preserved Between System Boots	2-41
2.27.8	Security Auditing Failure-Mode Settings Preserved Across Initializations	2-41
2.27.9	Spawned Subprocesses that Execute SET UIC	2-41
2.27.10	Suppressing Duplicate Logging of Security Alarms by OPCOM	2-42
2.27.11	SYSECURITY.COM Command Procedure—New Site-Specific Configuration File	2-42
2.27.12	User Authorization File (UAF) Notes	2-43
2.27.12.1	Captive Accounts—Batch and Network Restrictions Removed	2-43
2.27.12.2	Captive Accounts—Security and Application PRINT Commands	2-43
2.27.12.3	Changes to the CAPTIVE Flag	2-44
2.27.12.4	DISIMAGE Flag	2-46
2.27.12.5	RESTRICTED Accounts—Incompatibility Problems Corrected	2-47
2.27.12.6	UAF Record Length Enforcement	2-47
2.27.12.7	UAF Template File Changes	2-47
2.28	Security Review Recommended	2-48
2.29	SPKITBLD.COM Procedure—Change	2-48
2.30	Standalone BACKUP Correction	2-49
2.31	Standalone BACKUP Restriction	2-49
2.32	Startup and Shutdown Procedures Modified	2-49
2.32.1	Startup Procedure Template Changes	2-49
2.32.2	SHUTDOWN.COM—Changes for Autostart Queues	2-50
2.32.3	SHUTDOWN.COM—Change in Disk Dismount Reporting	2-50
2.33	SYSGEN Utility Notes	2-50
2.33.1	Configuration Note	2-51
2.33.2	CTLPAGES Parameter	2-51
2.33.3	PRIORITY_OFFSET Parameter	2-51
2.33.4	Symmetric Multiprocessing (SMP) Not Supported on Uniprocessor Systems Running DECwindows Server	2-51
2.34	SYSLOA—Page Fault Problem Corrected	2-52
2.35	System Disk Size Recommendation: 115 MB	2-52
2.36	TA90E Tape Drive	2-52
2.36.1	TA90E Usage Guidelines	2-52
2.36.2	Limitation of Using BACKUP with the TA90E Tape Drive	2-53

2.37	TLZ04 Tape Driver—Problem Corrected	2-53
2.38	UETP Load Phase Failure in Vector Processing Systems	2-53
2.39	UNIBUS and Q22 Bus Interrupt Vector Change	2-54
2.40	Upgrade Notes	2-56
2.40.1	Applying the PADRIVER01u3054 to VMS Version 5.4-3 System Disks	2-56
2.40.2	CIXCD Firmware/Microcode Requirements for Booting VMS Version 5.5	2-56
2.40.3	Layered Product Cautions	2-56
2.40.3.1	ALL-IN-1 Version 2.4 Shareable Images Requirement for CDA Support	2-57
2.40.3.2	DECwindows Developer's Kit Requirement	2-57
2.40.3.3	VAX APL Interpreter Requirement	2-58
2.40.4	Rolling Upgrades from VMS Version 5.4 to Version 5.5	2-58
2.40.5	VAXstation 8000 Upgrade Unsupported	2-58
2.41	VAXft SYSTEMS—EFDRIVER OPCOM Messages	2-58
2.42	VAX 3100 Model 76 Notes	2-60
2.42.1	System Crash	2-60
2.42.2	AUTOGEN LOCKDIRWT Parameter	2-60
2.43	VAX 6000 Series Computer Notes	2-60
2.43.1	VAX 6000 Series Console Tapes and Tape Serving Devices	2-60
2.43.2	VAX 6000 Model 500 Series Configuration Note	2-62
2.43.3	VAX 6000 Model 600 Series—Restriction	2-62
2.43.4	DSSI and FDDI Boot Devices Supported on VAX 6000 Series Systems	2-62
2.43.4.1	Bootting Standalone BACKUP on a VAX 6000 Series Computer from an InfoServer System	2-62
2.44	VAX 8000 Series Computer Notes	2-62
2.44.1	DMB32 Product Software Required for DMB32 Communications Controller	2-62
2.44.2	Halting a VAX 8530, 8550, 8700, or 8800 System	2-63
2.44.3	SET TIME/CLUSTER Command—Caution	2-63
2.44.4	SET TIME Command—Problem	2-63
2.44.5	VAX 8800 Systems Running SMP—Problem	2-63
2.44.6	VAXBI 5 Restriction	2-64
2.45	VAX 9000 Computer—AUTOGEN with FEEDBACK Requirement	2-64
2.46	VAX Computers—VMS Support	2-64
2.47	VAXstation 3520 and 3540 Computer Notes	2-65
2.47.1	Console Support on the Graphics Terminal	2-65
2.47.2	Software Products Availability	2-65
2.48	VAXstation 4000 Series Computer—Changing Font Size	2-65
2.49	VMS License Management Facility Notes	2-66
2.49.1	Increasing License Capacity	2-66
2.49.1.1	Replacement PAKs	2-66
2.49.1.2	Additive PAKs	2-66
2.49.1.3	PAKs with the NOSHARE option	2-66
2.49.2	List Size Restrictions	2-66
2.49.2.1	The Per PAK Limit = 5000 Characters	2-67
2.49.2.2	The Per Product Limit = 30000 Characters	2-67
2.49.3	Logical Name LMF\$DISPLAY_OPCOM_MESSAGE	2-68
2.49.4	Preserving License Audit Trails	2-68
2.49.5	Purging License History Records	2-68
2.49.6	Installation Considerations for VAXclusters	2-69
2.50	VMS VAXcluster Notes	2-70

2.50.1	Large VAXcluster Configurations—SYSGEN Parameter SCSCONNCNT	2-71
2.50.2	Mixing a Multiadapter Local Area VAXcluster and VMS Version 5.4-2 or Earlier	2-71
2.50.3	Selecting Boot Disk Path	2-72
2.50.4	SCS SYSGEN/TIMVCFAIL Parameter	2-72
2.50.5	VAXcluster Reconfiguration Time Reduced	2-72
2.50.6	VAXcluster Support for FDDI Adapter	2-72
2.50.7	VAXcluster Support for Multiple LAN Adapters	2-72
2.50.8	VAXcluster Use of DEMNA	2-73
2.51	Volume Shadowing (Phase II) Notes	2-73
2.51.1	Overview	2-74
2.51.2	Batch and Print Jobs—Reentering After Conversion	2-74
2.51.3	Changing Physical Device Names	2-74
2.51.4	Corrections	2-75
2.51.4.1	SHOW DEVICES Display Is Consistent with Shadow Set Membership Status	2-75
2.51.4.2	Volume Shadowing Merge Copy Progresses Successfully	2-75
2.51.5	MOUNT/CLUSTER Command	2-76
2.51.6	SHADBOOTFAIL Bugcheck Message	2-76
2.51.7	Shutting Down a System with a VMS Volume Shadowing (Phase II) Shadowed System Disk	2-76
2.51.8	Obtaining \$GETDVI FREEBLOCK Information	2-76
2.51.9	SYSGEN Parameter VIRTUALPAGECOUNT Adjustment	2-76
2.51.10	UAF Parameter Changes	2-77
2.52	Working Set Quota (WSQUOTA) Size Limited	2-77
2.53	YFDRIVER Terminal Port Driver Supports New Baud Rate	2-78

3 Programmer Release Notes

3.1	VMS Version 5.5 Specific Release Notes for Programmers	3-1
3.2	Activating an Image with System Version Mismatch—Change	3-1
3.3	BASIC INKEY\$ Function—Change	3-2
3.4	DCL Subroutine—Modifications to Subroutine Entry Point Scoping	3-2
3.5	DCL Substring Assignment Problem Corrected	3-3
3.6	Debugger Notes	3-3
3.6.1	Corrected Problems or Restrictions	3-3
3.6.2	Debugger Problems or Restrictions	3-4
3.6.2.1	DEPOSIT/TYPE Command with C Programs	3-4
3.6.2.2	\$WAKE Call Followed by \$HIBER Call	3-4
3.6.2.3	Examining LABEL[n] for a Code Location	3-5
3.6.2.4	Register Window in DECwindows Interface	3-5
3.6.2.5	SET IMAGE Command Limitation	3-5
3.6.2.6	Using Concealed Rooted-Directory Logical Names for Source Files	3-5
3.6.2.7	Using Debugger Commands in DCL Command Procedures	3-5
3.6.2.8	Using the Abort Key or Stop Button After a SPAWN Command ..	3-6
3.6.2.9	Using the Debugger on a VAXstation Running VWS	3-6
3.6.2.10	Using the DECwindows Interface	3-6
3.6.2.11	Vector-Support Restrictions and Problems	3-8
3.6.2.12	Watchpoints in Installed Writable Shareable Images	3-8
3.6.3	Use of Single-Process (Pre-Version 5.2) Debugging Configuration ..	3-9
3.7	DEC CDA Base Services	3-10
3.7.1	User Get and Put Routine Changes	3-10

3.7.2	Sending Compound Documents with PostScript/Text References	3-10
3.8	DECnet—File Access Protocol Extensions	3-10
3.9	DECthreads Restriction	3-10
3.10	DECwindows Notes	3-11
3.10.1	DECterm Terminal Emulator	3-11
3.10.1.1	Color Table Report—Reporting Problem Corrected	3-11
3.10.1.2	CREATE/TERMINAL Command—Negative Values Problem Corrected	3-11
3.10.1.3	ReGIS Locator Report—Omitted Coordinates	3-11
3.10.1.4	VT52-Mode Cursor Addressing—Restriction Removed	3-11
3.10.2	DECwindows Server and Driver—General	3-11
3.10.2.1	Misspelled Cursor Screen Boundary \$QIO Constant	3-12
3.10.2.2	Problems and Restrictions	3-12
3.10.2.3	Setting a Cursor Pattern \$QIO Call	3-12
3.10.2.4	Virtual Memory Space Problem with Large Pixmaps	3-12
3.10.2.5	VAXstation Configurations	3-13
3.10.3	DECwindows X11 Display Server	3-14
3.10.3.1	PostScript (DPS) Extension Correction	3-14
3.10.3.2	Problem Updating Cursor Colors	3-14
3.10.3.3	Problem Accessing Layered Products Fonts	3-14
3.10.3.4	Narrow Terminal Fonts—Increased Height	3-15
3.10.4	Display PostScript Notes	3-15
3.10.4.1	Display PostScript Documentation	3-15
3.10.4.2	Ada Bindings Not Available	3-15
3.10.4.3	VAX Calling Standard Bindings—DPS\$PRINTF Routine Not Implemented	3-15
3.10.5	SET DISPLAY Command—Restriction	3-15
3.10.6	User Interface Language (UIL) Compiler Notes	3-16
3.10.6.1	Built-In Tables—Additions	3-16
3.10.6.2	Corrected Problems	3-16
3.10.6.3	Specifying Multiline Compound Strings	3-17
3.10.7	XUI Toolkit Notes	3-18
3.10.7.1	DRM Routines—Unavailable VAX Bindings	3-18
3.10.7.2	Font-Unit Values—Change in Properties	3-18
3.10.7.3	Internal Format of Compound Strings	3-18
3.10.7.4	List Box Dynamic Sizing	3-18
3.10.7.5	Problems and Restrictions in the XUI Toolkit	3-18
3.10.7.6	Redrawing Widgets	3-19
3.10.7.7	Selection Push Buttons	3-20
3.11	Disk Forced Error Counting Implemented	3-20
3.12	DSA Devices	3-21
3.12.1	DSA Disk Driver—Alternate Host Information Change	3-21
3.12.2	I/O Preferred Path Interface for DSA Devices—Problem Corrected	3-21
3.13	I/O Device Driver Notes	3-22
3.13.1	Logical End-of-Volume Detection Now Always in Effect	3-22
3.13.2	Opening a Sequential-Media File Now More Efficient	3-22
3.13.3	User EOT Mode Correction	3-22
3.14	IF-THEN-ELSE Construct and \$STATUS Symbol	3-22
3.15	LAT Release Notes	3-22
3.15.1	Change to Disconnect Processing	3-23
3.15.2	FORTRAN Programs Using \$LATDEF	3-23
3.15.3	New SETMODE and SENSEMODE QIO Functions	3-23
3.15.4	Forward Port Connections and Disconnections	3-23
3.15.5	LTA0 Now a Template Device	3-23

3.15.6	Port Modifier QIO Functions Ignored	3-24
3.15.7	QIO Completion Status	3-24
3.15.8	QIO Functions Superseded	3-24
3.16	Lock Manager	3-25
3.16.1	Dynamic Remastering	3-25
3.16.2	Extended Lock ID Table	3-25
3.17	Linker Utility	3-25
3.17.1	Analyzing Images Created with the VMS Version 5.5 Linker	3-25
3.17.2	Linker Now Combines Psects with the GBL, OVR, and NOSHR Attributes Set	3-25
3.18	Mailbox Driver Write	3-26
3.19	Message Router Version 3.0 Installation	3-26
3.20	Modem Signal Control Change	3-27
3.21	Processor Register Definition Symbols	3-27
3.22	Programming Languages—Reinstallation Required	3-28
3.23	Run-Time Library (RTL) Notes	3-28
3.23.1	LIB\$CREATE_VM_ZONE Routine—New Flags Added	3-28
3.23.2	LIB\$DECODE_FAULT Use with Vector Processor	3-29
3.23.3	PPL\$TRIGGER_EVENT Routine Memory Problem Corrected	3-29
3.23.4	Specifying Input Format Mnemonics	3-29
3.24	SET HOST/DTE/DIAL Command—DMF-32 Controller Problem	3-30
3.25	VAX 9000 Computer—BI Device Drivers Conformance Requirement	3-30
3.26	VAX Ada Run-Time Library Notes	3-30
3.26.1	VAX Ada Version 2.2—Superseded	3-30
3.26.2	CALENDAR.SPLIT Improvement	3-31
3.26.3	CLOSE Procedures—Change in Implementation	3-31
3.26.4	CONSTRAINT_ERROR Now Raised in Place of NUMERIC_ERROR	3-31
3.26.5	'STORAGE_SIZE Attribute Change	3-31
3.26.6	Restriction on END_OF_FILE Function	3-32
3.26.7	VAX Vector Registers Not Saved During Task Switches	3-32
3.27	VAX C Notes	3-32
3.27.1	Error Checking Enhanced	3-32
3.27.2	Mixing D_FLOAT and G_FLOAT Modules	3-32
3.27.3	Run-Time Library Correction	3-33
3.27.4	Sample Programs	3-33
3.28	VAX MACRO Notes	3-33
3.28.1	VAX MACRO Corrections	3-33
3.28.1.1	Debug Symbol Table (DST) Corruption	3-33
3.28.1.2	Debug Symbol Table (DST) Corruption Caused by a Repeat Loop in a Macro	3-35
3.28.2	Known Problems in VAX MACRO	3-37
3.29	VAX Math Run-Time Library (MTHRTL)—Improved Accuracy	3-38
3.30	VMS Convert/Reclaim Utility—Problem	3-38
3.31	VMS Record Management Services (RMS) Notes	3-39
3.31.1	Expiration of RMS Files—Change	3-39
3.31.2	RAB\$V_ASY Qualifier Now Supported for Process-Permanent Files	3-40
3.31.3	RMS Corrections	3-40
3.31.4	RMS Indexed File Local Buffers—New Default	3-41
3.31.5	RMS Statistics Restrictions	3-41

3.31.6	RMS Journaling Notes	3-42
3.31.6.1	Moving Recovery-Unit Journalled RMS Indexed Files to Systems Running VMS Version 4.7 and Earlier	3-42
3.31.6.2	SET FILE/BI_JOURNAL or SET FILE/BI_JOURNAL Command—Correct Usage	3-42
3.31.6.3	Synchronous Recovery	3-42
3.31.6.4	SYSGEN Parameter PIOPAGES—Change in Usage	3-43
3.32	VMS System Services Notes	3-43
3.32.1	Error Messages Referring to Shared Memory	3-43
3.32.2	Mailbox Driver Error Checking	3-43
3.32.3	\$GETQUI—QUI\$_QUEUE_STATUS Longword	3-44
3.32.4	\$GETSYI System Service: VAX Computer Model Numbers Added to Item Codes	3-44
3.32.4.1	Model Numbers for the SYI\$_CPU Item Code	3-44
3.32.4.2	Model Numbers for the SYI\$_XCPU Item Code	3-45
3.32.4.3	Model Numbers for the SYI\$_HW_NAME Item Code	3-45
3.32.5	Self-Modifying Item Lists with \$GETxxx Services	3-46
3.32.6	SJC\$_START_QUEUE_MANAGER Function Code for \$SNDJBC Service	3-47
3.32.7	\$SNDJBC Service—Obsolete Item Codes	3-47
3.33	XDELTA Notes	3-47
3.33.1	Invoking XDELTA	3-47
3.33.1.1	Booting VMS Computer Systems Post Version 5.4	3-48
3.33.1.2	Booting a VAX-11/730 or VAX-11/750 Computer Using the VMS Console TU58	3-48
3.33.1.3	Booting XDELTA on a MicroVAX 2000, VAXstation 2000, MicroVAX 3400 Series, VAXstation 3520, VAXstation 3540, MicroVAX/VAXstation 3600 Series, MicroVAX 3900, VAX 4000 Series, MicroVAX II, or VAX-11/750 Computer	3-49
3.33.1.4	Booting XDELTA on a VAX-11/780 or a VAX-11/785 Computer	3-50
3.33.1.5	Booting XDELTA on a VAX 6000-Series Computer	3-50
3.33.1.6	Booting XDELTA on a VAX 8200, 8250, 8300, or 8350 Computer	3-51
3.33.1.7	Booting XDELTA on a VAX 8530, 8550, 8810 (8700), 8820, 8820-N (8800), 8830, or 8840 Computer	3-51
3.33.1.8	Booting XDELTA on a VAX 8600 or 8650 Computer	3-52
3.33.1.9	Booting XDELTA on a VAX 9000 Computer	3-53
3.33.1.10	Booting XDELTA on a VAXft 3000 Computer	3-54
3.33.1.11	Booting XDELTA on a VAXstation 3100 or MicroVAX 3100-Series Computer	3-54
3.33.2	Requesting an Interrupt	3-54

4 Documentation Release Notes

4.1	VMS Version 5.5 Specific Release Notes Pertaining to VMS Documentation	4-1
4.2	<i>Using VMS BACKUP</i>	4-2
4.3	<i>Building Dependable Systems: The VMS Approach</i>	4-2
4.4	<i>Guide to Maintaining a VMS System</i>	4-2
4.5	<i>Guide to VMS File Applications</i>	4-2
4.6	<i>Guide to VMS Programming Resources</i>	4-2
4.7	<i>VAX MACRO and Instruction Set Reference Manual</i>	4-3
4.8	<i>VMS DCL Dictionary</i>	4-3
4.8.1	COPY/TRUNCATE Command	4-3

4.8.2	DEFINE/FORM Command	4-3
4.8.3	DELETE/QUEUE Command	4-3
4.8.4	F\$FAO Lexical Function	4-3
4.8.5	F\$GETDVI Lexical Function	4-4
4.8.6	F\$GETQUI Lexical Function	4-4
4.8.7	INITIALIZE/MEDIA_FORMAT=[NO]COMPACTION Command	4-4
4.8.8	PRINT/PRIORITY Command	4-4
4.8.9	REPLY Command	4-4
4.8.10	RUN (Process) Command	4-4
4.8.11	SET CLUSTER/EXPECTED_VOTES Command	4-5
4.8.12	SET DEVICE/SERVED Command	4-5
4.8.13	SET ENTRY/AFTER Command	4-5
4.8.14	SET ENTRY/PRIORITY Command	4-5
4.8.15	SET ACL/OBJECT_TYPE Command	4-5
4.8.16	SHOW SYSTEM/INTERACTIVE Command	4-5
4.8.17	SUBMIT/PRIORITY Command	4-6
4.9	<i>VMS DECwindows Guide to Application Programming</i>	4-6
4.10	<i>VMS DECwindows User Interface Language Reference Manual</i>	4-8
4.11	<i>VMS DECwindows Xlib Programming Volume</i>	4-9
4.12	<i>VMS Device Support Manual</i>	4-9
4.12.1	DMA I/O Operations—Driver Code Reference	4-9
4.12.2	UNIBUS and Q22 Bus CSR Address Assignment	4-10
4.12.3	VAX 8600/8650 I/O Offset Value	4-10
4.13	<i>VMS Device Support Reference Manual</i>	4-10
4.13.1	COM\$SETATTNAST Routine	4-10
4.13.2	EXE\$ALLOCBUF and EXE\$ALLOCIRP Routine	4-10
4.13.3	SPI\$CONNECT Macro Return Values	4-10
4.13.4	SPI\$GET_CONNECTION_CHAR and SPI\$SET_CONNECTION_CHAR Macro Buffer Format	4-11
4.14	VMS Installation and Operations Manuals—Title Changes	4-11
4.15	<i>VMS I/O User's Reference Manual: Part I</i>	4-11
4.16	<i>VMS Linker Utility Manual</i>	4-12
4.17	<i>VMS Monitor Utility Manual</i>	4-12
4.18	<i>VMS Networking Manual</i>	4-12
4.19	<i>VMS Record Management Services Manual</i>	4-13
4.20	<i>VMS RTL Library (LIB\$) Manual</i>	4-14
	LIB\$SYS_TRNLOG	4-15
4.21	<i>VMS RTL Screen Management (SMG\$) Manual</i>	4-18
4.22	<i>VMS System Generation Utility Manual</i>	4-18
4.23	<i>VMS System Services Reference Manual</i>	4-18
4.24	<i>VMS Upgrade and Installation Supplement: VAXstation 2000, MicroVAX 2000</i>	4-18
4.25	<i>VMS Upgrade and Installation Supplement: VAX 8600, 8650</i>	4-19
4.26	<i>VMS Utility Routines Manual</i>	4-19
4.27	<i>VMS Version 5.4-3 Release Notes</i>	4-19

A VMS DECwindows Performance Considerations

A.1	Recommended Minimum Memory Configurations for DECwindows	A-1
A.2	Running VMS DECwindows Applications Remotely	A-1
A.3	Suggestion for Running Applications Remotely	A-2
A.4	Using AUTOGEN	A-2
A.5	Improving DECwindows Memory Sharing	A-3

B Building and Copying a VMS System Disk

B.1	Introduction	B-1
B.2	Building the Operating System on Another Disk	B-1
B.3	Copying the Operating System Files to Another Disk	B-3
B.4	Adding an Alternate System Root Directory	B-4

C File Design Attributes

C.1	File Design Attributes	C-1
C.1.1	Initial File Allocation	C-1
C.1.2	Contiguity	C-1
C.1.3	Extending a File	C-2
C.1.3.1	Auto-Extend Size Selection	C-2
C.1.3.2	Establishing Auto-Extend Default Quantities	C-3
C.1.3.3	Placement and Contiguity of Extends	C-5
C.1.4	Truncating a File	C-6
C.1.5	Units of I/O	C-7
C.1.6	Multiple Areas for Indexed Files	C-7
C.1.7	Bucket Fill Factor for Indexed Files	C-7

Index

Examples

2-1	Monitor Utility CLUSTER Class Display Incorrect	2-35
2-2	SHOW DEVICES Display	2-75
3-1	Sample ERL\$LOGSTATUS Entry	3-20
3-2	Sample Debug Session Before the Fix Is Applied	3-35
3-3	Sample Debug Session After Fix Is Applied	3-36
3-4	Source Code to Generate Sample Sessions	3-36
4-1	Freeing Compound Strings in the Hello World! Sample Application	4-7
4-2	Freeing Memory Allocated by the VMS SET ARG Routine	4-8

Figures

2-1	Valid Mixed Version VAXcluster with Separate License Databases ...	2-70
2-2	Invalid Mixed Version VAXcluster with Shared License Database ...	2-70

Tables

1-1	DECterm Gray Levels	1-5
1-2	Session Manager Logical Names	1-14
2-1	KFQSA Controller Device Names	2-24
2-2	ISL-Capable Systems	2-27
2-3	VMS Versions and Associated VAX Computers	2-64
2-4	Ethernet and FDDI Adapters	2-73
2-5	Recommended UAF Parameters for the System Account	2-77
3-1	Additional UIL Resources	3-16

3-2	New Flags Added to LIB\$CREATE_VM_ZONE	3-29
3-3	Valid XDELTA Values for the BOOT Command Qualifier	3-48
3-4	Booting XDELTA on VMS Computer Systems Post Version 5.4	3-48
3-5	XDELTA BOOT Command Qualifiers for VAX 6000-Series Computers	3-50
3-6	BOOT Command Qualifiers for VAX 9000 Computers	3-53
4-1	Arguments Unsupported by the UIL Compiler	4-9



Preface

Intended Audience

This manual is intended for all system users. Read this manual before you install, upgrade, or use Version 5.5 of the VMS operating system.

Document Structure

This manual contains the following chapters and appendixes:

- Chapter 1 contains release notes intended for general users.
- Chapter 2 contains release notes intended for system managers.
- Chapter 3 contains release notes intended for programmers.
- Chapter 4 contains additions and corrections to manuals in the VMS documentation set.
- Appendix A contains DECwindows performance considerations.
- Appendix B contains replacement text for the section Building and Copying a VMS System Disk in the *Guide to Setting Up a VMS System*.
- Appendix C contains replacement text for Section 3.2.1 of the *Guide to VMS File Applications*.

Associated Documents

For additional information about VMS Version 5.5, see the following documents:

- *VMS Version 5.5 New Features Manual*
- *VMS Version 5.5 Upgrade and Installation Manual*
- *Guide to Maintaining a VMS System*
- *Guide to Setting Up a VMS System*
- *Guide to Using VMS*
- *Guide to VMS File Applications*
- *Guide to VMS Files and Devices*
- *Guide to VMS System Security*
- *Introduction to VMS System Management*
- *Introduction to VMS System Routines*
- *Introduction to VMS System Services*
- *Overview of VMS Documentation*
- *VAX RMS Journaling Manual*

- *VAX Text Processing Utility Manual*
- *VMS Accounting Utility Manual*
- *VMS Authorize Utility Manual*
- *VMS Backup Utility Manual*
- *VMS DCL Concepts Manual*
- *VMS DCL Dictionary*
- *VMS Debugger Manual*
- *VMS DECwindows User's Guide*
- *VMS Delta/XDelta Utility Manual*
- *VMS Developer's Guide to VMSINSTAL*
- *VMS Device Support Manual*
- *VMS Device Support Reference Manual*
- *VMS File Definition Language Facility Manual*
- *VMS I/O User's Reference Manual: Part I*
- *VMS I/O User's Reference Manual: Part II*
- *VMS LAD Control Program (LADCP) Manual*
- *VMS LAT Control Program (LATCP) Manual*
- *VMS Librarian Utility Manual*
- *VAX MACRO and Instruction Set Reference Manual*
- *VMS Monitor Utility Manual*
- *VMS Record Management Services Manual*
- *VMS RTL Library (LIB\$) Manual*
- *VMS RTL Mathematics (MTH\$) Manual*
- *VMS SYSMAN Utility Manual*
- *VMS System Dump Analyzer Utility Manual*
- *VMS System Generation Utility Manual*
- *VMS System Messages and Recovery Procedures Reference Manual*
- *VMS System Services Reference Manual*
- *VMS User's Manual*
- *VMS Utility Routines Manual*
- *VMS VAXcluster Manual*
- *VAX Volume Shadowing Manual*
- *VMS Volume Shadowing Manual*

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	Red ink indicates information that you must enter from the keyboard or a screen object that you must choose or click on. For online versions of the book, user input is shown in bold .
boldface text	Boldface text represents the introduction of a new term or the name of an argument, an attribute, or a reason. Boldface text is also used to show user input in online versions of the book.
<i>italic text</i>	Italic text represents information that can vary in system messages (for example, Internal error <i>number</i>).

UPPERCASE TEXT

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.

Hyphens in coding examples indicate that additional arguments to the request are provided on the line that follows.

numbers

Unless otherwise noted, all numbers in the text are assumed to be decimal. Nondecimal radices—binary, octal, or hexadecimal—are explicitly indicated.

General User Release Notes

This chapter discusses information about the VMS Version 5.5 operating system that is of interest to the general user.

For information about the new features included in VMS Version 5.5, see the *VMS Version 5.5 New Features Manual*.

1.1 VMS Version 5.5 Specific Release Notes for General Users

The release notes in this manual are cumulative from VMS Version 5.0. The following sections contain general user release notes that pertain specifically to VMS Version 5.5:

- Section 1.3.3—DELETE/ENTRY Command—Problem Corrected
- Section 1.3.4—Invalid Qualifiers Removed from Queue Commands
- Section 1.4.2.9—VAXstation 3520 and 3540 Screens Blank Out
- Section 1.5.1—CONVERT/DOCUMENT Command—New /MESSAGE_FILE Qualifier
- Section 1.5.3—DIRECTORY Command
- Section 1.5.5.2—VFC Record Format Used with OPEN Command
- Section 1.5.6—RUN (Process) Command—Modifications
- Section 1.5.10—SYNCHRONIZE Command—Restriction

1.2 Characters Changed from Nonprintable to Printable

V5.4-1

The VMS print symbiont uses a nonprintable character table to calculate the number of characters in the line. To support the internationalized printer environment, VMS Version 5.4-1 changes the following codes from nonprintable to printable:

- A4 (decimal 164)
- A6 (decimal 166)
- AC (decimal 172)
- AD (decimal 173)
- AE (decimal 174)
- AF (decimal 175)
- B4 (decimal 180)
- B8 (decimal 184)
- BE (decimal 190)

General User Release Notes

1.2 Characters Changed from Nonprintable to Printable

- D0 (decimal 208)
- DE (decimal 222)
- F0 (decimal 240)
- FE (decimal 254)

These codes are part of the DEC supplemental graphic set in the DEC Multinational Character Set (MCS) and were reserved for future standardization.

For a description of MCS and a listing of its contents, see the *Guide to Using VMS*.

1.3 Batch and Print Queuing System Notes

The release notes in this section pertain to the VMS batch and print queuing system.

1.3.1 \$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 most recently 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 next updated. 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'
```

1.3.2 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
```

1.3.3 DELETE/ENTRY Command—Problem Corrected

V5.5

In previous versions of VMS, when a user entered the DELETE/ENTRY command for an executing job in a queue set to retain jobs, the job was not deleted. It was aborted and retained.

This problem is fixed in VMS Version 5.5. The job is aborted and then deleted.

For more information about queues set to retain jobs, see the description of the /RETAIN qualifier for the INITIALIZE/QUEUE command in the *VMS DCL Dictionary*.

1.3.4 Invalid Qualifiers Removed from Queue Commands

V5.5 In VMS Version 5.4 and previous versions, when you entered an invalid qualifier with a batch or print-related command, the qualifier was ignored.

With Version 5.5, the queue manager eliminates any qualifiers that are invalid in the context of a individual command. If you enter a command with a qualifier that is invalid in a context, the queue manager returns the following message:

```
%JBC-I-ITMREMOVED, meaningless items were removed from request
```

For example, certain qualifiers for the START/QUEUE command (including /JOB_LIMIT) are invalid for paused queues. Prior to VMS Version 5.5, when you entered the START/QUEUE/JOB_LIMIT command for a paused queue, the /JOB_LIMIT qualifier was ignored. With VMS Version 5.5, when you enter the START/QUEUE/JOB_LIMIT command for a paused queue, the queue manager displays the message telling you that meaningless items were removed.

1.3.5 Print Accounting Record Page Count—Problem Corrected

V5.4-3 In previous versions of VMS, the accounting record written at the completion of a print request, in certain special cases, included the page count of both the last job printed and the one preceding it.

The following example illustrates the special case that created the incorrect accounting records.

Suppose a user prints a job named JOB1 and the queue is then stopped by the STOP/QUEUE/NEXT command. JOB1 finishes printing and its accounting record is correctly written. If another user prints a job named JOB2 to the same queue, when the queue is started and JOB2 prints, JOB2's accounting record has a page count equal to that of both JOB1 and JOB2.

This problem has been corrected.

1.3.6 SUBMIT/DELETE Command—Modifications

V5.3 In previous versions of VMS, if a user was denied delete (D) access to one or more files that were included as parameters to the SUBMIT/DELETE command, processing specified in the SUBMIT command continued. If any files specified in the SUBMIT command were successfully added to the job request (the file or files that the user had unsuccessfully designated for deletion were not included in the job request), a batch job was created and entered in the appropriate queue.

In VMS Version 5.3, if a user enters a SUBMIT/DELETE command and includes in that command's parameter list the name of a file to which the user is denied delete (D) access, the SUBMIT command processing stops and no batch job is created. This is also true if the user enters the SUBMIT command and includes the /DELETE qualifier after the name of a file to which that user is denied delete (D) access.

This same behavior happens if you enter 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.

General User Release Notes

1.3 Batch and Print Queuing System Notes

1.3.7 SUBMIT or PRINT /DELETE Qualifier—Problem Corrected

V5.4-3 In previous versions of VMS, a file submitted or printed with the /DELETE option in a VAXcluster environment was not always deleted, even if the file was accessible to the VAXcluster node on which the PRINT or SUBMIT request was executed.

This problem has been corrected. Now, a file submitted or printed with the /DELETE command is always deleted unless the file cannot be printed or executed because it cannot be accessed.

For example, suppose you enter the following command:

```
$ PRINT/DELETE FILE1 NODE_Z::PRINT_QUEUE
```

FILE1 is located on your local workstation disk. PRINT_QUEUE is located on NODE_Z. If your local disk is not available to NODE_Z, FILE1 cannot be printed and, therefore, will not be deleted.

1.4 DECwindows Notes

The release notes in this section pertain to the VMS DECwindows software supplied with VMS Version 5.5. Applications included in VMS Version 5.5 are identical to those in Version 5.4. These release notes do not apply to new applications using the Motif interface, which are available with the VMS DECwindows Motif layered product.

1.4.1 Cutting and Pasting Between XUI Windows and DECwindows Motif Windows—Restriction

V5.4-3 When running two versions of DECwindows (for example, the layered product VMS DECwindows Motif Version 1.0 from a standalone workstation and the VMS DECwindows XUI version from a cluster with VMS Version 5.4–n installed), you can cut and paste between the XUI and Motif windows on the screen. However, when you begin a session, the first cut or copy operation must be from a DECwindows XUI window.

If the first piece of information that you put on the clipboard during a cut operation comes from a VMS DECwindows Motif window, subsequent cut and paste operations with XUI windows causes the XUI application to hang and data to be lost.

1.4.2 DECterm Terminal Emulator

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

1.4.2.1 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.

1.4.2.2 DECTerm 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 might 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.

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 1-1 lists the previous and new gray levels according to color index.

Table 1-1 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

(continued on next page)

General User Release Notes

1.4 DECwindows Notes

Table 1-1 (Cont.) DECterm Gray Levels

Color Index	Previous Gray Level	New Gray Level
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.

1.4.2.3 DECterm—Default Server ID Specification Problem

V5.4-3

When you enter the DCL command `CREATE/TERM/DISPLAY=NODENAME`, you must include the server ID field of the display name specification (1 or 0) as in the following example:

```
$ CREATE/TERM/DISPLAY=NODENAME::0
```

If the server ID is not specified, then the error message "Can't open display" is displayed. This problem with the default server ID specification will be corrected in a future release.

1.4.2.4 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 systemwide 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

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:

```
$ !  
$ ! SYSS$MANAGER:SYLOGIN.COM and users' LOGIN.COM might 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 can  
$ ! 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:  
$ ENDIF
```

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 might disappear.

1.4.2.5 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.

1.4.2.6 Memory—DECterm 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 does not display an error message but ignores the resize request.

1.4.2.7 PC Interoperability Restrictions for DECterm

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

1.4 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* (order number EK-DTERM-PM-001).

1.4.2.8 DECterm Text

V5.1-1 The following information is specific to DECterm text:

- User-loadable characters (DRCS), 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.

1.4.2.9 VAXstation 3520 and 3540 Screens Blank Out

V5.5 VAXstation 3520 and 3540 screens blank out for 15 seconds after session logout. To improve system reliability, a graphics device power-up self test is executed after session manager logout. This causes the graphics display to go blank for approximately 15 seconds.

1.4.2.10 DECterm 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.

1.4.3 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 was changed for VMS Version 5.4.

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.

1.4.4 DECwindows Startup Restriction

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.

1.4.5 Desktop Applications

The following sections describe information pertaining to DECwindows Desktop Applications.

1.4.5.1 Calendar—Corrected Problems

V5.4 The following Calendar problems were 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).

1.4.5.2 Calendar—Restrictions and Limitations

V5.3 Starting 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 data file to the new format. Once converted, your database can no longer be accessed by previous versions of DECwindows Calendar.

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

General User Release Notes

1.4 DECwindows Notes

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.

1.4.5.3 Cardfiler—Scroll Bar

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

1.4.5.4 Clock—Changing the System Time

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

1.4.5.5 CDA Viewer—Corrections

V5.4 The following corrections were 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.

1.4.5.6 CDA Viewer—Problems and Unexpected Behavior

V5.4 The following problems and unexpected behavior exist:

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 might not show the results of that page. This situation occurs because, when skipping pages, the interpreter's drawing is deactivated; it does not 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.
- 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

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.

1.4.5.7 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.

General User Release Notes

1.4 DECwindows Notes

1.4.5.8 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.
- 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 can cause an access violation.
- V5.4
 - When you cannot read a PostScript file using Mail or the CDA Viewer, it is usually caused by extraneous text (such as Mail headers) preceding the %! character that must be at the beginning of any PostScript file. To solve the problem, invoke an editor and delete any text that precedes the %! character. 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.

1.4.5.9 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.

1.4.5.10 Print Screen—Corrections

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.

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 corrected this problem.

1.4.5.11 Print Screen—Problems and Unexpected Behavior

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 can cause memory overflow errors.

1.4.5.12 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.

1.4.6 Server Pseudomouse Key Combination Change

V5.4-3 The pseudomouse key combination has been changed from control F3 to Control Shift F3. If you need to change your pointer from mouse device to arrow keys, you must enter the new key combination to turn the pseudomouse mode on and enter the same key combination to turn it off.

1.4.7 Session Manager

The following sections contain information pertinent to the Session Manager.

1.4.7.1 Corrected Problems

V5.4 The following Session Manager problems have been corrected:

- 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.

1.4.7.2 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/SYSTEM/EXECUTIVE_MODE 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, loginout 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.

General User Release Notes

1.4 DECwindows Notes

1.4.8 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.

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.

1.4.9 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.

1.4.10 Window Manager—Enhancements

V5.4 You can now use the logical names listed in Table 1–2 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 1–2 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

(continued on next page)

Table 1–2 (Cont.) Session Manager Logical Names

Logical	Default Value
decw\$winmgr_enqlm	600
decw\$winmgr_jtquota	PQL value
decw\$winmgr_tqelm	PQL value
decw\$winmgr_wsquota	650
decw\$winmgr_wsdefault	512
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 loginout 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 loginout 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.

1.4.11 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.

1.4.12 Workstations with Dual-Head Support—Restrictions

V5.4-1

The following restrictions apply to workstations with dual-head support, such as VAXstation 3100 series systems. These workstations support two monitors.

Setting the DPI

If you use monitors with different dots-per-inch (dpi) settings, you might experience what appears to be window corruption but what is actually an overlapping of characters on the screen. For example, if you run an application on a workstation that has monitor 0 set to one dpi setting and monitor 1 set to a different dpi setting, the application will use fonts based on the dpi setting of monitor 0. When the application runs on monitor 1, you might perceive window corruption (overlapping characters).

To avoid this problem, Digital recommends that you set your monitors to the same dpi setting using the DECW\$PRIVATE_SERVER_SETUP.COM file. However, if you must maintain two monitors with different dpi settings, then you should run dpi/font-sensitive applications on monitor 0 only.

General User Release Notes

1.4 DECwindows Notes

Customizing Colors

When customizing colors on a dual-head system that is configured with both a color monitor and a monochrome monitor, make sure you specify colors with sufficient contrast. If the color settings do not contrast enough, then applications customized for a color display might produce black-on-black or white-on-white output when run on the monochrome monitor.

1.5 DIGITAL Command Language (DCL) Notes

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

1.5.1 CONVERT/DOCUMENT Command—New /MESSAGE_FILE Qualifier

V5.5 A new qualifier, /MESSAGE_FILE=filespec, has been added to the CONVERT /DOCUMENT command. Specify the /MESSAGE_FILE qualifier to create a message file to which messages are logged during the conversion of your document.

To take advantage of this new qualifier, you must install the DEC CDA Base Services shipping with VMS DECwindows Motif Version 1.0 or later. Prior versions of the DEC CDA Base Services are compatible with VMS Version 5.5, but do not support the new /MESSAGE_FILE qualifier.

1.5.2 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 might 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.

1.5.3 DIRECTORY Command

V5.5 When using the DIRECTORY/ACL or DIRECTORY/SECURITY command, please note that the SECURITY privilege must be turned on in order to display any ACEs that were created with the hidden option.

1.5.4 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-MSNGENDS, missing or misspelled ENDSUBROUTINE statement detected
while scanning for label
```

1.5.5 OPEN Command

The release notes in this section pertain to the OPEN command.

1.5.5.1 Negating Problem Corrected

V5.4

In previous versions of VMS, if you negated a qualifier while 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; now the qualifiers to the OPEN command cannot be negated. If you negate a qualifier using the DCL command OPEN, the following error message is displayed:

```
%DLC-W NOTNEG, qualifier or keyword not negatable - remove "NO" or omit
\NOSHARE\
```

1.5.5.2 VFC Record Format Used

V5.5

When the OPEN command is used to create a new file, variable fixed control (VFC) record format is used. Concatenating a file of this record format with a file of another record format might be impossible due to record format incompatibilities. To avoid the VFC format, use the CREATE command to create the file.

When the OPEN command is specified on an existing file, the record type of that file is used.

1.5.6 RUN (Process) Command—Modifications

V5.5

The description of the RUN (Process) command in the *VMS DCL Dictionary* states that a detached process is created if you specify the /UIC qualifier.

The RUN command creates a detached process if you specify either the /UIC or the /DETACHED qualifier.

The following changes affect the RUN/[NO]AUTHORIZE command:

- When you specify the /NOAUTHORIZE qualifier, quotas are derived from the SYSGEN parameters that set process quota default limits (parameters prefixed by PQL_D).
- When you specify the /AUTHORIZE qualifier, quotas are derived from the user authorization file (UAF) record of the process' owner. Any qualifiers to the RUN command that specify other quotas are ignored in favor of these UAF-based quotas.

1.5.7 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.

General User Release Notes

1.5 DIGITAL Command Language (DCL) Notes

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 let you select all configuration characteristics, such as XON/XOFF flow control, maximum number of buffers, read delay, and parity.
- A new interactive command mode lets you 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*.

1.5.8 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.

1.5.9 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.

1.5.10 SYNCHRONIZE Command—Restriction

V5.4-2 The SYNCHRONIZE command provides job synchronization by placing a process in wait state until the specified job completes. If the specified job completes with a returning status with the low 16-bits as zeros, the process waiting on the SYNCHRONIZE command will not be released from the wait state.

This restriction will be removed in a future release.

1.6 DUMP Command on KZQSA Tape Drives

V5.4-2 Simultaneous DUMP operations on two or more tape drives connected to a KZQSA controller are unsupported and can cause system hangs.

1.7 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)

1.8 F\$CONTEXT Lexical Function Problem

V5.4-2 The DCL lexical function F\$CONTEXT can cause the deletion of a process unless you use the function in one of the following ways:

- Specify CANCEL as the third parameter of F\$CONTEXT.
- Specify all parameters of F\$CONTEXT.

This problem will be addressed in a future release of the VMS operating system.

1.9 Forced Password Changes at Login

V5.4 Beginning with VMS Version 5.4, the password change logic in LOGINOUT was modified. Users are no longer prompted for their old password during a forced password change at login since the old password was used to enter the system.

A password change at any other time does prompt for the old password.

1.10 Image Data Verification in Batch Mode—Problem Corrected

V5.4-1 Prior to VMS Version 5.4-1, a login command procedure containing the DCL command SET NOVERIFY (specifying that image data records were not to be echoed, or verified) did not work properly when the job was running in batch mode. VMS did not propagate the setting to the command procedure submitted, causing the image data records to be echoed rather than suppressing the output as directed. VMS Version 5.4-1 corrected this problem.

1.11 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.

1.12 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. 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
```

Starting 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 precede the process name. For example, to reference the process BATCH_69 on node NNAME, use the name NNAME::BATCH_69.

General User Release Notes

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

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.

1.13 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.

1.14 TLZ04 Tape Drive Performance Considerations

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

1.15 VAXstation 3100, Model 76 Computer Release Notes

V5.4-1

The following release note applies to the VAXstation 3100, Model 76 system.

The mouse included with the VAXstation 3100, Model 76 system uses two “feet” (one for moving horizontally and one for moving vertically) instead of a ball. For optimal performance with this type of mouse, choose the Pointer... option from the Session Manager Customize menu. Then, set the mouse speed (Pointer Acceleration) in the Customize Pointer dialog box to Fast instead of Medium (which is the default setting for a mouse with the ball design).

Mouse performance, including improved diagonal movement, will be addressed in a future version of DECwindows software. Mouse performance on systems running VMS Workstation Software (VWS) is unaffected.

1.16 VAX Text Processing Utility (VAXTPU) Notes

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

1.16.1 /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.

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 the VMS operating system.

1.16.2 /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 the VMS operating system.

1.16.3 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.

1.17 VMS Mail Utility

The release notes in this section pertain to the VMS Mail Utility.

1.17.1 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 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 enclosed the folder name in quotation marks, it was left in mixed case. Version 5.1 restored the Version 4.x capability and supports mixed case 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.

1.17.2 Multiple Copies of a Message Delivered to the Same Recipient—Problem Corrected

V5.4-1 Prior to VMS Version 5.4-1, the VMS Mail Utility delivered multiple copies of a message to the same recipient when two conditions were present:

1. The sender used a distribution list with ULTRIX mail or some other mail agent that retries on errors.
2. A recipient could not receive mail for some unusual reason, for example, if the recipient had mail forwarded and exceeded disk quota.

The VMS Mail Utility reported nondelivery to the recipient and to some other recipient for whom the delivery was successful. On retrying the delivery, the sending mail agent incorrectly delivered the message to someone who had already received it.

Starting with VMS Version 5.4-1, the Mail Utility correctly reports nondelivery to the sender.

General User Release Notes

1.17 VMS Mail Utility

1.17.3 Use Quotes in Address When Forwarding Mail to ULTRIX Users

V5.4-1

The VMS Mail Utility requires addresses, including forwarding addresses, to be stored in uppercase letters. However, non-VMS mail systems, such as ULTRIX, may require addresses to contain lowercase characters. Prior to VMS Version 5.0, the VMS Mail Utility allowed addresses containing lowercase characters to be stored in a user profile and correctly passed them to the non-VMS mail systems. From VMS Version 5.0 through Version 5.4, the VMS Mail Utility converted all addresses to uppercase before storing them in a user profile. Beginning with VMS Version 5.4-1, the VMS Mail Utility again stores addresses containing lowercase addresses in a user profile, provided you use one of the following methods to set a forwarding address:

- If the initial part of the address, such as the node name, is in uppercase and the rest of the address is in lowercase, preserve the lowercase part of the address by enclosing it in three sets of quotation marks, as in the following example:

```
MAIL> SET FORWARD NODE::" "name" " "
```

- To preserve the entire address in lowercase, enclose the entire address in five sets of quotation marks, as in the following example:

```
MAIL> SET FORWARD " " " "node::name" " " " "
```

Note that you cannot use the lowercase format for forwarding mail to VMS users.

System Manager Release Notes

This chapter contains information about VMS Version 5.5 that is of interest to the system manager.

For information about the new features included in VMS Version 5.5, see the *VMS Version 5.5 New Features Manual*.

2.1 VMS Version 5.5 Specific Release Notes for System Managers

The release notes in this manual are cumulative from VMS Version 5.0. The following sections contain system management release notes that pertain specifically to VMS Version 5.5:

- Section 2.5.1—OPER Privilege Requirement
- Section 2.7.1—Supported Versions for the VMS Batch and Print Queuing System
- Section 2.7.2—DECdtm Files Required for the Batch and Print Queuing System
- Section 2.7.3—Deleting the DEFAULT Print Form
- Section 2.7.4—Changing Print Forms and Characteristics
- Section 2.7.5—Generic Queue Restriction Lifted
- Section 2.7.6—Job Scheduling Priority
- Section 2.7.8—Queue Manager Notes
- Section 2.7.9—SHOW QUEUE and SHOW ENTRY Displays
- Section 2.7.10—STOP/QUEUE/RESET Command
- Section 2.7.11—SYSGEN Parameters
- Section 2.7.13—Using Banner Pages for Jobs Printed from MAIL—Change in Behavior
- Section 2.7.14—VAX Distributed Queuing System (DQS)—Recommended Version
- Section 2.8—Buffered I/O Byte Limit (BYTLM)—Increase Required
- Section 2.9—COBRTL Separate Installation Requirement Removed
- Section 2.12.1—DECwindows Server Logical Switch for Server Connect
- Section 2.21—Multiple Backups Now Supported
- Section 2.22—LAT Release Notes
- Section 2.26—SCSI Notes
- Section 2.27.1—COMMSYNC Qualifier to the SET TERMINAL Command

System Manager Release Notes

2.1 VMS Version 5.5 Specific Release Notes for System Managers

- Section 2.27.3—Modifying the System Password Dictionary
- Section 2.30—Standalone BACKUP Correction
- Section 2.32.1—Startup Procedure Template Changes
- Section 2.32.2—SHUTDOWN.COM—Changes for Autostart Queues
- Section 2.35—System Disk Size Recommendation: 115 MB
- Section 2.40—Upgrade Notes
- Section 2.41—VAXft SYSTEMS—EFDRIVER OPCOM Messages
- Section 2.43.1—Naming VAX 6000 Series Console Tapes and Tape Serving Devices
- Section 2.43.4—DSSI and FDDI Boot Devices Supported on VAX 6000-Series Systems
- Section 2.43.4.1—Booting Standalone BACKUP on a VAX 6000-Series Computer from an InfoServer System
- Section 2.46—VAX Computers—VMS Support
- Section 2.48—VAXstation 4000 Series Computer—Changing Font Size
- Section 2.49—VMS License Management Facility Notes
- Section 2.50.1—Large VAXcluster Configurations—SYSGEN Parameter SCSCONNCNT
- Section 2.50.3—Selecting Boot Disk Path
- Section 2.50.4—SCS SYSGEN /TIMVCFAIL Parameter
- Section 2.50.8—VAXcluster Use of DEMNA
- Section 2.51.5—MOUNT/CLUSTER Command

2.2 Automatic Attempt to Load Continuation Volume of Magnetic Tape

V5.4-1

When you use MOUNT/AUTOMATIC to mount labeled tape volume sets on multiple drives, the magnetic tape ancillary control process (MTAACP) expects continuation volumes to be loaded on the proper drives and attempts to mount them without requesting operator assistance.

Prior to VMS Version 5.4-1, the MTAACP required operator intervention to switch to the next continuation volume if only one magnetic tape drive was allocated to the volume set. This requirement reduced the effectiveness of the TA90 cartridge loaders, which can automatically switch volumes on a single drive.

Starting with VMS Version 5.4-1, when you use the MOUNT/AUTOMATIC command to mount tapes on a single drive, the MTAACP attempts to load the continuation volume for 100 seconds before it requires operator assistance.

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

2.3 \$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, causing each node to use a different rights database.

Starting 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.)

2.4 Authorize Utility Notes

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

2.4.1 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 about the ADD/PROXY and MODIFY/PROXY commands, see the *VMS Authorize Utility Manual*.

2.4.2 Minimum Lengths for Generated Passwords

V5.4-3 For consistency with the SET PASSWORD command, the AUTHORIZE Utility now only permits generated passwords with minimum lengths of 1 to 10 characters. If an account has a password minimum length of more than 10 characters, it is minimized to a character length of 10 for generated passwords.

In the case of nongenerated passwords, the password minimum length of the account is used.

2.4.3 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.

System Manager Release Notes

2.4 Authorize Utility Notes

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.

2.4.4 Setting the DISCTLY Flag

V5.3-1 Due to an error in VMS Version 5.2, the login procedure did not set the DISCTLY 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 DISCTLY flag for all accounts that are marked either CAPTIVE or RESTRICTED.

2.4.5 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 were removed in VMS Version 5.3-2. With the restrictions removed, network server objects now function properly with the CAPTIVE flag set.

Digital now recommends that you use the MODIFY/NOBATCH and MODIFY/NONETWORK commands in AUTHORIZE to disable BATCH and NETWORK access in captive (or restricted) accounts.

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

2.5 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 5.5 New Features Manual*.

2.5.1 OPER Privilege Requirement

V5.5 With VMS Version 5.5, AUTOGEN uses the System Management Utility (SYSMAN) when changing system parameters. For this reason, you now need the OPER privilege to run the GENPARAMS and SETPARAMS phases of AUTOGEN in addition to the privileges you already need to run AUTOGEN. Digital recommends you run AUTOGEN from the system manager's account (SYSTEM) to be sure you have the required privileges. For more information on AUTOGEN, see the *Guide to Setting Up a VMS System*.

2.5.2 Switching Window Systems

V5.2

With VMS Version 5.2 and subsequent versions, if you want to switch from VWS to VMS 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.

To switch from one window system to another, follow these steps:

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. Shutdown and 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 systems more than once, save copies of your system parameter files (SYS\$SYSTEM:VAXVMSSYS.PAR) for both DECwindows and VWS. You can subsequently change the 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).

For information about performing a conversational boot refer to *Guide to Setting Up a VMS System*.

2.6 Backup Utility Notes

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

2.6.1 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.

2.6.2 BACKUP ACL Behavior—Change

V5.4-3 In previous VMS versions, a problem occurred when a nonprivileged user attempted to restore a file that had an Access Control List (ACL) and was owned by a resource ID to a directory in which the user had read and write privileges.

With VMS Version 5.4-3, when a file is restored from a BACKUP save set, it possesses only the Access Control Entries (ACE) that were in the file's ACL when it was saved. BACKUP no longer attempts to merge a propagated directory ACE into the restored file's ACL. To change the restored file's ACL to make it consistent with other files in the directory, use the DCL command SET ACL or the ACL editor.

Any BACKUP images from Digital Services that merged a directory ACE into the restored file's ACL are obsolete and are superseded by VMS Version 5.4-3.

2.6.3 BACKUP/RECORD Copy Operations Problem

V5.2 With VMS Version 5.2 and higher, there is a problem with disk-to-disk BACKUP/RECORD copy operations. If a disk-to-disk BACKUP/RECORD copy fails (for example, if the output disk runs out of free blocks or if the operation exceeds the disk quota), BACKUP continues to process the remaining input files instead of aborting. As a result, BACKUP records the current date and time in the BACKUP date field of each input file, even if the file was not successfully copied.

2.6.4 Backup Utility—Problem Corrected

V5.4 The following BACKUP problem existed in VMS Version 5.4. This problem 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.

2.6.5 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.)

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.

2.6.6 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[...]
```

2.7 Batch and Print Queuing System Notes

The release notes in this section pertain to the batch and print queuing system.

2.7.1 Supported Versions for the VMS Batch and Print Queuing System

V5.5 VMS Version 5.5 provides a new batch and print queuing system. In a VAXcluster environment, the new queuing system requires that all nodes in a cluster run VMS Version 5.5 (V5.5). If you cannot upgrade all your cluster nodes to V5.5, you cannot run the new queuing system on any nodes in the cluster. However, you can begin the upgrade path to V5.5 and stop at the intermediate version A5.5. This allows you to run a queuing system that is compatible with that of VMS Version 5.4 until you can upgrade all your VAXcluster nodes to V5.5. For more information, see the *VMS Version 5.5 Upgrade and Installation Manual*.

While a node or VAXcluster is running the intermediate version A5.5, the following new batch and print queuing system features are not available:

- Autostart queues:
 - The ENABLE AUTOSTART command
 - The DISABLE AUTOSTART command
 - The /AUTOSTART_ON qualifier for the INITIALIZE/QUEUE and START/QUEUE commands
- Clusterwide queue manager
 - Queue Manager changes described in Section 2.7.8
 - The STOP/QUEUE/MANAGER/CLUSTER command
 - The STOP/QUEUES/ON_NODE command
 - New queue database format
 - Queue manager automatic start
 - Queue manager failover
- Changes to the \$GETQUI and \$SNDJBC system services listed in the *VMS Version 5.5 New Features Manual*.
- The /RETAIN qualifier for the SUBMIT and PRINT commands

System Manager Release Notes

2.7 Batch and Print Queuing System Notes

- The stalled job state
- The **jobname** parameter to the SHOW ENTRY command
- The /NOTE qualifier for the SUBMIT command
- Changes to the F\$GETQUI lexical function described in the *VMS Version 5.5 New Features Manual*.

The following table explains where to find information about the batch and print queuing system for each VMS version:

Version	Manuals
VMS Version 5.4 or A5.5	VMS Version 5.0 <i>Guide to Maintaining a VMS System</i> VMS Version 5.4 <i>VMS DCL Dictionary</i> VMS Version 5.0 <i>VMS System Services Reference Manual</i>
VMS Version 5.5 (V5.5)	VMS Version 5.5 <i>Guide to Maintaining a VMS System</i> VMS Version 5.5 <i>New Features Manual</i> VMS Version 5.5 DCL Help VMS Version 5.5 <i>VMS System Services Reference Manual</i>

2.7.2 DECdtm Files Required for the Batch and Print Queuing System

V5.5 The VMS Version 5.5 batch and print queuing system requires the following DECdtm files:

- SYS\$LOADABLE_IMAGES:SYS\$TRANSACTION_SERVICES.EXE
- SYS\$SHARE:IPC\$SHARE.EXE
- SYS\$SYSTEM:IPCACP.EXE
- SYS\$STARTUP:IPC\$STARTUP.COM

If you plan to use the new queuing system, do not delete these files.

2.7.3 Deleting the DEFAULT Print Form

V5.5 With previous VMS versions, system managers could delete the systemwide default print form defined by the system. This form, named DEFAULT, corresponds to the form number 0. A queue initialized without a specific form uses the DEFAULT form to process print jobs not explicitly associated with a form definition.

With VMS Version 5.5, when you attempt to delete the DEFAULT form, you get the following error message:

```
%DELETE-E-NOTDELETED, error deleting DEFAULT-JBC-E-REFERENCED,  
existing references prevent deletion
```

For more information about default print forms, see the *Guide to Maintaining a VMS System*.

2.7.4 Changing Print Forms and Characteristics

V5.5

In previous VMS versions, you could assign a new number to an existing characteristic name by re-entering the DCL command DEFINE/CHARACTERISTIC. Likewise, you could assign a new number to an existing form name by re-entering the DCL command DEFINE/FORM. With the previous queuing system, you could also modify values for any DEFINE/FORM qualifier by re-entering the DCL command DEFINE/FORM with different qualifier values. However, when making these changes, results were unpredictable.

With VMS Version 5.5, to modify a characteristic's name or number or a form's name or number, you must delete and redefine the characteristic or form. Values for any DEFINE/FORM qualifier can still be modified by re-entering the DEFINE/FORM command with different values, as long as the form name and number remain the same.

You can modify the stock of a form by specifying the DEFINE/FORM command with the /STOCK qualifier unless references to the form exist. For information on removing references to a form, see the *Guide to Maintaining a VMS System*.

For information about defining forms and characteristics, see the DEFINE/FORM and DEFINE/CHARACTERISTIC commands in the *VMS DCL Dictionary*. For information on deleting forms and characteristics, see the DELETE/FORM and DELETE/CHARACTERISTIC commands in the *VMS DCL Dictionary*.

2.7.5 Generic Queue Restriction Lifted

V5.5

With VMS Version 5.0–2, no more than eight generic queues could target an execution queue. With VMS Version 5.5, there is no limit to the number of generic queues that can feed an execution queue.

2.7.6 Job Scheduling Priority

V5.5

In VMS Version 5.4, if a user did not specify a scheduling priority for a job, the job's scheduling priority was set to the value of the SYSGEN parameter DEFQUEPRI. If that value was zero, the job scheduling priority was set to the process priority of the user submitting the job.

However, job scheduling priority and process priority are unrelated. Process priority has a range from 1 to 31, while job scheduling priority has a range from 0 to 255. For this reason, with VMS Version 5.5, process priority is no longer used to define job scheduling priority. If a user does not specify a scheduling priority for a job, the job's scheduling priority is set to the value of the SYSGEN parameter DEFQUEPRI. If that value is zero, the job scheduling priority is set to zero. The default value of DEFQUEPRI is 100.

Without ALTPRI or OPER privilege, the user cannot specify a job scheduling priority that is greater than the queue's maximum scheduling priority. The queue's maximum scheduling priority is set to the greater value of the following two SYSGEN parameters:

- DEFQUEPRI
- MAXQUEPRI

System Manager Release Notes

2.7 Batch and Print Queuing System Notes

2.7.7 Print Symbiont Working Set Purge Less Frequent

V5.2 In previous versions of VMS, the print symbiont purged its working set when it finished processing a file if no other supported queue 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 and subsequent versions, 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.

2.7.8 Queue Manager Notes

The release notes in this section pertain to the queue manager.

2.7.8.1 Change in Behavior

V5.5 With previous VMS versions, if the queue manager is running and you attempt to start it (either by entering the DCL command `START/QUEUE/MANAGER` or by issuing a `$$SNDJBC` request with the `SJC$_START_QUEUE_MANAGER` function code), the following error code is returned:

```
JBC-E-JOBQUEENA, system job queue manager is already running
```

In VMS Version 5.5, if the queue manager is running and you attempt to start it, a success status is returned. If you provide any new parameters with the `START/QUEUE/MANAGER` command or `SJC$_START_QUEUE_MANAGER` request, the queue manager process is changed to reflect them.

2.7.8.2 Change in the /NEW_VERSION Qualifier

V5.5 In VMS versions prior to Version 5.5, the `/NEW_VERSION` qualifier of the `START/QUEUE/MANAGER` command created a queue file which *superseded* any previous versions of the file. With VMS Version 5.5, the `/NEW_VERSION` qualifier supersedes only queue configuration information in the queue database. Job information is *replaced*.

Caution

Do not use the `/NEW_VERSION` qualifier with the `START/QUEUE/MANAGER` command unless no database exists or you no longer need the existing queue database.

2.7.8.3 Mounting the Queue Database Disk

V5.5 With VMS Version 5.4, the queue manager is started by the DCL command `START/QUEUE/MANAGER` in `SYSTARTUP_V5.COM`. With the new queuing system, you need enter the `START/QUEUE/MANAGER` command only once. Thereafter, the command is stored in the queue database and is used by the job controller to start the queue manager automatically during reboot.

With VMS Version 5.5, if the master queue file `QMAN$MASTER.DAT` is not located on the system disk, the disk on which it is located should be mounted in the startup command procedure `SYLOGICALS.COM`. Although `SYLOGICALS` is normally used to define logical names, it is important that the disk holding the master file be mounted in the `SYLOGICALS` startup procedure, so it is available before the job controller starts.

System Manager Release Notes

2.7 Batch and Print Queuing System Notes

For more information about the new queue database files, see the *VMS Version 5.5 New Features Manual*. For information on mounting a disk in SYLOGICALS.COM, see the section Coordinating Shared System Files with Multiple Common System Disks in the *VMS VAXcluster Manual*.

2.7.8.4 Obsolete Qualifiers

V5.5 The /EXTEND, /BUFFER_COUNT, and /RESTART qualifiers to the DCL command START/QUEUE/MANAGER are obsolete with VMS Version 5.5.

2.7.9 SHOW QUEUE and SHOW ENTRY Displays

V5.5 In VMS Version 5.5, the output from the SHOW QUEUE and SHOW ENTRY commands have changed. Most notably, the entry number column now appears first. Also, new queue states are now reported. If you have a command procedure that parses the output of the SHOW QUEUE or SHOW ENTRY command, the new output can cause the procedure to break.

As a workaround, Version 5.5 includes an alternate image, QUEMAN_OLD.EXE, which produces the format used in earlier versions. If you require the old format, you can assign the string SYS\$SYSTEM:QUEMAN_OLD.EXE to the logical name QUEMAN as follows to specify that the alternate image be used instead of the default image:

```
$ ASSIGN SYS$SYSTEM:QUEMAN_OLD.EXE QUEMAN
```

Include the /PROCESS, /JOB, /GROUP, or /SYSTEM qualifier to specify the scope of the logical name, depending on the group of users you want to affect. For more information on these qualifiers, see the *VMS DCL Dictionary*.

Digital recommends that you begin to change any command procedures that parse the output of the SHOW QUEUE command or SHOW ENTRY command to use the F\$GETQUI lexical function described in the *VMS DCL Dictionary*. The alternate image QUEMAN_OLD.EXE might not be included in future releases.

For information on the changes to the SHOW QUEUE and SHOW ENTRY displays, see the *VMS Version 5.5 New Features Manual*.

2.7.10 STOP/QUEUE/RESET Command

V5.5 Under certain conditions, executing the START/QUEUE command within one minute of executing the STOP/QUEUE/RESET command can cause a symbiont process and all queues serviced by the symbiont to stop. When this happens, the following messages are written to the operator log:

```
%%%%%%%%%% OPCOM 20-AUG-1991 09:53:29.20 %%%%%%%%%%%
Message from user QUEUE_MANAGE on VMSPRT
%QMAN-E-SYMDEL, unexpected symbiont process termination

%%%%%%%%%% OPCOM 20-AUG-1991 09:53:29.21 %%%%%%%%%%%
Message from user QUEUE_MANAGE on VMSPRT
-PSM-F-BADLOGIC, internal logic error detected at PC 00000000
```

To avoid this problem, if you enter a STOP/QUEUE/RESET command, wait at least one minute before entering the START/QUEUE command.

System Manager Release Notes

2.7 Batch and Print Queuing System Notes

2.7.11 SYSGEN Parameters for Queuing

V5.5 In previous VMS versions, the following SYSGEN parameters used to set queuing system defaults were taken from the node on which the queue was defined:

- DEFQUEPRI
- DEFPRI
- MAXQUEPRI

The Version 5.5 queuing system provides a single queue manager to serve processes on all nodes in the cluster. The SYSGEN parameter values used to set queuing system defaults are taken from the node on which the queue manager process is running.

Because the queue manager process can fail over from one node to another, Digital recommends you set these parameters to the same values on all nodes in a cluster.

2.7.12 SET TIME/CLUSTER Command Synchronizes Cluster Time

V5.0 In a VAXcluster environment, a batch job submitted to execute at a specified time might 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` might execute at 23:58 relative to the host system's clock.

This problem can occur in a cluster even if a job is run on the same machine from which it was submitted. The queue manager process will schedule the job when the node on which the queue manager is running reaches the specified time. 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) might 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.

2.7.13 Using Banner Pages for Jobs Printed from MAIL—Change in Behavior

V5.5 Prior to Version 5.5, system managers could take advantage of the `SET QUEUE /DEFAULT` command to specify flag and trailer page settings on generic print queues. The `SHOW QUEUE` command would not display `/DEFAULT` settings for generic queues, but these settings were stored in `JBCSYSQUE.DAT`, the job controller's job and queue data file. When the `MAIL` image prepared to send a job to a queue in response to the `PRINT` command specified at the `MAIL>` prompt, the image would get information about the queue to determine which default flag and trailer options should be specified for the print job.

In VMS Version 5.5, the `SET QUEUE /DEFAULT` command does not affect generic queues, and an informational message will be returned to the user stating that meaningless items were removed from the request. In order to allow default flag and trailer pages to still be specified with print jobs sent from `MAIL` to a generic queue, the following changes were made:

System Manager Release Notes

2.7 Batch and Print Queuing System Notes

1. If a target list is specified with the /GENERIC qualifier when the generic queue is created or started, the flag and trailer defaults specified for the first queue in the target list will be used to determine the flag and trailer defaults used for MAIL's print jobs sent to the generic queue.
2. If no target queues are specified for the generic queue, defaults of one flag page and no trailer page will be used for each print job sent from MAIL to the generic queue.

2.7.14 VAX Distributed Queuing System (DQS)—Recommended Version

V5.5 Changes to the batch and print queuing system have required changes in the VAX Distributed Queuing System (DQS). If you use DQS, it is recommended that you upgrade to DQS V1.2 when you upgrade the VMS operating system.

2.8 Buffered I/O Byte Limit (BYTLM)—Increase Required

V5.5 With VMS Version 5.5, more system dynamic memory is needed to run processes. To make sure there is enough memory, increase the buffered I/O limit (BYTLM) for each user account by 320.

Use the MODIFY command of the Authorize Utility to change the BYTLM value.

2.9 COBRTL Separate Installation Requirement Removed

V5.5 VMS Version 5.5 includes execution support for programs compiled with the VAX COBOL Version 4.4 compiler. This release of the compiler improves the speed of some compute intensive programs when executed on a VAX computer which emulates decimal instructions. The COBRTL.EXE included with this release incorporates bug fixes not available in previous versions. The COBRTL shipped with the compiler no longer needs to be installed separately for programs compiled with that version of the compiler to execute. Programs compiled with older versions of the compiler continue to execute correctly with the latest COBRTL.EXE.

2.10 Debugger Notes

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

2.10.1 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. Section 2.10.2 describes the resources used by the debugger, so that you can tune your system for this activity.

System Manager Release Notes

2.10 Debugger Notes

Note that the information in this section pertains only to the resources used by the debugger. In the case of multiprocess programs, you might also need to tune the system to support the programs themselves.

2.10.2 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 might 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.

2.10.3 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 might need to increase each user's PGFLQUOTA. If a user has an insufficient PGFLQUOTA, the debugger might fail to activate or can cause "virtual memory exceeded" errors during execution.
- You might 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.

2.11 DECnet-VAX Notes

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

For additional DECnet-VAX information, see the following notes:

- Ethernet Notes (Section 2.18)
- NETCONFIG.COM Security Enhancements (Section 2.27.4)
- Security Review Recommended (Section 2.28)
- DECnet—File Access Protocol Extensions (Section 3.8)

2.11.1 DECnet Links Dropping—Problem Corrected

V5.4-3 Prior to VMS Version 5.4-3, a problem with NETDRIVER caused links to drop immediately after a circuit went down. VMS Version 5.4-3 corrects this problem.

2.11.2 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.

2.11.3 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 SYS\$MANAGER:SYSTARTUP_V5.COM.

2.11.4 NETDRIVER Acknowledges Delay Time

V5.4-3 Prior to VMS Version 5.4-3, on a link that allowed delayed acknowledgment, NETDRIVER did not include the acknowledgment delay time in the determination of the retransmission timeout period. This allowed unnecessary timeouts and retransmissions.

In VMS Version 5.4-3, NETDRIVER includes an acknowledgment delay time value of three seconds in the determination of the retransmission timeout period.

2.11.5 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.

2.11.6 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 corrected this error in node-level access control for outgoing connections.

2.11.7 Startup Restriction with Ethernet-Based Applications in a Local Area VAXcluster

V5.4-3 Prior to VMS 5.4-3, on a system that participated in a local area VAXcluster, you could start Ethernet-based applications on the system before you started DECnet. For example, you could start the Local Area Transport (LAT) software before starting DECnet.

In VMS Version 5.4-3, you must first start DECnet on any Ethernet device that DECnet uses before you start any Ethernet-based applications that use the Ethernet device.

Before you update to VMS Version 5.4-3, if DECnet and LAT share the same Local Area Network (LAN) device, make certain that your system startup procedure does one of the following:

- Starts LAT after DECnet is started.
- Uses the CREATE/LINK /DECNET command to start LAT. This command causes LAT to use the DECnet address instead of the LAN hardware address.

System Manager Release Notes

2.11 DECnet-VAX Notes

2.11.8 SYS\$CLUSTER_NODE Logical Name—Correction

V5.4-1 In previous releases of VMS, DECnet-VAX did not deassign the SYS\$CLUSTER_NODE logical name when the executor was set to the OFF state. Starting with VMS Version 5.4-1, DECnet-VAX deassigns the SYS\$CLUSTER_NODE logical name whenever the executor is set to the OFF state. The SYS\$CLUSTER_NODE logical name is defined when you start the DECnet-VAX software, and the ALIAS NODE characteristic is defined for the executor node.

2.11.9 UNA Circuit Default Cost Changed

V5.4-1 VMS Version 5.4-1 changes the default cost of a UNA circuit from 3 to 4. This change makes the default costs of all Ethernet circuit types equal.

2.12 DECwindows Notes

The release notes in this section pertain to the VMS DECwindows software supplied with VMS Version 5.5. Applications included in VMS Version 5.5 are identical to those in Version 5.4. These release notes do not apply to new applications using the Motif interface, which are available with the VMS DECwindows Motif layered product.

2.12.1 DECwindows Server Logical Switch for Server Connect/Disconnect Messages

V5.5 The symbol DECW\$SERVER_CONNECT_LOG has been added to the SYS\$MANAGER:DECW\$PRIVATE_SERVER_SETUP.TEMPLATE file. This symbol determines which client connect/disconnect messages are logged to the server error log file.

The DECW\$SERVER_CONNECT_LOG symbol has a default value of "F", which means that successful connect/disconnect are not logged. This results in a moderate performance increase.

To enable logging of successful connect/disconnect messages, change the default value of the DECW\$SERVER_CONNECT_LOG symbol to "T" and reset your server.

2.12.2 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.

2.12.3 Starting the ULTRIX Connection (UCX) Before DECnet—Problem

V5.3 The UCX startup command file must be executed after the DECnet startup command file has completed. If UCX is started before DECnet, DECnet does not work properly.

The file SYS\$MANAGER:SYSTARTUP_V5.TEMPLATE contains 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
```

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

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

2.12.4 Tailoring DECwindows

V5.2 If you have a small system disk (RD53) and you tailor off the DECwindows files, you might 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 adequate free space. Tailoring does not check that there is sufficient space for the selected files.

2.12.5 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.

2.12.6 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.

2.12.7 X Servers—Interoperability Restriction

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.

System Manager Release Notes

2.12 DECwindows Notes

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.

2.13 Disk Quota Cache Entries—Maximum Value

V5.4-3 The following command enables disk quota caching and allows you to specify the number of quota cache entries for a quota-enabled disk:

```
$ MOUNT/CACHE=(QUOTA=n)
```

Please note that the current maximum value for the SYSGEN parameter (n) ACP_QUOCACHE is 2337.

2.14 Disk Header Space Problem

V5.4 Large-capacity disks (such as RA90s) can 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 allows the INDEXF.SYS file to be extended for approximately 50,000 file extents only. 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 perform an image backup and restore operation using the BACKUP/IMAGE command to compress the data on the disk.

To determine whether a 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
```

System Manager Release Notes 2.14 Disk Header Space Problem

A sample portion of the output is as follows:

```
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 might 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, perform an image backup and restore operation using the BACKUP/IMAGE command to compress the data on the disk. It is important that you use the /INITIALIZE qualifier when you restore the image backup 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/INITIALIZE 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

2.15 DISMOUNT Command—Changes Regarding Open Files

2.15 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 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 2.15.1).

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:

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

System Manager Release Notes

2.15 DISMOUNT Command—Changes Regarding Open Files

2.15.1 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 might 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:SYSPAGSWPFILES.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 INSTALL command LIST. You can then cancel the installed status of the files, as described in the *VMS Install Utility Manual*.

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

System Manager Release Notes

2.15 DISMOUNT Command—Changes Regarding Open Files

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.

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

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

2.15.3 Restoring the Previous Behavior of the DISMOUNT Command

V5.2

In some cases you might 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

System Manager Release Notes

2.15 DISMOUNT Command—Changes Regarding Open Files

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\$_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 was modified in VMS Version 5.2 to specify the /OVERRIDE=CHECKS qualifier when dismounting volumes.

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

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

The following informational message is displayed, and the device is dismounted:

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

2.16 DNS RTL Routines DNS\$PARSE_USERNAME_STRING and DNS\$CVT_TO_USERNAME_STRING—Corrections

V5.4-1

In previous versions of VMS, a problem in the DNS Run-Time Library (RTL) routines caused generation of an incorrect opaque full name for client applications that used the DNS RTL routines DNS\$PARSE_USERNAME_STRING and DNS\$CVT_TO_USERNAME_STRING. DNS client applications can use these routines to manipulate principals as access control entries of objects in the namespace or to manipulate members of a group object. The problem caused DNS\$PARSE_USERNAME_STRING, when given the DECnet Phase IV format (*node::user*), to give an incorrect full name. The DNS server expects a correct opaque full name, so the DNS server rejected a DNS client application that used DNS\$PARSE_USERNAME_STRING. The problem also caused DNS\$CVT_TO_USERNAME_STRING to convert the incorrect opaque full name back to the DECnet Phase IV format (*node::user*).

For objects showing this problem, use the DNS client application with the DNS\$RTL image installed with VMS Version 5.4-1 to add the attribute value again.

2.17 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.

System Manager Release Notes

2.17 DSSI Device Naming No Longer Dependent on SYSGEN Parameter VMS5

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 scheme used for DSSI disks attached to embedded adapters, such as those on MicroVAX 3300/3400 systems.

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 2-1 for ports or disks, or both.

Table 2-1 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).

2.17 DSSI Device Naming No Longer Dependent on SYSGEN Parameter VMS5

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.

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 about the console command SET HOST/DUP, see the section "Configuring RF30 and RF71 Devices in a VAXcluster" in the *VMS Upgrade and Installation Supplement: VAX 4000 Series and MicroVAX, VAXstation, and VAXserver 3200, 3300/3400, 3500/3600, 3800/3900 Series*, or the hardware information for your system.

2.18 Ethernet Notes

The release notes in this section pertain to the Ethernet.

2.18.1 DEBNI Ethernet/802 Controller—New Support for the VAXBI Bus

The notes in this section pertain to the DEBNI Ethernet controller.

2.18.1.1 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.

System Manager Release Notes

2.18 Ethernet Notes

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*cu*

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.

2.18.2 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.

2.18.3 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:

XQ*cu*

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.

2.18.4 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.

2.19 Hierarchical Storage Controller (HSC) Revision Levels Required

V5.4-1 Starting with VMS Version 5.4-1, the following Hierarchical Storage Controller (HSC) microcode revision levels are required for all HSC software:

- V500 for HSC40 and HSC70
- V400 for HSC50

2.20 ISL Support for VAX Systems—Extended

V5.4-3 Beginning with Version 5.4-3, VMS includes InfoServer System Load (ISL) support for the VAX systems listed in Table 2-2.

Table 2-2 ISL-Capable Systems

System	Ethernet Controller
Small Systems (ISL_SVAX_U3054.SYS)	
MicroVAX II	DELQA (XQ)
MicroVAX III	LANCE (ES)
VAX 3200	LANCE (ES)
VAX 3500	LANCE (ES)
VAX 3600	LANCE (ES)
VAX 3300	LANCE (ES), DELQA (XQ)
VAX 3400	LANCE (ES), DELQA (XQ)
VAX 3800	DELQA (XQ)
VAX 3900	DELQA (XQ)
VAX 4000-200	SGEC (EZ), DELQA (XQ)
VAX 4000-300	SGEC (EZ), DELQA (XQ)

(continued on next page)

System Manager Release Notes

2.20 ISL Support for VAX Systems—Extended

Table 2–2 (Cont.) ISL-Capable Systems

System	Ethernet Controller
VAXstation 2000	LANCE (ES)
VAX 3100	LANCE (ES)
VAX 3100 M30	LANCE (ES)
VAX 3100 M40	LANCE (ES)
VAX 3100 M38	LANCE (ES)
VAX 3100 M48	LANCE (ES)
VAX 3100 M76	LANCE (ES)
VAX 3000–FT	LANCE (EP)
VAX 3520	LANCE (ES), DELQA (XQ)
VAX 3540	LANCE (ES), DELQA (XQ)
Large Systems (ISL_LVAX_U3054.SYS)	
VAX 6000–200	DEMFA (FX), DEMNA (ET), DEBNI (ET), DEBNA (ET)
VAX 6000–300	DEMFA (FX), DEMNA (ET), DEBNI (ET), DEBNA (ET)
VAX 6000–400	DEMFA (FX), DEMNA (ET), DEBNI (ET), DEBNA (ET)
VAX 6000–500	DEMFA (FX), DEMNA (ET), DEBNI (ET), DEBNA (ET)
VAX 9000 (VMB_9AQ.EXE)	
VAX 9000–xxx	DEMFA (FX), DEMNA (EX)

2.21 Multiple Backups Now Supported

V5.5 In VMS Version 5.4, multiple backup operations were not supported when booting standalone Backup from the InfoServer. This problem has been corrected.

2.22 LAT Release Notes

V5.5 The following sections include information for system managers working with the new LAT software included in Version 5.5 of the VMS operating system. See Section 3.15 for related information.

Note

You can enter LATCP commands either at the LATCP> prompt or as a DCL command (interactively or in a program). If you choose the latter method, you must first define the symbol LCP and then precede each DCL command with that symbol, as shown in the following example:

```
$ LCP := $LATCP
$ LCP SET NODE/STATE=ON
```

For a description of the new features associated with this LAT software, see the *VMS Version 5.5 New Features Manual* and the revised *VMS LAT Control Program (LATCP) Manual*.

2.22.1 Setting Up LAT on the System

V5.5

The *VMS Version 5.5 New Features Manual* provides complete information about setting up your node as a LAT service and starting the LAT protocol software. Note that the recommended method for starting the LAT software is to add the following command line to to your SYS\$MANAGER:SYSTARTUP_V5.COM procedure:

```
$ @SYS$STARTUP:LAT$STARTUP.COM
```

When you boot the system, it will automatically invoke LAT\$STARTUP.COM, which in turn invokes LAT\$CONFIG.COM (which loads LTDRIVER, the LAT terminal driver) and LAT\$SYSTARTUP.COM (which defines site-specific LAT characteristics).

Note the following as well:

- SYS\$MANAGER:LAT\$SYSTARTUP.COM now replaces LTLOAD.COM. If you have not previously done so, move all your LAT site-specific commands from your original LTLOAD.COM file to SYS\$MANAGER:LAT\$SYSTARTUP.COM and change any references to LTLOAD.COM to reflect that modification. Use LAT\$SYSTARTUP.TEMPLATE as a guide. Do *not* edit LAT\$STARTUP.COM or LAT\$CONFIG.COM.
- Make sure your LAT\$SYSTARTUP.COM contains only LATCP commands. The SYSGEN command to load the LTDRIVER in your original LTLOAD.COM must not be included (LAT\$CONFIG.COM performs that task).
- Make sure that your SYSTARTUP_V5.COM file does not install LATCP.
- Note that the XTERMINAL host services software is started automatically when you boot the system.
- After the system is booted, reboot all VT1000 and VT1200 X terminals that are connected to this system (using LAT as the X Transport Protocol).

2.22.2 Installing and Setting Privileges for LATCP

V5.5

Do *not* use the VMS Install Utility to install LATCP.

With this new version of the LAT software, LATCP does not require CMKRNL privilege (the LAT software will not work if LATCP is installed with that privilege). Note as well that, while LATCP does not require any privilege to display information, it does require the OPER privilege to do SET and CREATE functions. Other privileges might be required to perform special management functions. See the LATCP help and the revised *VMS LAT Control Program (LATCP) Manual* for more information.

2.22.3 LATCP Commands Replaced

V5.5

Several LATCP commands have been replaced. Although existing command procedures that use the pre-Version 5.5 LATCP commands are still supported in this release, Digital recommends that you modify those existing command procedures accordingly and begin using the new LATCP commands when you create new command procedures.

The new commands and the old commands they replace are shown in the following table. See the LATCP help and the revised *VMS LAT Control Program (LATCP) Manual* for more information about the new commands.

System Manager Release Notes

2.22 LAT Release Notes

Old Command	New Command
START NODE	SET NODE/STATE=ON
STOP NODE	SET NODE/STATE=OFF
SET PORT/LINK[=link-name]	SET PORT (ignoring qualifier)
SET COUNTERS/ZERO	ZERO COUNTERS/NODE
SHOW CHARACTERISTICS	SHOW NODE
SHOW COUNTERS/LINK[=link-name]	SHOW LINK/COUNTERS [link-name]
SHOW COUNTERS/NODE	SHOW NODE/COUNTERS

2.22.4 Modifications to LATCP Command SET NODE

V5.5 Two qualifiers to the LATCP command SET NODE have been replaced. Although existing command procedures that use the pre-Version 5.5 qualifiers are still supported in this release, Digital recommends that you modify those existing command procedures accordingly and begin using the new LATCP commands when you create new command procedures.

The new qualifiers and the qualifiers they replace are shown in the following table:

Old Qualifier	New Qualifier
/DISABLE=group-code	/GROUPS=DISABLE=group-code
/ENABLE=group-code	/GROUPS=ENABLE=group-code

2.22.5 LATCP Qualifiers Ignored (Offering Services Over Specific Links Removed)

V5.5 With this new version of the LAT software, the ability to offer services selectively over specific links (by enabling different group codes) has been removed because all services offered by a VMS node are now offered over all available links. Group codes are now an attribute of the VMS node rather than of a specific link. Because of this change, LAT software now ignores qualifiers to certain LATCP commands, as follows:

LATCP Command	Qualifiers Ignored
CREATE LINK	/DISABLE /ENABLE
CREATE SERVICE	/LINK
SET LINK	/DISABLE /ENABLE
SET SERVICE	/LINK
START NODE	/LINK
STOP NODE	/LINK

2.22.6 Additional LATCP Commands and Qualifiers Ignored

V5.5 The following LATCP command and qualifiers are also accepted but ignored:

- SHOW SERVERS command
- SHOW COUNTERS command qualifiers:

/DEVICE
/INACTIVE
/SERVERS

2.22.7 Using the LATCP Command SET NODE/STATE=OFF or /STATE=SHUT

V5.5 The LATCP command SET NODE/STATE=OFF stops the LAT port driver (and LAT protocol software) on your node. Any existing LAT connections are aborted. Any characteristics that you changed or set with LATCP are lost.

The LATCP command SET NODE/STATE=SHUT will cause your VMS system to reject further incoming and outgoing LAT connection requests. It will not disconnect current sessions. When all sessions become disconnected, LTDRIVER will stop. This command also stops the LATAACP (indicated by an OPCOM message) preventing you from performing any LAT management functions. (See Section 2.22.8 for information about this command's affect on LAT print queues.)

To restart the LAT software on your node, execute LAT\$STARTUP.COM. LATAACP starts (indicated by an OPCOM message). The LAT characteristics defined in LAT\$SYSTARTUP.COM will then take effect.

2.22.8 LAT Print Symbiont (LATSYP)

V5.5 Because of the change to disconnect processing (described in Section 3.15.1), LATSYP no longer imposes a 5-second delay at the start of a print job.

Note

If the LAT software is stopped (by the LATCP command SET NODE /STATE=OFF or SET NODE/STATE=SHUT), LATSYP will shut down all print queues that it is processing. The system will then generate an OPCOM message indicating that the print queues are stopped. You must manually restart those print queues.

2.22.9 READSYNC Not Supported by LAT Software

V5.5 Note that the terminal characteristic TT\$M_READSYNC (which enables read synchronization) is not valid for LAT terminals. Therefore, do not use the DCL command SET TERMINAL/READSYNC to set this characteristic on LAT terminals. If you set this characteristic, your terminal might exhibit unexpected behavior when you use certain VMS utilities.

2.22.10 Setting the Large Request Packet (LRP) Value to Support X Terminals

V5.5 If your system has several X terminals or supports several incoming LAT connections from other VMS hosts, your system might use more large request packets (LRP) than previous releases of the VMS operating system because the new LAT software is allocated more transmit buffers.

System Manager Release Notes

2.22 LAT Release Notes

To avoid this problem and the slower system performance that could result, Digital recommends that you calculate the number of incoming LAT connections your system supports (based on the number of X terminals and VAX computers that will be using the DCL command SET HOST/LAT to access your system) and multiply that number by 7. Use that figure as the *minimum* setting for the number of large request packets on your system (more than the minimum is recommended). For example, if you plan to support 100 LAT connections from X terminals and other VMS hosts, use the LRPCOUNT parameter within the System Generation Utility (SYSGEN) to set the number of large request packets on your system to 700 (or more).

You do not need to make this adjustment to the LRP setting if the only LAT connections that enter your system are from terminal servers, rather than from a large number of X terminals or SET HOST /LAT users entering your system.

2.22.11 Setting LTA MAX Units Through LATCP

V5.5 You can set MAX units for LTA devices with the LATCP command SET NODE/UNIT_NUMBER_MAXIMUM. You cannot set this value through SYSGEN. See the LATCP help and the revised *VMS LAT Control Program (LATCP) Manual* for more information about the SET NODE/UNIT_NUMBER_MAXIMUM command.

2.22.12 Saving Nonpaged Pool Memory and Avoiding Duplicate Application Ports

V5.5 When using the CREATE PORT command to create an application port (for example, CREATE PORT LTA5001:/APPLICATION), you might receive an error message similar to the following:

```
%LAT-W-CMDERROR, error reported by command executor
-SYSTEM-F-DUPLNAM, duplicate name
```

This error occurs because the LAT application port that you are trying to create has already been created by some other application. That other application could be LATCP itself because LATCP's port, LATCP\$MGMT_PORT, is used to communicate with LTDRIVER.

You can avoid creating duplicate ports in two ways:

1. Use the SET NODE/DEVICE_SEED command to move the lower boundary of the device unit number range beyond the LTA devices that you are intending to use as applications ports. (By default, LTA device units that originate from the \$ASSIGN system service to LTA0: have unit numbers that fall within a range from 1 through 9999.) For example, if you know that all LTA devices from LTA7000: onward are not used as application ports, you could enter the following commands:

```
LATCP> SET NODE/DEVICE SEED=7000
LATCP> CREATE PORT LTA5001:/APPLICATION
      :
      :
LATCP> CREATE PORT LTA5010:/APPLICATION
```

2. Execute the LATCP command SET NODE/STATE=ON (either interactively or in a program) before any LTA application or dedicated ports are created. Because every LATCP Management port (LATCP\$MGMT_PORT) that was created by the previous LATCP invocation is deleted, there will be no conflict with LAT application ports or dedicated ports that are created anew. Specifying this command also minimizes the use of nonpaged pool memory.

For more information, see the descriptions of the /DEVICE_SEED and /STATE qualifiers in the SET NODE reference section of the *VMS LAT Control Program (LATCP) Manual*.

2.22.13 CREATE PORT/LOGICAL Command and BYTCNT Quota

V5.5 Note that, for each port you create with the LATCP CREATE PORT/LOGICAL command, approximately 500 bytes of BYTCNT quota will be debited from the calling process.

2.22.14 Using Dedicated Ports

V5.5 LTDRIVER no longer creates services needed for dedicated ports. To use a dedicated port with an application service, you must create the service and explicitly tell LTDRIVER that the service is an application service to be used by a dedicated port. See the CREATE PORT and CREATE SERVICE commands in LATCP help and the revised *VMS LAT Control Program (LATCP) Manual* for more information.

2.22.15 LAT Support for FDDI Controller

V5.5 This version of the LAT software supports the FDDI controller. For example, to create a link on the FDDI controller, enter the following command:

```
$ LCP CREATE LINK/DEVICE=FXA0:/STATE=ON FDDI$LINK
```

For more information on creating links, refer to the *VMS LAT Control Program (LATCP) Manual*.

2.23 Magnetic Tape ACP Notes

The release notes in this section pertain to the magnetic tape ancillary control process (ACP).

2.23.1 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 was 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.

2.23.2 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

2.24 Monitor Utility Notes

2.24 Monitor Utility Notes

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

2.24.1 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.

2.24.2 MONITOR CLUSTER Command

The following release notes pertain to the MONITOR CLUSTER command.

2.24.2.1 Error Messages

V5.4

The MONITOR CLUSTER command might 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 might 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.

System Manager Release Notes

2.24 Monitor Utility Notes

In the lower left quadrant of the display, \$DSA2000 should be DSA2000. This is also true for \$DSA90 and \$DSA125.

2.25 NMLSHR Image Superseded

V5.4-3 In VMS Version 5.4-1 the NMLSHR image was placed in SYS\$COMMON:[SYSEXE] instead of SYS\$LIBRARY. This image is updated in Version 5.4-3, inclusive of its Version 5.4-1 changes, and is located in SYS\$LIBRARY.

You can delete the NMLSHR image remaining in SYS\$COMMON:[SYSEXE] from a Version 5.4-1 or Version 5.4-1A installation after the VMS Version 5.4-3 update, as it is superseded by the SYS\$LIBRARY:NMLSHR provided on this kit.

2.26 SCSI Notes

The release notes in this section pertain to Digital Small Computer System Interface (SCSI) Devices.

2.26.1 SCSI Concepts

V5.5 This section describes the following commands and concepts:

- FORMAT UNIT command
- REASSIGN BLOCKS command
- No forced error bit
- Non-last track disks

FORMAT UNIT Command

All SCSI disks can implement a FORMAT UNIT command. This command makes sure that every block on a disk is accessible. It also causes a drive to initialize control structures that manage media defects.

REASSIGN BLOCKS Command

Some SCSI disks can implement a REASSIGN BLOCKS command. Host software can use this command to relocate the data for a specific logical block to a different physical location on a fixed disk. It is used when a read or write operation to a disk block fails due to a media defect. The REASSIGN BLOCKS command is not guaranteed to move the data from the old to the new location on the disk.

No Forced Error Bit

A unique forced error bit is associated with each block on a Digital Storage Architecture (DSA) disk. When a read operation to a DSA disk block results in a nonrecoverable error, the block is reassigned to a different physical location on the disk and the forced error bit for this block is set. Once this bit is set, error status is displayed when an attempt is made to read this disk block.

SCSI disks have no forced error bit. This means that in the particular case of a nonrecoverable read error, reassignment operations are not performed. If a read operation to a SCSI disk block results in a nonrecoverable error and the block is reassigned to a different physical location on the disk, subsequent reads to this block result in success status.

Non-Last Track

Some types of disks reserve the last track for recording bad block information. SCSI disks, however, use the last good block to record bad block information. These disks are called "non-last track disks."

2.26.2 Bad Block Support for SCSI Disks

V5.5 This section describes the VMS bad block support for SCSI disks. The following components contain VMS bad block support:

- Bad Block Locator Utility (BAD)
- Console disk formatter
- INITIALIZE command and the file system
- SCSI disk class driver

2.26.2.1 BAD Utility

V5.5 The BAD Utility ensures that every block on disk can store data reliably. Like the console disk formatter, it performs read and write operations to every block on the disk. Information about any sequence of bad blocks found on the disk is saved in the Detected Bad Block File (DBBF). This file is stored on the last good block of the disk.

2.26.2.2 Console Disk Formatter

V5.5 The console disk formatter is a program that can be run before VMS is booted and before any user data is placed on the disk. It performs the FORMAT UNIT command, followed by a sequence of read and write operations to ensure that every disk block can store data reliably. Reassignment operations are performed for blocks that return errors during read or write operations.

2.26.2.3 INITIALIZE Command and the File System

V5.5 The INITIALIZE command prepares disks to store files. It creates the file BADBLK.SYS (located at [0,0]). Blocks found to have nonrecoverable errors are allocated to this file so that they will not be allocated to user files.

Usually, all I/O operations to SCSI disks go through the SCSI disk class driver. If a nonrecoverable error is detected for a SCSI disk block, the driver returns an error status (SS\$_PARITY) to the file system. The file system then sets a flag in the file header for the file that contains the bad block. When you delete a file that contains a bad block, a process is created that attempts to read and write every block within the file. This is called the “scrub” process. It allocates any bad blocks that are found to BADBLK.SYS.

Note the following about the reinitialization of a disk:

- The contents of BADBLK.SYS are *not* preserved.
- Any blocks entered in the DBBF by the BAD Utility are *not* allocated to BADBLK.SYS.

2.26.2.4 SCSI Disk Class Driver

V5.5 Upon completion of a read or write operation, SCSI disks return one of the following status indications:

- Successful
- Recoverable (soft error)
- Nonrecoverable (hard error)

Recoverable errors fall into two categories:

- Recoverable with read retries—These errors are considered to be successful by the disk class driver.

System Manager Release Notes

2.26 SCSI Notes

- Recoverable with ECC correction—These errors can lead to the class driver attempting the operation again and possibly reassigning the blocks.

Nonrecoverable errors lead to retries and possibly to the reassignment of blocks.

2.26.3 Error Recovery Parameters

V5.5

SCSI disks have a set of flags, called error recovery parameters. These flags control what happens when an error is detected. The settings of each of these flags are as follows:

- **AWRE, ARRE**—Control whether the disk automatically reassigns blocks when errors are detected during write and read operations. All SCSI disks supplied by Digital have these flags set to zero, preventing automatic block reassignment. Thus, all reassign operations are performed under host software control.
- **TB**—Controls whether the disk transfers the block for which an error occurs. This feature is enabled by the SCSI disk class driver (that is, the bad block is transferred).
- **PER**—Controls whether the disk informs the host when a recoverable error occurs. This flag is set by the SCSI disk class driver.
- **DTE**—Determines whether the disk stops a transfer when the first error is detected. If this flag is not set, the disk class driver is informed of only the last error in a transfer with errors on multiple blocks. The disk class driver sets this flag for all disks that support this feature.
- **DCR**—When set, prevents the disk from applying corrections during error recovery. The disk class driver clears this flag for all SCSI disks.

2.26.4 What the Disk Class Driver Does When It Detects an Error

V5.5

When the disk class driver detects a read or write error, it tries the operation again three times. If appropriate, it reassigns blocks to prevent media defects from causing future errors. The following list is a summary of the algorithms implemented by the SCSI disk class driver for attempting the operation again (retry):

1. The driver detects an error during a read or write operation.
2. The driver retries the operation three times. If an attempt (retry) is successful, it exits.
3. If the three attempts fail and this is a nonrecoverable read operation, the driver returns a status of `SS$_PARITY`. (A reassignment would result in undetected user data corruption because there is no forced error bit in SCSI.)
In all other cases—a recoverable read, a recoverable write, or a nonrecoverable write operation—the driver attempts to reassign the block up to three times.
4. If the reassignment succeeds, the driver writes the original data to the reassigned block. The `REASSIGN BLOCKS` command is not guaranteed to move the data from the old location to the new location on the disk.
If the write operation after the reassignment fails and fewer than three reassignments, try the reassignment operation again.

2.26.5 Use of Disk Caching Products with SCSI Disks

V5.5 In VMS releases from Version 5.4 to Version 5.4-3, running some third-party disk caching products on systems with SCSI disks can lead to disk corruption. This is a side effect of a disk-seek optimization feature that was added in VMS Version 5.4. A replacement SCSI disk class driver image (DKDRIVER) is available from the Customer Support Center. This replacement driver disables disk seek optimization and allows third party caching products to work without causing disk corruption.

The version of the SCSI disk class driver shipped with VMS Version 5.5 fixes this problem.

2.26.6 Timing in MicroVAX 3100 SCSI Port Driver Fixed

V5.4 In VMS Version V5.4, the SCSI port driver for the MicroVAX and VAXstation 3100 (PKNDRIVER) was changed to conform to the timing values specified by the SCSI specification. These changes caused the SCSI port driver to be incompatible with some third party disk and tape devices. The SCSI port driver supplied in VMS Version 5.5 reverts to the timing of the pre-Version 5.4 driver.

2.27 Security Features—Notes

The release notes in this section pertain to system security features.

2.27.1 COMMSYNC Qualifier to the SET TERMINAL Command

V5.5 The /COMMSYNC qualifier to the DCL command SET TERMINAL (and the TT\$M_COMMSYNC characteristic for the TTDRIVER interface) should *never* be set on a line with a modem hookup intended for interactive use. The qualifier disables the modem terminal characteristic that disconnects a user process from the terminal line in case of a modem phone line failure. With the /COMMSYNCH qualifier enabled, the next call on the terminal line could be attached to the previous user's process. The /COMMSYNC qualifier is intended to allow connection of asynchronous printers and other devices to terminal ports by using modem signals as flow control. Security administrators should be aware that the characteristic should *not* be used on interactive terminal ports.

2.27.2 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 the resulting ERAPATLOA.EXE system loadable image in the directory SYS\$LOADABLE_IMAGES.

VMS Version 5.2 contains several corrections that address all problems relating to using a loadable, nonzero erase pattern.

2.27.3 Modifying the System Password Dictionary

V5.5 In Version 5.4, the VMS operating system screened potential passwords for their acceptability. The DCL command SET PASSWORD takes a user's proposed password, converts it to lowercase (if necessary) and compares it to entries in a system dictionary. If a proposed password is found in the dictionary, it is rejected as a valid user password and the user has to suggest another.

System Manager Release Notes

2.27 Security Features—Notes

Many system managers would like to modify the system password dictionary to include words of significance to their site. The following procedure allows system managers to add words to or to delete words from the system dictionary in VMS Version 5.5. The procedure allows a site to retain a file of the words they consider unacceptable.

1. Create a file containing passwords you would like to add to the dictionary. Each password should be on a separate line and in lowercase, as follows:

```
$ CREATE LOCAL_PASSWORD_DICTIONARY.DATA
somefamous
localheros
Ctrl/Z
```

2. Enable SYSPRV and merge your local additions:

```
$ SET PROCESS/PRIVILEGE=SYSPRV
$ CONVERT/MERGE/PAD LOCAL_PASSWORD_DICTIONARY.DATA -
_$ SYSS$LIBRARY:VMS$PASSWORD_DICTIONARY.DATA
```

2.27.4 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 DECNET account, 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 5.5 New Features Manual*. For more information about network security, refer to the *Guide to VMS System Security*.

2.27.5 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.

2.27.6 Reestablishing the 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.

2.27.7 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 SYSTARTUP_V5.COM startup command procedure.

2.27.8 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 about security auditing, see the *Guide to VMS System Security*.

2.27.9 Spawned Subprocesses that Execute SET UIC

V5.4 The DCL command SPAWN uses a temporary mailbox to pass information to a newly created subprocess. In VMS Version 5.4, the protection on this mailbox was changed from S:RWLP,O:RWLP,G:RWLP,W:RWLP (the default protection for mailboxes) to S,O:RWLP,G,W.

System Manager Release Notes

2.27 Security Features—Notes

The change of protection on temporary mailboxes does not affect unprivileged users, but it does affect system managers who run command procedures that spawn subprocesses that execute the DCL command SET UIC. These system managers must ensure that the command procedure changes its UIC back to its original value before logging out. Unless the command procedure modifies its UIC, it is unable to log out and the system returns the error message "Insufficient privilege or file protection violation."

2.27.10 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 might 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 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 entering the command:

```
$ REPLY/LOG/DISABLE=SECURITY
```

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).

2.27.11 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 want 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("$254$dual18", "mnt") -  
  then mount/system $254$dual18 audit audit$ /norebuild  
$ define/system/exec audit_server audit$:[audit]audit_server.dat
```

Because you invoke the SYSECURITY.COM command procedure before you start the audit server, do not place any SET AUDIT commands in this file.

2.27.12 User Authorization File (UAF) Notes

The following sections pertain to the User Authorization File (UAF).

2.27.12.1 Captive Accounts—Batch and Network Restrictions Removed

V5.4

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).

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 can 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.

2.27.12.2 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 2.27.12.1), 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.

System Manager Release Notes

2.27 Security Features—Notes

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.

2.27.12.3 Changes to the CAPTIVE Flag

V5.2

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 might 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, 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 want to take advantage of the new behavior of CAPTIVE must recompile and relink from source.

Many Digital programming 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 programming 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.

System Manager Release Notes

2.27 Security Features—Notes

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.

2.27.12.4 DISIMAGE Flag

V5.2

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 UAF flags RESTRICTED or DEFCLI, 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.

2.27.12.5 RESTRICTED Accounts—Incompatibility Problems Corrected

V5.4 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.

2.27.12.6 UAF Record Length Enforcement

V5.2 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 make sure 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.

2.27.12.7 UAF Template File Changes

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.

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.

With VMS Version 5.2, the following changes were made to the UAF template file (SYS\$SYSTEM:SYSUAF.TEMPLATE):

- The UAF parameter PWDLIFETIME for the accounts FIELD, SYSTEM, SYSTEST, and SYSTEST_CLIG has been lowered from 90 days to 30 days

System Manager Release Notes

2.27 Security Features—Notes

- 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 might want 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 might not function correctly. Depending upon the layered product installation procedures, this might 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.

2.28 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.

2.29 SPKITBLD.COM Procedure—Change

V5.4-3

When using a data file to run the SPKITBLD.COM procedure to build an update kit on a large media device (specified by the user), the procedure would not build the kit on another media device even though the following prompt (which you use to specify another device) was included in the data file:

```
SPKITBLD$NEW_MEDIA_FOR_SAVESET_N = "YES"
```

The SPKITBLD.COM procedure no longer ignores the previously described prompt and now allows you to build a kit on multiple media devices (as long as the prompt in the data file is answered with "YES").

2.30 Standalone BACKUP Correction

V5.5

In VMS Version 5.4–3, standalone BACKUP could fail with several different CPU and peripheral configurations, as follows:

- The I/O peripherals (disk, tape, and console devices) might not configure correctly.
- The system might halt during booting.
- You might receive the following error message when using standalone BACKUP to restore the B saveset to the system disk:

```
%SYSTEM-F-INSVIRPAG, insufficient virtual pages
```

To avoid these problems, you had to perform a conversational boot of the standalone BACKUP kit by manually setting the number 5 console boot register to 1, using the following SYSGEN parameters:

```
SYSBOOT> SET NPAGEVIR 2000000  
SYSBOOT> CONTINUE
```

This problem has been corrected in VMS Version 5.5.

2.31 Standalone BACKUP Restriction

V5.4

Note the following restriction when using standalone BACKUP:

Using SYS\$UPDATE:STABACKIT.COM to build a standalone BACKUP on a VAX–11/780 or VAX–11/785 console device will fail because the STANDALON.EXE image now exceeds the capacity of RX01 console media. To avoid this problem, use RX01 diskettes to store a standalone BACKUP kit built using a previous version of the VMS operating system.

2.32 Startup and Shutdown Procedures Modified

The system startup procedure template (SYS\$MANAGER:SYSTARTUP_V5.TEMPLATE) and the system shutdown procedure (SYS\$SYSTEM:SHUTDOWN.COM) have been modified to accommodate changes to the new batch and print queuing system.

Sections 2.32.1 and 2.32.2 describe the changes.

2.32.1 Startup Procedure Template Changes

V5.5

The DCL command START/QUEUE/MANAGER is no longer included in the startup procedure template. After you enter the START/QUEUE/MANAGER command for the first time (for example, during installation), the command is stored in the queue database and will be automatically issued during system startup. Once the queue manager is started, you need enter the START/QUEUE/MANAGER command only after you enter the DCL command STOP/QUEUE/MANAGER/CLUSTER. You can delete the START/QUEUE/MANAGER command from your STARTUP_V5.COM procedure.

To take advantage of the queue autostart feature, the template now includes the DCL command ENABLE AUTOSTART/QUEUES. This command automatically starts all active autostart queues on the node. It also allows any queues that were executing on another node in a cluster and waiting to fail over to this node to do so. For more information, see the description of the autostart feature in the *VMS Version 5.5 New Features Manual*.

System Manager Release Notes

2.32 Startup and Shutdown Procedures Modified

2.32.2 SHUTDOWN.COM—Changes for Autostart Queues

V5.5

The shutdown procedure SYS\$SYSTEM:SHUTDOWN.COM has been modified to work with the batch and print queuing system autostart feature. During shutdown, autostart is disabled to allow autostart queues with failover lists to fail over to another node. It also prevents any autostart queue running on another node in the cluster to fail over to the node being shut down.

By default, the shutdown procedure disables autostart at the beginning of the shutdown sequence. This operation has no effect if no autostart queues exist on the node.

You can change the time at which autostart is disabled in the shutdown sequence in one of two ways:

- Define the logical name SHUTDOWN\$DISABLE_AUTOSTART.
- Specify the DISABLE_AUTOSTART option during the shutdown procedure. (The value you specify as a shutdown option value overrides the value specified for the SHUTDOWN\$DISABLE_AUTOSTART logical name.)

Define the logical name SHUTDOWN\$DISABLE_AUTOSTART as follows, where *n* is the number of minutes before shutdown when autostart is to be disabled:

```
$ DEFINE/SYSTEM/EXECUTIVE SHUTDOWN$DISABLE_AUTOSTART n
```

You can add this logical name definition to SYLOGICALS.COM.

The value you specify for *n* is the default value for the node. If this number is greater than the number of minutes specified for the entire shutdown sequence, autostart will be disabled at the beginning of the sequence. The following example disables autostart 20 minutes before the node shuts down:

```
$ DEFINE/SYSTEM/EXECUTIVE SHUTDOWN$DISABLE_AUTOSTART 20
```

If you have not defined the logical name or if you want to override the default value, specify the DISABLE_AUTOSTART=*n* option (where *n* is the number of minutes) when you are prompted to enter shutdown options in the shutdown command procedure as follows:

```
Shutdown options (enter as a comma-separated list):
REBOOT CHECK          Check existence of basic system files
SAVE_FEEDBACK        Save AUTOGEN feedback information from this boot
DISABLE_AUTOSTART    Disable autostart queues
```

```
Shutdown options [NONE]: DISABLE_AUTOSTART=5,REBOOT_CHECK
```

2.32.3 SHUTDOWN.COM—Change in Disk Dismount Reporting

V5.4

In previous VMS versions, 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.

2.33 SYSGEN Utility Notes

The release notes in this section pertain to the SYSGEN utility.

2.33.1 Configuration Note

V5.4-2 The Autoconfigure facility of the System Generation Utility (SYSGEN) incorrectly configures a CIBCA device in the following circumstances:

- When a DEBNA, DEBNI, TBK50, or TBK70 device is in the VAXBI slot directly preceding the CIBCA.
- When a DEBNA, DEBNI, TBK50, or TBK70 device is in the last slot on one VAXBI, and the CIBCA is in the first slot on the next-highest VAXBI.

Note

This error will not occur if you boot the system from the CIBCA, because the boot device is not configured by SYSGEN when you boot from the CIBCA.

The configuration error results in timeout errors on the PAA0 device when you boot standalone BACKUP or the VMS operating system. To avoid timeout errors, reconfigure the bus. This problem will be fixed in a future version of the VMS operating system.

2.33.2 CTLPAGES Parameter

V5.4 Beginning with VMS Version 5.4, the SYSGEN Utility parameter CTLPAGES specifies the number of pages in the P1 space variable-length pool. CTLPAGES determines the size of the process allocation region, which is a P1 space dynamic memory pool. The parameter is used primarily for data structures that need to survive image rundown.

2.33.3 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 if the current process is executing at priority 1, then 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.

2.33.4 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).

Modifying the SYSGEN parameter MULTIPROCESSING on a uniprocessor workstation can result in unexpected system failures. If full SMP checking is desired, you must reboot the uniprocessor workstation and must not run DECwindows while full SMP checking is in effect.

System Manager Release Notes

2.34 SYSLOA—Page Fault Problem Corrected

2.34 SYSLOA—Page Fault Problem Corrected

V5.3 In previous VMS versions, 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.

2.35 System Disk Size Recommendation: 115 MB

V5.5 With VMS Version 5.5, to support full VMS and full DECwindows, a system disk of greater than 115 MB is recommended. (This recommendation does not include the dump file space.) When you use a smaller disk, additional tailoring is required prior to installing options such as DECwindows.

2.36 TA90E Tape Drive

The release notes in this section pertain to the TA90E tape drive.

2.36.1 TA90E 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 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 TEST.DAT MUA0:JUNE91.BCK /MEDIA_FORMAT=NOCOMPACTION -  
_ $ /REWIND/LABEL=TAPE1
```

For more information about using BACKUP with the TA90E, see Section 2.36.2.

2.36.2 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.

2.37 TLZ04 Tape Driver—Problem Corrected

V5.4-1

VMS Version 5.4-1 corrected a problem with the TLZ04 tape driver that is related to the PORT driver associated with the Q-bus-to-SCSI Adapter (QZA) and that is found on systems using that adapter.

Prior to VMS Version 5.4-1, mixed bus configurations such as the Small Computer System Interface (SCSI) and Digital Storage Systems Interconnect (DSSI) sometimes saw multiple bus resets when a system crash occurred and attempted to reboot immediately after the crash. (A bus reset causes all of the devices on the bus to go off-line and into mount verification.) When the system disk was not a SCSI device and the bus was not reset, the PKI driver software assumed asynchronous operation until it was notified otherwise. When notification did not take place, the driver detected phase or bus errors and issued a bus reset. The driver then attempted to retry the failed operation and failed again. This scenario led to bus resets in loop mode. VMS Version 5.4-1 corrects the multiple bus reset problem.

2.38 UETP Load Phase Failure in Vector Processing Systems

V5.4-1

The load phase of the UETP does not run correctly on a system with installed vector processors.

For systems without vector processors, UETP assumes that each load to be tested will consume one process control block (PCB) slot. However, in a vector processing system, UETLOAD02.COM must spawn two additional processes for each vector processor that is present and run as many loads as there are available PCB slots. When the vector processor load test attempts to spawn these additional processes, the attempts fail, displaying the following error text:

```
*****
* UETVECTOR          *
* Error count = 1    *
*****
%SYSTEM-F-NOSLOT, no PCB available
-UETP-E-SUBSPNERR, Error spawning subordinate process.
-UETP-E-SUBSCHERR, Error scheduling subordinate process.
%UETP-I-ENDED, UETLOAD02_0060 ended at 18-JUN-1991 16:01:58.47
```

System Manager Release Notes

2.38 UETP Load Phase Failure in Vector Processing Systems

To successfully run the load phase in a vector processing system, you must reduce the computed number of loads. The approximate reduction is a function of the number of loads to be run and the number of vector processors in the system. Reduce the default number of loads computed by UETP according to the following formula:

$$(2 * \text{number of vector processors}) * \frac{\text{Number of loads}}{10}$$

Instructions for running UETP appear in the *VMS Version 5.5 Upgrade and Installation Manual*.

2.39 UNIBUS and Q22 Bus Interrupt Vector Change

V5.4-3

Interrupt vector changes to some hardware devices might be required when upgrading an existing UNIBUS or Q22 bus 4.n system to VMS Version 5.0 or higher.

The VMS algorithm used to allocate interrupt vectors for UNIBUS and Q22 bus peripherals changed at VMS Version 5.0. When systems are autoconfigured during booting, some UNIBUS or Q22 bus systems might require peripherals to have their interrupt vectors modified by a Digital Service representative. (Note that this change does not affect most VMS systems. Although this problem has existed since Version 5.0, this is the first time that it appears fully corrected in a release note.)

The kinds of 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
- LNV21
- VS100
- VSV21
- IBQ01

Refer to the SYSGEN device table discussion in the *VMS Device Support Manual* for more information concerning device vector assignments.

Behavior of Old and New Algorithms

SYSGEN has been modified to support the old and new vector allocation algorithms. The new algorithm is used as the default behavior. The old algorithm might not be supported indefinitely. Selection of the autoconfiguration algorithm is controlled by the SYSGEN parameter VMS8. This parameter can be set to give either the old or new algorithm behavior according to the following chart:

VMS8 Value	Description
VMS8 = 0	Gives new behavior algorithm with error messages
VMS8 = 1	Gives old behavior algorithm with error messages

System Manager Release Notes

2.39 UNIBUS and Q22 Bus Interrupt Vector Change

VMS8 Value	Description
VMS8 = 2	Gives new behavior algorithm without error messages

In Case of Problems

If SYSGEN detects a difference between the old and the new algorithms while using the new behavior algorithm, it displays the following error message (UNIBUS example):

```
SYSGEN-W-MISCONFUNI, UNIBUS device DMF32 has been misconfigured,  
interrupt vector should be 000324
```

This error message for UNIBUS or Q22 bus systems is written to the terminal that is running SYSGEN, OPCOM, and the error logger. If this error message is received, call your Digital Services representative or proceed as follows:

1. Reboot the system, using the SYSBOOT breakpoint, R5 = 1.
2. Set the SYSGEN parameter VMS8 = 1 (see the preceding chart of VMS8 parameter values). This allows the system to boot successfully.
3. Continue to boot the system until the following message is displayed:

```
%SYSGEN-W-NONSTDUNI, UNIBUS in nonstandard configuration,  
device DMF32 vector is wrong
```
4. Execute the following command to determine the present interrupt vector settings after the system has booted:

```
$ RUN SYS$SYSTEM:SYSGEN SHOW/CONFIGURE/COMMAND_FILE
```
5. Next, use the CONFIGURE command to determine the correct vectors for all devices. Enter the following commands:

```
$ RUN SYS$SYSTEM:SYSGEN  
SYSGEN> CONFIGURE  
DEVICE> device-name  
DEVICE> [Ctrl/Z]
```
6. Compare the present (incorrect) interrupt vector settings with the just determined new (correct) interrupt vector settings.
7. The boards with the incorrect interrupt vectors must now be rejumped with the new (correct) interrupt vector settings.
8. Once all of the incorrect vectors have been rejumped, the system should be rebooted with VMS8 set to 2. Setting VMS8 = 2 disables error messages for bad vector problems and uses the new vector allocation algorithm.

Caution

Setting VMS8 = 2 without changing the proper interrupt vectors can result in system failures and crashes.

Note

As long as VMS8 is set to 1, and a device found on the bus is misconfigured, the following message is displayed:

```
%SYSGEN-W-NONSTDUNI, UNIBUS in nonstandard configuration,  
device DMF32 vector is wrong
```

System Manager Release Notes

2.39 UNIBUS and Q22 Bus Interrupt Vector Change

This error message means that a device has been found on the bus whose interrupt vector is incorrect and should be changed. This message will continue to be displayed until the situation is corrected.

2.40 Upgrade Notes

The release notes in this section contain information that pertains to upgrading the VMS operating system. See Section 2.49.6 for installation considerations for VAXclusters.

2.40.1 Applying the PADRIVER01u3054 to VMS Version 5.4–3 System Disks

V5.4-3

The PADRIVER01u3054 kit must be applied to all VMS Version 5.4–3 system disks. The VMS Version 5.5 Mandatory Update (MUP) tape contains the required saveset. The following steps describe the procedure for applying the PADRIVER01u3054 kit; steps 1 through 5 must be performed *before* Version 5.5 nodes with Version 5.4 systems join the VAXcluster:

1. Log in to the Version 5.4-3 system on the SYSTEM account.
2. Mount the VMSMUP055 tape on the tape drive (MUA0:).
3. Use BACKUP to extract the PADRIVER01u3054.a saveset:

```
$ BACKUP /SELECT=PADRIVER01U3054.A -  
MUA0:VMSMUP055.B/SAV SYSS$COMMON:[SYSUPD]*
```
4. Apply the PADRIVER01u3054 kit:

```
$ @SYSS$UPDATE:VMSINSTAL PADRIVER01U3054
```
5. Reboot all Version 5.4–3 systems.
6. Proceed with the Version 5.5 upgrade.

2.40.2 CIXCD Firmware/Microcode Requirements for Booting VMS Version 5.5

V5.5

The CIXCD is an XMI to CI bus adapter for all current VAX 9000 and 6000 systems. The minimum required version of Firmware/Microcode to use CIXCD on VMS Version 5.5 is 2.03. The CIXCD will not be brought online if the microcode revision requirement is not met. In this case, a warning will be printed and the port will be brought offline.

2.40.3 Layered Product Cautions

V5.5

After you upgrade to VMS Version 5.5, 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 5.5. Some available layered products that exhibit compatibility problems with this version are described in this section.

For a complete listing of layered products available for VMS Version 5.5, see the appropriate appendix in the *VMS Version 5.5 Upgrade and Installation Manual*. If you require additional information on any upgrade procedure or if you encounter problems, contact your Digital Services Group representative.

2.40.3.1 ALL-IN-1 Version 2.4 Shareable Images Requirement for CDA Support

V5.5

VMS Version 5.5 provides three 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 5.5 are not installed as known images. If you require CDA support for ALL-IN-1 Version 2.4, you must install the three new shareable images as known images. If you do not require CDA support, no action is required.

To install the three 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:

```
$ INSTALL CREATE SYS$SHARE:XDPS$DPSLIBSHR.EXE
$ INSTALL CREATE SYS$SHARE:XDPS$DPSCLIENTSHR.EXE
$ INSTALL CREATE SYS$SHARE:XDPS$DPSBINDINGSSHR.EXE
```

2.40.3.2 DECwindows Developer's Kit Requirement

V5.5

Applications that use the DECwindows Developer Kit on VMS for OSF/Motif, Version 1.1 cannot be linked on VMS Version 5.5 systems. A bug fix in the Version 5.5 linker makes it incompatible with the Motif Developer Kit. This problem is indicated by the following linker error message:

```
%LINK-E-MULSHRPSC, psect XTCXTTOOLKITERROR defined in shareable image
DSA0:[SYS3.SYSCOMMON.][SYSLIB]DECW$MOTIF$XMSHR.EXE;1
      is multiply defined in shareable image
DSA0:[SYS3.SYSCOMMON.][SYSLIB]DECW$MOTIF$XTSHR.EXE;1
-LINK-E-NOIMGFIL, image file not created
```

If your application uses the Motif Developers ToolKit, you have the following options:

- Use an earlier version of the VMS Linker to link your application.
- Use VMS DECwindows Motif Version 1.0 instead of the Motif Developer Kit. Applications that use VMS DECwindows Motif Version 1.0 can be linked using the V5.5 linker. Note that VMS DECwindows Motif Version 1.0 supersedes the Motif Developer Kit.

If you choose to link your application by using an earlier version of the VMS linker, make sure you save a copy of your current linker executable image *before* upgrading your system to VMS Version 5.5, as the following procedure describes.

1. Copy the LINK.EXE file from the SYS\$SYSTEM: directory to one of your own directories.
2. Perform the upgrade.
3. Copy the earlier version of LINK.EXE back into the SYS\$SYSTEM: directory, renaming it with a descriptive name, such as MOTIF-LINK.EXE, to distinguish it from the Version 5.5 linker. (You can copy the earlier version of LINK.EXE into another directory; however, if you do, adjust the logical name definition in the following step accordingly.)
4. Redefine the translation of the LINK command so that it invokes the older version of the Linker, as in the following example.

```
$ DEFINE LINK SYS$SYSTEM:MOTIF-LINK.EXE
```

System Manager Release Notes

2.40 Upgrade Notes

This command can be entered interactively or inserted in an application build procedure before the line in which the LINK command is invoked.

2.40.3.3 VAX APL Interpreter Requirement

V5.5

VAX APL Version 4.0 might not install on systems running VMS Version 5.4–3, Version 5.5, or on systems that have VMS DECwindows Motif installed on them. To successfully install VAX APL Version 4.0 on one of these systems you must do the following:

1. Check to see if there are font files in the following two directories:

```
$ DIR SYS$COMMON:[SYSFONT.DECW.USER_75DPI]*.DECW$FONT
$ DIR SYS$COMMON:[SYSFONT.DECW.USER_100DPI]*.DECW$FONT
```

If there are font files in both directories, no action is required, and you can skip the next step.

2. If there are no font files, copy a font file from a system font directory to each of the following directories:

```
$ COPY SYS$COMMON:[SYSFONT.DECW.75DPI]FIXED.DECW$FONT -
_$ SYS$COMMON:[SYSFONT.DECW.USER_75DPI]
_$ COPY SYS$COMMON:[SYSFONT.DECW.100DPI]FIXED_100DPI.DECW$FONT -
_$ SYS$COMMON:[SYSFONT.DECW.USER_100DPI]
```

3. Install APL Version 4.0 as you would normally.
4. Execute the following command procedure after installing APL Version 4.0:

```
$ @SYS$UPDATE:DECW$MKFONDIR
```

2.40.4 Rolling Upgrades from VMS Version 5.4 to Version 5.5

V5.5

To perform a VMS Version 5.5 rolling upgrade, all system disks in the cluster must be running at least VMS Version 5.4. For detailed information about the upgrade procedure, see the *VMS Version 5.5 Upgrade and Installation Manual*.

2.40.5 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 or a subsequent version on a VAXstation 8000.

2.41 VAXft SYSTEMS—EFDRIVER OPCOM Messages

V5.5

This section describes EFDRIVER OPCOM messages. EFDRIVER messages are prefixed by a logical adapter device name and also, in some cases, by a physical device name. OPCOM messages for EFDRIVER have the following format:

```
%%%%%%%%%% OPCOM dd-mmm-yyyy hh:mm:ss.ss %%%%%%%%%%%
Message from user EFDRIVER on node-name
ddcu: (ddcu:) - message-text
```

Failover to Standby Ethernet Adapter

Explanation: An Ethernet failover has occurred on a VAXft system.

User Action: If the Failover Set Manager (FSM) Utility was used to force a failover or if a zone was intentionally stopped, no further action is required.

However, if this message occurred for another reason, such as loss of a zone due to an unrelated fault, take appropriate action depending upon the probable cause.

- Verify that the FSM Utility has not been used to force a failover.

System Manager Release Notes

2.41 VAXft SYSTEMS—EFDRIVER OPCOM Messages

- Verify that a zone has not been intentionally stopped.
- Check the connections of both physical Ethernet adapters to the Ethernet.
- Check to see that both zones are still active.

If the FSM Utility has not been used to force a failover, both zones are active, and all Ethernet connections are intact, then contact Digital Services.

Loopback Between Adapters Fails

Explanation: A loopback test between the standby Ethernet adapter and the current, active Ethernet adapter on a VAXft system has timed out. This test previously succeeded. This message is normally followed by a failover message.

User Action: Take appropriate action depending upon the probable cause.

- Verify that a zone has not been intentionally stopped.
- Check the connections of both physical Ethernet adapters to the Ethernet.
- Check to see that both zones are still active.

If both Ethernet connections are intact, both zones are active, and both physical adapters are still connected to the same Ethernet, then contact Digital Services.

Loopback Between Adapters Fails From Both Directions

Explanation: A loopback test between the standby Ethernet adapter and the current, active Ethernet adapter on a VAXft system has timed out. This test failed after trying both adapters as the current adapter. The status of the Ethernet adapters is not known by the software, so the logical adapter is not running in a fault tolerant mode. Possibly the two adapters are no longer connected to the same Ethernet or there is some network hardware problem that cannot be isolated by the VMS software.

User Action: If both physical Ethernet adapters in the failover set have terminators installed, you can ignore this message. Otherwise, check for loose Ethernet cables and check that both physical adapters in the failover set are attached to the same Ethernet. Do this by observing where each transceiver cable leading from its I/O module connects to its respective transceiver on the Ethernet. Ethernet failover cannot occur until this problem is corrected. If you cannot correct the problem, contact Digital Services.

Loopback Between Adapters Fails On First Attempt

Explanation: A loopback test between the standby Ethernet adapter and the current, active Ethernet adapter on a VAXft system has timed out. This test has not succeeded since the system was bootstrapped or since the failover set was altered.

User Action: You can ignore this message if both physical Ethernet adapters in the failover set have terminators installed. Otherwise, both physical Ethernet adapters in the failover set must be attached to the same Ethernet. Verify this by observing where each transceiver cable leading from its I/O module connects to its respective transceiver on the Ethernet. The fault tolerance of the Ethernet is compromised until this problem is corrected. If you cannot correct the problem, contact Digital Services.

System Manager Release Notes

2.41 VAXft SYSTEMS—EFDRIVER OPCOM Messages

Loopback Between Adapters Now Succeeds

Explanation: A condition that previously caused the failure of a loopback test between the standby Ethernet adapter and the current, active Ethernet adapter on a VAXft system has been corrected.

User Action: None.

Physical Ethernet Adapter Is Now Operational

Explanation: A VAXft physical Ethernet adapter that previously failed a verification test now succeeds.

User Action: None.

Physical Ethernet Adapter Verification Failure

Explanation: A verification test performed on a VAXft physical Ethernet adapter has failed.

User Action: Verify that the failed physical Ethernet adapter's transceiver cable is properly connected to both the I/O module and the transceiver, or that the I/O module is properly terminated. Verify that both zones are active. If the Ethernet connections are intact and both zones are active, then contact Digital Services.

2.42 VAX 3100 Model 76 Notes

This section contains notes about the VAX 3100 Model 76 system.

2.42.1 System Crash

V5.4-2

Some VAX 3100 Model 76 systems might exhibit a rare system crash from a soft interrupt (due to a cache parity error). The crash generates a display on the stack that includes SYSLOA+00D2C.

2.42.2 AUTOGEN LOCKDIRWT Parameter

V5.4-2

Previously, the default AUTOGEN parameters set LOCKDIRWT to 1, which created potential performance problems on the VAX 3100 Model 76. This problem has been corrected; LOCKDIRWT is now set to 0.

2.43 VAX 6000 Series Computer Notes

The release notes in this section pertain to VAX 6000-series computers.

2.43.1 VAX 6000 Series Console Tapes and Tape Serving Devices

V5.5

All clusterwide accessible devices (devices that are accessed by more than one node in a VAXcluster) must be uniquely named in the cluster. The device name must provide a reliable way to access it in the cluster. The TK console tape drive (the TK tape drive within the VAX 6000 cabinet) that is a part of many VAX 6000 systems is typically named MUA6 or MUB6. This can introduce a naming conflict when using the VMS tape server in a cluster comprised of two or more VAX 6000s when tape serving is desired.

Examples of tape devices that are accessible by more than one node in a cluster include dual-ported tapes, for example a TA90 tape drive connected to two HSC controllers, and tapes that are served to all nodes in a cluster by the VMS tape server. A tape device that is dual pathed between two computers must be identified by a unique, path-independent name that includes a **tape allocation class**. The tape allocation class is a numeric value from 0 to 255 that is used to create a device name in the following format:

System Manager Release Notes 2.43 VAX 6000 Series Computer Notes

\$tape-allocation-class\$device-name

For example, \$1\$MUA6, \$1\$MUB6, \$2\$MUA6 are all unique device names. The first two have the same tape allocation class but different controller letters, A and B respectively. The third device has a tape allocation class that is different from the first two.

The VAX 6000 series console tape, if made a cluster-accessible device via the VMS tape server, must have a name unique from all other cluster-accessible tape drives. Providing the console TK tape drive with a unique name can be accomplished by several methods.

- All VAX 6000 systems can be given a unique SYSGEN parameter value for TAPE_ALLOCLASS. In a VAXcluster with multiple VAX 6000 series systems, if each VAX 6000 system has a different TAPE_ALLOCLASS value, it is not necessary to change the unit number of any of the TK tape drives.

This might not be the desired option when serving dual ported tape drives because it limits the accessibility of HSC tapes to NI nodes in a VAXcluster to just one node and does not provide a method of failover in the event that the serving node should become unavailable.

- All VAX 6000 system console tape unit numbers can be set to a unique value. In a VAXcluster with multiple VAX 6000 systems serving tapes using the TMSCP tape server, it might be necessary to change the unit number of the TK70 or TK50 tape drive in the system cabinet to adhere to the cluster device naming conventions. If the TAPE_ALLOCLASS of two or more VAX 6000 systems in the same VAXcluster are the same and there are multiple TK drives with the same controller letter and unit number, it will be necessary to change the unit numbers of these tape drives such that they are unique and conform to the cluster device naming conventions.

The unit number of the TK console drives is controlled by the BI Bus unit number plug of the TKB70 controller in the VAX 6000 BI backplane; it must not conflict with any other tape drive in the cluster. The unit numbers available are in the range of 0 to 15, inclusive, and must not be the same as any other controller card in the BI backplane. The operation of changing the unit number should be performed by a Digital Services technician. The unit number should be set to a value other than 6 (the default value) and must not conflict with any other tape drive in the cluster.

- In a cluster with two VAX 6000 systems, each having a console tape drive with the same unit number, the console tapes can be set up to have different controller letters. For VAX 6000 systems that contain two BI backplanes, the problem can be resolved by moving the TKB70 controller card in one of the VAX systems from one BI backplane to the other. This operation will change the controller letter of the tape drive without changing the unit number; for example, MUA6 will become MUB6. This operation alone will satisfy the cluster device naming conventions when there are only two VAX 6000s in the VAXcluster. This operation should be performed by Digital Services.

Failure to provide a clusterwide unique device name for TK tape drives in the VAX 6000 cabinet in a VAXcluster that has multiple VAX 6000 systems with the same TAPE_ALLOCLASS will result in a violation of the cluster device naming convention and could have unpredictable results.

Please refer to the *VMS VAXcluster Manual* for a complete description of cluster device naming conventions.

System Manager Release Notes

2.43 VAX 6000 Series Computer Notes

2.43.2 VAX 6000 Model 500 Series Configuration Note

V5.4-2 On a VAX 6000 Model 500 series system, Digital recommends that primary CPU occupy the lowest numbered XMI slot. VAX 6000 Model 500 series systems are usually configured this way. If you ignore this restriction, the SHUTDOWN procedure might not complete normally and you will have to halt your system using the console.

2.43.3 VAX 6000 Model 600 Series—Restriction

V5.4-2 The *VMS System Generation Utility Manual* notes that you should not use the SYSGEN command SHOW/UNIBUS on a running VMS system; that command is valid only during a conversational boot. If you plan to use the SYSGEN command SHOW/UNIBUS on a VAX 6000 Model 600 series system during a conversational boot, there is an additional restriction: Digital recommends that you not use that specific SYSGEN command if you have a VAX 6000 Model 600 series system equipped with a DWMBBA/A (rather than DWMBB/A) adapter because it will take as long as 30 minutes for such a system to execute the command (due to a fundamental difference in the way the DWMBBA/A and DWMBB/A adapters report errors during probes to UNIBUS address space). In contrast, a VAX 6000 Model 600 series system equipped with a DWMBB/A adapter will take only 5 to 8 seconds to execute the SYSGEN command SHOW/UNIBUS.

2.43.4 DSSI and FDDI Boot Devices Supported on VAX 6000 Series Systems

V5.5 The following boot devices are supported on VAX 6000 series systems. For VAX 6000 Model 600 series systems, see the owner's manual for additional boot device information.

Device	Device Name	Boot Name
TF85 DSSI Tape Drive	MI	MI
RF70 and RF72 DSSI Disk Drive	DI	DI
FDDI-Based InfoServer	DAD	FX

2.43.4.1 Booting Standalone BACKUP on a VAX 6000 Series Computer from an InfoServer System

V5.5 When you boot Standalone BACKUP on a VAX 6000 series system from either an Ethernet- or FDDI-based InfoServer system, follow the revised procedure documented in the *VMS Version 5.5 Upgrade and Installation Manual* instead of the procedure currently documented in the *VMS Upgrade and Installation Supplement: VAX 6000 Series*. Note that one of the major changes in the procedure is that you must specify ISL_LVAX_055 as the boot file name instead of ISL_LVAX.

2.44 VAX 8000 Series Computer Notes

The release notes in this section pertain to VAX 8000 series computer systems.

2.44.1 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 communication port. The VMS operating system kit does not contain the DMB32 software.

2.44.2 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.

2.44.3 SET TIME/CLUSTER Command—Caution

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 hangs and eventually crashes.

2.44.4 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 date and time each time you boot the system.

2.44.5 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.

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 might 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.

System Manager Release Notes

2.44 VAX 8000 Series Computer Notes

2.44.6 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.

2.45 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 might 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 before the initial run of AUTOGEN:

```
MIN_SRPCOUNT = 3000
MIN_IRPCOUNT = 1500
MIN_LRPCOUNT = 100
MIN_MAXPROCESSCNT = 400
MIN_NPAGEDYN = 5000000
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 VIRTUALPAGECNT to be higher than 300,000, you must reduce the value of the BALSETCNT parameter.

2.46 VAX Computers—VMS Support

V5.5 Table 2-3 lists the versions of the VMS operating system and the VAX computers that these versions first supported.

Table 2-3 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

(continued on next page)

Table 2-3 (Cont.) VMS Versions and Associated VAX Computers

VMS Version	First Computer Support
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 Model 300
Version 5.4	VAXstation 3100 Model 76, VAX 9000, VAX 6000 Model 5xx, VAXft 3000
Version 5.4-1	No CPUs introduced
Version 5.4-2	VAX 4000 Model 200
Version 5.4-3	VAXft Models 410, 610, 612
Version 5.5	Microvax 3100 Models 30, 40 and 80, VAX 4000 Model 500, VAXstation 4000 VLC, VAXft Model 110, VAXstation 4000 Model 60, VAX 4000 Model 500, VAX 6000 Model 6xx

2.47 VAXstation 3520 and 3540 Computer Notes

The release notes in this section pertain to the VAXstation 3520 and 3540 computers.

2.47.1 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.

2.47.2 Software Products Availability

V5.4 Certain software products might 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.

2.48 VAXstation 4000 Series Computer—Changing Font Size

V5.5 The default font for the VAXstation 4000 series computers is 100 DPI, which is larger than the default font size for VAXstation 3100 series computers (75 DPI). If you want to change the default font size on a VAXstation 4000 series computer to the smaller size, you can do so when you reach the point in the post installation procedures (described in the *VMS Version 5.5 Upgrade and Installation Manual*) when you customize the DECwindows server startup. At that time, you can change the font size on your VAXstation 4000 series computer as follows:

1. Make a copy of SYS\$MANAGER:DECW\$PRIVATE_SERVER_SETUP.TEMPLATE and rename that file to DECW\$PRIVATE_SERVER_SETUP.COM.
2. Add the following line to DECW\$PRIVATE_SERVER_SETUP.COM:
\$ DECW\$SERVER_DENSITY == 75

System Manager Release Notes

2.48 VAXstation 4000 Series Computer—Changing Font Size

The new default font size, along with any other modifications you make to `DECW$PRIVATE_SERVER_SETUP.COM`, will be enabled when you reboot the system (which automatically restarts the DECwindows software).

2.49 VMS License Management Facility Notes

The release notes in this section pertain to the License Management Facility (LMF).

2.49.1 Increasing License Capacity

V5.5 At times you might need to increase the license capacity of a product. Vendors issue two kinds of Product Authorization Keys (PAKs) to increase license capacity: replacement PAKs and additive PAKs. PAKs are vendor specific and you should consult with your vendor if you have questions about the kind of PAK you have.

This section describes what to do when you receive these “upgrade” PAKs. However, always check with your software vendor to be sure you are complying with the terms and conditions of the license you have been granted. See the *VMS License Management Utility Manual* for more information about registering PAK.

2.49.1.1 Replacement PAKs

V5.5 For certain products, some vendors have chosen to issue replacement (noncombinable) PAKs. Replacement PAKs are PAKs whose size (in license units) reflects the new level of license usage for which you are authorized. If you receive a replacement PAK, you typically delete or disable the older PAK before registering the replacement PAK, since the replacement PAK already reflects the license authorization contained in the older PAK plus the additional number of new license units.

2.49.1.2 Additive PAKs

V5.5 Vendors of other products might choose to issue additive (combinable) PAKs. If you receive an additive PAK, you should leave the older PAK enabled because the LMF automatically adds (or combines) the license usage of the additive PAK to the license usage of the older PAK in order to establish your new level of license authorization.

In most cases, additive PAKs represent new, independent licenses, which work with your old license to provide a higher level of license authorization; whereas replacement PAKs represent an upgrade to an existing license.

2.49.1.3 PAKs with the NOSHARE option

V5.5 With this release of the LMF, the behavior of the NOSHARE option has changed. PAKs issued with the NOSHARE option can now be combined by the LMF; previously they could not be combined.

Previously, all NOSHARE PAKs were replacement PAKs; however, now vendors may choose to issue additive NOSHARE PAKs, as well as replacement NOSHARE PAKs. If you are unsure what kind of PAK you have, contact the vendor who issued the PAK to you.

2.49.2 List Size Restrictions

V5.5 Due to a limitation in the LMF, two restrictions apply to the size of lists (reserve lists, include lists, or exclude lists). These restrictions apply to PAKs of all license types (availability, activity, and reserved user licenses).

System Manager Release Notes

2.49 VMS License Management Facility Notes

2.49.2.1 The Per PAK Limit = 5000 Characters

V5.5

On any single PAK, the sum of characters contained in all lists must not exceed 5000 characters.

Because the length of names vary and some overhead is used for each name, this 5000 character limit cannot be expressed as an exact number of permissible names. However, Digital guarantees that at least 400 names, in total, can be specified in the various types of list. For example, each of the following represents the minimally guaranteed number of names:

- Reserve list with up to 400 user names
- Reserve list with up to 200 user names plus an include list with up to 200 node names (totalling up to 400)
- Reserve list with up to 200 user names plus an exclude list with up to 200 node names (totalling up to 400)
- Include list with up to 400 node names
- Exclude list with up to 400 node names

Note

If you enter more names than are permitted, the LICENSE LIST command might not be able to display all names entered. In this case, you will receive the error message LICENSE-F-CORRUP. However, the license database is not actually corrupt and the PAKs can still be loaded in memory (though the names are not displayed).

2.49.2.2 The Per Product Limit = 30000 Characters

V5.5

The LICENSE LOAD and LICENSE START commands can load into memory a reserve list with no more 30,000 characters. (Include and exclude lists, which are not loaded into memory, are irrelevant to the 30,000 character limit.)

Because the length of names vary and some overhead is used for each name, this 30,000 character limit cannot be expressed as an exact number of permissible names. But Digital guarantees that, for each product, at least 2000 user names can appear on reserve lists. In the case of a VAXcluster, this is a per-node limit.

Note that, because 2000 user names is a per-product limit and because there can be more than one PAK per product, the number of user names on a per-product basis is the *sum* of the user names specified on each PAK.

For example, if three activity PAKs for the product DECwrite were registered on a system and each PAK specified a reserve list with 200 user names, the total number of user names for that product is 600. This is safely below the 30,000 character (2000 user name) limit and below the 5,000 character limit (400 user name) limit.

Caution

If the 30,000 character limit is exceeded, the LMF truncates the reserve list that is loaded into memory; no error message is issued.

System Manager Release Notes

2.49 VMS License Management Facility Notes

See Section 9, License Combination, in the *VMS License Management Utility Manual* for more information on how the LMF combines PAKs for the same product.

2.49.3 Logical Name LMF\$DISPLAY_OPCOM_MESSAGE

V5.5 A previous version of the *VMS License Management Utility Manual* incorrectly stated that you must define the logical name LMF\$DISPLAY_OPCOM_MESSAGE in order for messages to appear. This statement has been removed from the new version of the *VMS License Management Utility Manual*.

If you have already defined this logical name, you should delete the definition.

Define the LMF\$DISPLAY_OPCOM_MESSAGE logical name only if you are explicitly directed by Digital to do so (for debugging purposes). When defined, this logical name causes many “noise” messages to be displayed on the operator’s console; some of these messages can be confusing and misleading to the point of suggesting that a product is not licensed, when in fact it is.

To see if this logical name has been defined on your system, enter the following command:

```
$ SHOW LOGICAL LMF$DISPLAY_OPCOM_MESSAGE
```

To delete the logical name, enter the following command:

```
$ DEASSIGN LMF$DISPLAY_OPCOM MESSAGE
```

2.49.4 Preserving License Audit Trails

V5.5 The LMF writes a history record to the license database every time you modify or delete a PAK. Each history record contains an exact copy of the license record before modification, the LICENSE command used to modify the record, and the date and time of modification.

These history records accumulate over time and provide a comprehensive audit trail of all modifications you make to the license database. Most software issuers, including Digital Equipment Corporation, require that you retain this information to demonstrate that you are complying with license terms and conditions.

At any time, you can display this information by executing the following command:

```
$ LICENSE LIST /HISTORY
```

Or, you can create a hardcopy by entering the following:

```
$ LICENSE LIST /HISTORY /OUTPUT=LICENSE.LIS  
$ PRINT LICENSE.LIS
```

You can also preserve your license history records by using the DCL command COPY or the BACKUP Utility to make a copy of the license database.

2.49.5 Purging License History Records

V5.5 If, over time, you notice that LICENSE commands (including the LICENSE START command which is issued automatically during system startup) are taking longer than usual to execute, this could be due to an accumulation of license history records in the license database.

If you notice any such delays, Digital recommends that you purge the history records in your active license databases, but only after first preserving this information in one or more backup locations. Section 2.49.4 describes several ways to preserve this history information.

System Manager Release Notes

2.49 VMS License Management Facility Notes

To purge the history records in your active license database, enter the following command:

```
$ LICENSE DELETE /STATUS=EXTINCT *
```

Note

Be certain that you do not omit the /STATUS=EXTINCT qualifier in the above command; if you do, the license records themselves will be deleted and your license database will be unusable.

The LICENSE DELETE command will improve performance degradation caused by an excessive number of history records in the license database. It does this by marking all history records for deletion, thus making them invisible to any subsequent LICENSE command.

Additionally, you might want to use the CONVERT utility to create a new, compressed version of the license database, thus reclaiming the disk space formerly occupied by the (now deleted) history records. To do this, enter the following command:

```
$ CONVERT license_database_name license_database_name
```

This command creates a new, compressed license database with a higher version number than the previous license database. The previous version of the license database can now be deleted (for example, by using the DCL command PURGE), but not before ensuring that you have an adequate backup copy as described in Section 2.49.4.

2.49.6 Installation Considerations for VAXclusters

V5.5

Digital recommends that you install VMS Version 5.5 on all system disks in a VAXcluster system. Doing this ensures that all nodes in the VAXcluster are running the latest version of the LMF.

If you upgrade some but not all system disks in a VAXcluster system, you will be running what is called a "mixed version VAXcluster." Some nodes in the VAXcluster are running the old LMF, and some are running the new VMS Version 5.5 LMF.

In this case, you must separate your license databases so that an old LMF can never read or write to a license database that has been modified by the new LMF. However, you can use the LICENSE MOVE and LICENSE COPY features of the new LMF to move and copy PAKs among all license databases and still access those databases with the old LMF.

Figure 2-1 depicts a mixed version VAXcluster in which one node is running the new VMS Version 5.5 LMF and the other node is running the old LMF. Since the license databases (LDBs) used by each node are separate, no problems should arise.

Caution

Figure 2-2 depicts a mixed version VAXcluster in which two nodes running different versions of the LMF are sharing the same license database. This configuration could bring about corruption of the license database (LDB), as well as instances of denial of service. If corruption occurs, you will receive the LMF message CORRUPREC. See the VMS

System Manager Release Notes

2.49 VMS License Management Facility Notes

License Management Utility Manual for a description of this error message.

Figure 2-1 Valid Mixed Version VAXcluster with Separate License Databases

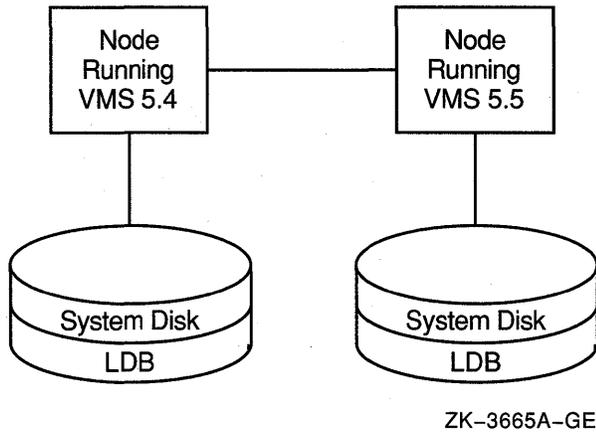
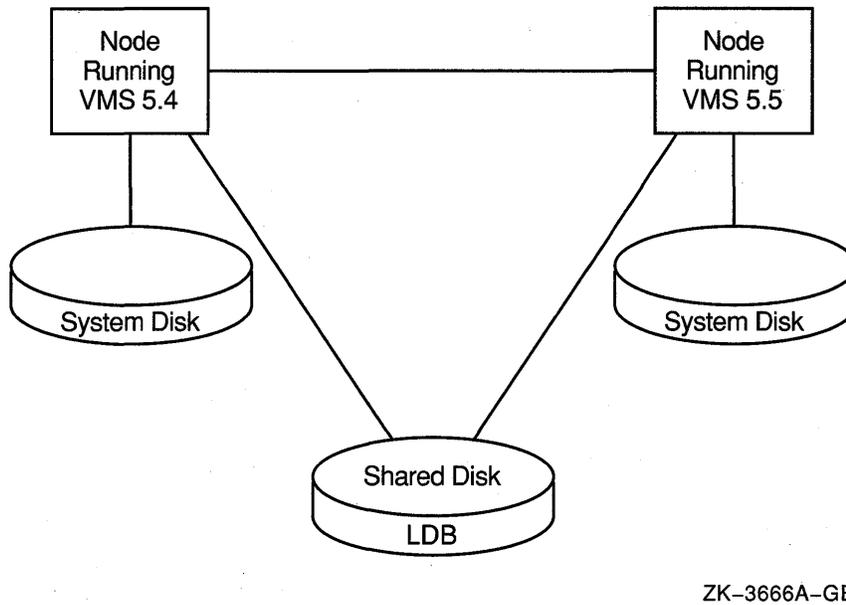


Figure 2-2 Invalid Mixed Version VAXcluster with Shared License Database



2.50 VMS VAXcluster Notes

The release notes in this section pertain to VMS VAXcluster software.

2.50.1 Large VAXcluster Configurations—SYSGEN Parameter SCSCONNCNT

V5.5

The SYSGEN parameter SCSCONNCNT should be set to 7 times the sum of the number of VAX processors plus the number of HSCs.

$SCSCONNCNT = 7 * (VAX\ processors + HSCs)$

The parameter should be set on all VAX nodes in the cluster. This parameter can be modified by specifying the parameter value in the MODPARAMS.DAT file and then running AUTOGEN.

If the value for SCSCONNCNT is too low, the cluster could have the following symptoms:

- Nodes will not join the cluster.
- The IPC connection will not form between nodes, affecting the queue manager and DECdtm.
- Remotely served disks and tapes might not be seen.

2.50.2 Mixing a Multiadapter Local Area VAXcluster and VMS Version 5.4-2 or Earlier

V5.4-3

VMS Version 5.4-3 and 5.5 multiadapter local area VAXcluster support is compatible with VMS Version 5.4 through VMS Version 5.4-2. In clusters containing systems running VMS Version 5.4-3 or 5.5 and systems running versions prior to VMS Version 5.4-3, if any Version 5.4-3 or 5.5 systems have multiple adapters, all systems display additional %CNXMAN messages on the console.

Prior to the multiple LAN adapter support, the channel control handshake messages were used to perform a handshake for the virtual circuit. Each time the channel control handshake messages were received, the virtual circuit was broken and then reformed.

Now, the channel control handshake messages are used to perform a handshake for the channel. The virtual circuit can form only after at least one channel is open to the remote system. If a single LAN adapter is used in the system running VMS Version 5.4-3 or later, there is no difference in how the virtual circuit forms.

However, if the system running VMS Version 5.4-3 uses multiple adapters, the virtual circuits to the systems running VMS Version 5.4-2 or earlier break and reform when the additional adapters start. This causes additional console messages to be displayed at the following times:

- When additional adapters start
- When a channel breaks and then reforms (for example, when a network component fails)
- Following a rolling upgrade from a version prior to VMS Version 5.4-3, when DECnet is started, all other nodes running VMS Version 5.4-3 or later might display the console messages until they are rebooted

The following are examples of the console messages:

```
%CNXMAN, Lost connection to system NODE1  
%CNXMAN, Re-established connection to system NODE1
```

When VMS Version 5.4-3 or later is running on all the cluster systems, the console displays these messages only when the virtual circuit fails. This matches the current behavior of VMS Version 5.4-2 and earlier releases of VMS.

System Manager Release Notes

2.50 VMS VAXcluster Notes

2.50.3 Selecting Boot Disk Path

V5.5 It is possible for node that has a direct path to its system disk to be configured as a satellite and have a served path to the disk. For example, in a dual host DSSI VAXcluster configuration, a host could boot a DSSI system disk directly via its DSSI adapter or be configured as a satellite of the other host and boot (via its Ethernet adapter) through the other host off the same disk.

When booting nodes that have local and served paths to their system disk, the direct path must be used if functional. If the direct path is not functional at boot time, the served path can be used. If a node is booted using a served path when a functional direct path is available, failover to the direct path will not occur. As a result, the node will not properly function as a disk server, potentially causing random satellite boot failures.

2.50.4 SCS SYSGEN/TIMVCFAIL Parameter

V5.5 The SCS SYSGEN parameter TIMVCFAIL specifies the time required for an adapter or virtual circuit failure to be detected. Digital recommends that the default value be used. Digital also recommends that this value be lowered only in VAXclusters of three CPUs or less, that the same value be used on each computer in the cluster, and that dedicated LAN segments be used for cluster I/O.

2.50.5 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.

2.50.6 VAXcluster Support for FDDI Adapter

V5.4-3 VMS Version 5.4-3 provides VAXcluster support for the Fiber Distributed Data Interface (FDDI) adapter—the DEC FDDIcontroller 400, known as the DEMFA. The DEMFA can be used on VAX 6000 and 9000 series systems. Configurations that include the DEMFA must follow the configuration rules and restrictions, as specified in the VAXcluster Software Product Description (SPD 29.78.05).

2.50.7 VAXcluster Support for Multiple LAN Adapters

V5.4-3 Previous versions of VMS allowed the use of only one LAN adapter for each local area VAXcluster system or node. VMS Version 5.4-3 and 5.5 supports up to four Local Area Network (LAN) adapters on each local area VAXcluster system. Adapters can be either Ethernet or FDDI. Table 2-4 lists the supported adapters.

Table 2-4 Ethernet and FDDI Adapters

Bus	Ethernet Adapters	FDDI Adapter
XMI	DEMNA	DEMFA
BI	DEBNA, DEBNI	
Qbus	DELQA, DESQA, DEQTA (DELQA-YM)	
Unibus	DEUNA, DELUA	
Integral	LANCE, SGEC	

Guidelines to VAXcluster System Configurations provides guidelines for configuring a VAXcluster system with multiple LAN adapters.

VAXcluster support for multiple LAN adapters allows PEDRIVER (the VAXport driver for the Local Area Network) to establish more than one channel to support the virtual circuit between the local and remote cluster nodes. A channel is a unique network path between two nodes that is represented by a pair of LAN addresses or LAN adapters. VAXcluster configurations with multiple LAN adapters have the following characteristics:

- At boot time, *all* Ethernet and FDDI adapters are automatically configured for VAXcluster use.
- PEDRIVER automatically detects and creates a new channel for each unique pair of LAN adapters between the local node and each remote cluster node.
- Channel viability is continuously monitored every three seconds, at a minimum.
- Channel failure does not interfere with node-to-node (virtual circuit) communications as long as there is at least one remaining functioning channel between the nodes.

2.50.8 VAXcluster Use of DEMNA

V5.5 VMS now provides a slight performance benefit with all Ethernet adapters. To get this performance benefit with the DEC LANcontroller 400 (DEMNA), you should use DEMNA firmware revision level Version 6.06 or later. This adapter is supported for 6000 and 9000 class machines and has a device name prefix of EX, for example, EXA0:

To display the DEMNA's firmware revision level, enter the following console command:

```
>>> SHOW CONFIG
```

To obtain a later revision level version of the DEMNA firmware, contact the Digital Services Group.

2.51 Volume Shadowing (Phase II) Notes

The release notes in this section pertain to VMS Volume Shadowing (Phase II).

System Manager Release Notes

2.51 Volume Shadowing (Phase II) Notes

2.51.1 Overview

V5.4-2

VMS Volume shadowing (Phase I) provides centralized shadowing on VMS systems that use HSC (Hierarchical Storage Controller) disks with identical DIGITAL Storage Architecture (DSA) disks. VMS Volume Shadowing Phase II is not limited to HSC-controlled disks; rather, it extends volume shadowing capabilities to the following:

- All DSA disks, including those on local adapters
- All DSSI (DIGITAL Small Systems Interconnect) RF-series disk drives on any VAX computer
- Disks served by the VMS mass storage control protocol (MSCP) server and located anywhere in any supported VAXcluster configuration

For more information, refer to the VMS Volume Shadowing SPD (27.29.07).

For VMS Version 5.4-2, VMS Volume Shadowing (Phase II) is limited to 32 shadow sets in a VAXcluster system. The configuration rules for VAX Volume Shadowing Phase I are specified in the VAX Volume Shadowing SPD.

2.51.2 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 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.

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.

2.51.3 Changing Physical Device Names

V5.4-2

VMS Volume Shadowing (Phase II) records the physical member device names in an on-disk data structure. This information is used for various purposes by subsequent MOUNT commands. A side effect of storing this information is that, even though a shadow set is dismounted across the cluster, changing the names of the data disks through techniques such as swapping unit plugs can cause subsequent MOUNT commands to fail. Use one of the following supported methods for changing the physical device names when the shadow set is dissolved and then remount the set:

- Change the virtual unit name when you recreate the shadow set.

System Manager Release Notes 2.51 Volume Shadowing (Phase II) Notes

- Mount the disks using the /OVERRIDE=SHADOW qualifier before recreating the set.

2.51.4 Corrections

The following problems are corrected in VMS Volume Shadowing (Phase II) for VMS Version 5.4-3.

2.51.4.1 SHOW DEVICES Display Is Consistent with Shadow Set Membership Status

V5.4-3

Prior to VMS Version 5.4-3, the SHOW DEVICES command would display erroneous information about the shadow set after an unsuccessful attempt to mount a shadow set. This problem was triggered when, during an attempt to mount a shadow set, a disk went off line and the shadow set could not be mounted. Even though the shadow set did not exist on the node, a subsequent SHOW DEVICES command would display erroneous information about the physical devices related to the shadow set.

This problem has been corrected in VMS Version 5.4-3.

2.51.4.2 Volume Shadowing Merge Copy Progresses Successfully

V5.4-3

Prior to VMS Version 5.4-3, a VMS Volume Shadowing (Phase II) merge copy might occasionally stall.

Even though a SHOW DEVICES command indicated that the merge copy had started, the merge copy would not progress. The SHOW DEVICES command would display the status of the merge copy as being 0% merged, as shown in Example 2-2.

Example 2-2 SHOW DEVICES Display

```
$ SHOW DEVICES
```

Device Name	Device Status	Error Count	Volume Label	Free Blocks	Trans Count	Mnt Cnt
DSA0:	Mounted	0	V54	292269	215	3
DSA2000:	Mounted	0	X4AU	292971	1	1
\$1\$DUA0:	(WHEN) Online	0	(remote shadow member)			
\$1\$DUA2:	(WHEN) ShadowMergeMbr	0	(merging DSA0: 0% merged)			
\$1\$DUA3:	(WHY) ShadowMergeMbr	0	(merging DSA0: 0% merged)			
\$1\$DUA1:	(WWWWW) Online	0				

This problem has been fixed in VMS Version 5.4-3.

Note

There are normal cases where merge copy operations are deferred and might seem to be stalled, such as those following:

- The number of current copies on a node exceeds the number specified by the SYSGEN parameter SHADOW_MAX_COPY, and there is no other node that can do the copy immediately.
 - A merging volume is under heavy load.
-

System Manager Release Notes

2.51 Volume Shadowing (Phase II) Notes

In addition, VMS Volume Shadowing (Phase II) has improved the merge copy algorithm such that it can dynamically throttle itself to obtain the best application I/O performance at the same time it efficiently executes the merge copy to bring the shadow set to a consistent state.

2.51.5 MOUNT/CLUSTER Command

V5.5 Using the MOUNT/CLUSTER command to simultaneously add new members to an existing shadow set and also to newly mount the set on other nodes **requires** that all source members be stipulated.

Do *not* use the MOUNT/CLUSTER qualifier when adding a member to an existing Volume Shadowing Phase II virtual unit that is not mounted on all nodes in the cluster.

2.51.6 SHADBOOTFAIL Bugcheck Message

V5.4-2 If you are using VMS Volume Shadowing (Phase II) and you get a SHADBOOTFAIL bugcheck message, you might not be able to access the dump if further analysis is required. Shadowing dump-file processing utilizes merge operations to perform a dump fixup. Because the set was never mounted on this node, a merge operation is not triggered.

Refer to Appendix A in the *VMS Volume Shadowing Manual* for more information about why the shadow set failed to boot from the system disk shadow set.

2.51.7 Shutting Down a System with a VMS Volume Shadowing (Phase II) Shadowed System Disk

V5.4-3 SHUTDOWN.COM and OPCCRASH.EXE require PHY_IO privilege to execute correctly if you have a VMS Volume Shadowing (Phase II) shadowed system disk.

If the process running SHUTDOWN.COM does not have PHY_IO privilege or any other required privileges, the command procedure will print a list of required privileges and stop. If the process running OPCCRASH.EXE does not have PHY_IO privilege, the dismount of a VMS Volume Shadowing (Phase II) shadowed system disk will be incomplete, triggering a shadow set merge operation on the remaining nodes that have the shadow set mounted. Note that a merge operation does not affect data correctness. It only consumes some system bandwidth.

If your process does not have the PHY_IO privilege currently, refer to the *VMS DCL Dictionary* for information about the DCL command SET PROCESS /PRIVILEGE, or contact your system manager.

2.51.8 Obtaining \$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.

2.51.9 SYSGEN Parameter VIRTUALPAGECOUNT Adjustment

V5.4-1 In general, VMS Volume Shadowing can perform full copy and merge operations without any changes to SYSGEN parameter values. In some cases, however, it might be necessary to adjust the values of the SYSGEN parameters VIRTUALPAGECOUNT and WSMAX to ensure that the volume shadowing server process can develop sufficient virtual address space to do copy and merge operations.

System Manager Release Notes 2.51 Volume Shadowing (Phase II) Notes

To determine whether the VIRTUALPAGECOUNT and WSMAX parameters need adjustment, compare the current values of these parameters (use the SYSGEN command SHOW to examine the parameter values) with the result of the following equation that uses the current value of the SYSGEN parameter SHADOW_MAX_COPY:

$$(\text{SHADOW_MAX_COPY} * 128) + 100$$

For example, if a system has a SHADOW_MAX_COPY parameter of 42 (the default value of the SHADOW_MAX_COPY parameter is 4), then the SYSGEN parameters VIRTUALPAGECOUNT and WSMAX should be no less than $(42 * 128) + 100$, or 5476 pages.

2.51.10 UAF Parameter Changes

V5.4-1 For VMS Volume Shadowing Phase II, the user authorization file (UAF) parameter changes listed in Table 2-5 are recommended for your system account (these are the best minimum estimates for the shadow server):

Table 2-5 Recommended UAF Parameters for the System Account

BYTLM	SHADOW_MAX_COPY * 3KB
BIOLM	SHADOW_MAX_COPY + 10
DIOLM	SHADOW_MAX_COPY + 10
ASTLM	SHADOW_MAX_COPY * 2 + 20
TQELM	SHADOW_MAX_COPY * 2 + 10
WSQUO	SHADOW_MAX_COPY * 128 + 100
WSDEF	SHADOW_MAX_COPY * 128/2 + 100
ENQLM	SHADOW_MAX_COPY * 2
PGFLQUO	SHADOW_MAX_COPY * 130 + 500

2.52 Working Set Quota (WSQUOTA) Size Limited

V5.4-3 If the Working Set Quota (WSQUOTA) size is greater than 65536 pages and VMS attempts to outswap a process, the system can crash with a bad page file address allocated. To prevent this, make sure that the WSQUOTA does not exceed 65000 pages. You can set the WSQUOTA value within the Authorize Utility by entering the following commands:

```
$ SET DEF SYS$SYSTEM
$ RUN AUTHORIZE
UAF> SET [user_name]/WSQUOTA=65000
UAF> EXIT
```

Note

For this change to take place, each user logged in with this user name must log out and then log back in.

For processes with WSQUOTA values greater than 65000 pages, it is recommended that you make these processes nonswappable by entering the following command or by using the set swap mode system service \$SETSWM:

System Manager Release Notes

2.52 Working Set Quota (WSQUOTA) Size Limited

§ SET PROCESS/NOSWAP

Note

You need PSWAPM privilege in order to disable swapping for your process.

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

2.53 YFDRIIVER Terminal Port Driver Supports New Baud Rate

V5.3-2

With VMS Version 5.3-2, YFDRIIVER 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.

Programmer Release Notes

This chapter includes information about the VMS Version 5.5 operating system that is of interest to both the application and system programmer.

For information about the new features included in VMS Version 5.5, see the *VMS Version 5.5 New Features Manual*.

3.1 VMS Version 5.5 Specific Release Notes for Programmers

The release notes in this manual are cumulative from VMS Version 5.0. The following sections contain programmer release notes that pertain specifically to VMS Version 5.5:

- Section 3.6.2.1—DEPOSIT/TYPE Command with C Programs
- Section 3.9—DECthreads Restriction
- Section 3.11—Disk Forced Error Counting Implemented
- Section 3.15—LAT Release Notes
- Section 3.16—Lock Manager Notes
- Section 3.17—Linker Utility
- Section 3.32.2—Mailbox Driver Error Checking
- Section 3.32.3—\$GETQUI—QUI\$_QUEUE_STATUS Longword
- Section 3.32.4—\$GETSYI System Service: VAX Computer Model Numbers Added to Item Codes
- Section 3.32.6—SJC\$_START_QUEUE_MANAGER Function Code for \$SNDJBC Service
- Section 3.32.7—\$SNDJBC Service—Obsolete Item Codes
- Section 3.33.1.1—Booting Post-VMS Version 5.4 Computer Systems

3.2 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
```

Programmer Release Notes

3.2 Activating an Image with System Version Mismatch—Change

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.

3.3 BASIC INKEY\$ Function—Change

V5.4-1

In previous versions of VMS, the first VAX BASIC INKEY\$ function left the application keypad in application mode until the end of the program. Starting with VMS Version 5.4-1, after the program executes the INKEY\$ function, the application keypad stays in numeric mode if it was in numeric mode before the program started.

This change also affects BASIC INPUT, INPUT LINE, and LINPUT statements that a program executes after the INKEY\$ function. Starting with VMS Version 5.4-1, when the program executes an INPUT, INPUT LINE, or LINPUT statement after an INKEY\$ function, the application keypad stays in numeric mode if it was in numeric mode before the program started.

3.4 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:

```
$ 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:

3.4 DCL Subroutine—Modifications to Subroutine Entry Point Scoping

```
$ 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
```

3.5 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.

3.6 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.

3.6.1 Corrected Problems or Restrictions

The following problems or restrictions in previous versions of the debugger have been corrected:

- V5.3
- In VMS Version 5.3
 - The RUN/DETACHED command entered after the LINK/DEBUG command did not work correctly. If you linked a program using the command LINK/DEBUG and then executed the program in a detached process (using the command RUN/DETACHED), the debugger went into an infinite loop. This problem has been fixed in this release.
- V5.4
- In VMS Version 5.4
 - The following source-line correlation problem affected the debugging of MACRO programs that invoke the \$FAO or \$FAO_S system service macro or an application-defined macro that contained any of the following directives:

```
.IRP
.IRPC
.REPT
```

Programmer Release Notes

3.6 Debugger Notes

.REPEAT

This problem caused the debug symbol table information about source lines and line numbers to be unreliable. The screen-mode source display (SRC) was not updated correctly when the debugger suspended execution. This problem has been corrected.

- The creation and sorting of the Static Address Table (SAT) at debugger startup was slow. The creation and sorting of the SAT has been improved. Startup time for very large programs is now faster.
- In large Ada programs, the commands SET MODULE/ALL or SET MODULE followed by SET BREAK caused ROPERAND faults. This problem has been corrected; the SET MODULE and SET BREAK commands now work correctly with large Ada programs.
- Pressing Ctrl/C did not abort the display of large amounts of watchpoint information. This problem has been corrected; pressing Ctrl/C now aborts the display of large amounts of watchpoint information.
- Support for Ada descriptors was inadequate. This problem has been corrected; Ada extended descriptors are now fully supported.
- In a C expression, the NOT (^) operator did not have the correct precedence and, as a result, the EVALUATE command did not evaluate expressions correctly. This problem has been corrected; the EVALUATE command now evaluates expressions containing the NOT operator correctly.
- At debugger startup with very large programs, %DEBUG-I-INVDMTPTR and %DEBUG-I-MISMODOEND errors sometimes occurred, and the program could not be debugged. This was a problem in both the linker and the debugger. The problem has been corrected; these startup errors no longer occur.

3.6.2 Debugger Problems or Restrictions

This section describes problems or restrictions with the debugger.

3.6.2.1 DEPOSIT/TYPE Command with C Programs

V5.5

When debugging a C program, you cannot use the DEPOSIT/TYPE command if the type specified is a mixed case or lowercase name. For example, suppose the program has a function like the following:

```
xyzzy_type foo ()
{
  xyzzy_type z;
  z = get_z ();
  return (z);
}
```

If you try to enter the following command, the debugger issues a message that it cannot find the type "xyzzy_type":

```
DBG> DEPOSIT/TYPE=(xyzzy_type) z = "whatever"
```

3.6.2.2 \$WAKE Call Followed by \$HIBER Call

V5.3

If a program, running under the two process or multiprocess debugger, issues a \$WAKE call followed by a \$HIBER call, the user application hibernates.

3.6.2.3 Examining LABEL[n] for a Code Location

V5.4 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.

3.6.2.4 Register Window in DECwindows Interface

V5.3 When using the debugger's DECwindows interface, you might 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.

3.6.2.5 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 might receive an internal error during the processing of a SET IMAGE command. In such cases, the image cannot be debugged.

3.6.2.6 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 following syntax:

```
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 command SET SOURCE to specify the new location of the source file.

3.6.2.7 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

Programmer Release Notes

3.6 Debugger Notes

point to that file. For example:

```
$ CREATE DEBUG COMMANDS.TMP
SET BREAK %LINE 17
GO
EXIT
CtrlZ
$ 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 3.6.3.

3.6.2.8 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. The debugger cannot re-enable these functions until it receives a signal back from the VMS operating system. At this time, the VMS operating system is not sending that signal.

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.

3.6.2.9 Using the Debugger on a VAXstation Running VWS

V5.0

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.

3.6.2.10 Using the DECwindows Interface

V5.2

The following problems or restrictions are specific to the VMS 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 can take up to a minute to display the first help topic. Subsequent help topics are displayed within a few seconds after you invoke them.

Programmer Release Notes

3.6 Debugger Notes

- 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.
 - 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.

Programmer Release Notes

3.6 Debugger Notes

- 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 might 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.)

3.6.2.11 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.

3.6.2.12 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"
```

3.6.3 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 3.6.2.8 and Section 3.6.2.7. If you want to use the single-process debugging configuration because it avoids the restrictions described in Section 3.6.2.8 and Section 3.6.2.7, 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 3.6.2.8 and Section 3.6.2.7), use the single-process configuration (established when the definition of `DBG$PROCESS` is `NONE`) only when necessary. The single-process configuration is unsupported and might 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).

Programmer Release Notes

3.7 DEC CDA Base Services

3.7 DEC CDA Base Services

The release notes in this section pertain to DEC CDA Base Services. The product pertains to the compound documentation architecture.

3.7.1 User Get and Put Routine Changes

V5.4-3

Applications that call CdaConvert can provide their own user get and put routines to perform stream I/O. Applications provide their own get and put routines by using the item-list items CDA\$_INPUT_PROCEDURE or CDA\$_OUTPUT_PROCEDURE. They can also specify an associated parameter value (via item-list items CDA\$_INPUT_PROCEDURE_PARM and CDA\$_OUTPUT_PROCEDURE_PARM) to be passed to the get or put routine.

Using the CDA Base Services shipped with VMS Version 5.4, applications that converted to or from DDIF or DTIF format and that provided user get or put routines were required to pass the actual value of the parameter in the item list. This was a change in behavior from the previous release of the CDA Base Services.

With VMS Version 5.4-3, applications must encode the associated parameter item as the address of a longword containing the value that is to be passed to the get or put routine. In other words, applications must provide an additional level of indirection, since the DDIF Front and Back Ends will dereference the item-list item before calling the user get or put routine.

As a result of this change, applications that call the CDA Viewer routine DvrViewerFile will also require similar recoding if they provide callback routine and parameter arguments to DvrViewerFile.

3.7.2 Sending Compound Documents with PostScript/Text References

V5.4-3

Using DECwindows Mail, you can now successfully mail compound documents that contain external references (links) to files containing text or PostScript.

3.8 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 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.

3.9 DECthreads Restriction

V5.5

With VMS Version 5.5, there are no language-specific include files that provide DECthreads interfaces and constants for any language except C. If you call DECthreads routines from a language other than C, you must explicitly declare the routines and datatypes.

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

3.10 DECwindows Notes

The release notes in this section pertain to the VMS DECwindows software supplied with VMS Version 5.5. Applications included in VMS Version 5.5 are identical to those in Version 5.4. These release notes do not apply to new applications using the Motif interface, which are available with the VMS DECwindows Motif layered product.

3.10.1 DECterm Terminal Emulator

The following release notes contain information about the DECterm terminal emulator.

3.10.1.1 Color Table Report—Reporting Problem Corrected

V5.4

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>
```

3.10.1.2 CREATE/TERMINAL Command—Negative Values Problem Corrected

V5.4

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.

3.10.1.3 ReGIS Locator Report—Omitted Coordinates

V5.4

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>
```

3.10.1.4 VT52-Mode Cursor Addressing—Restriction Removed

V5.4

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.

3.10.2 DECwindows Server and Driver—General

The release notes in this section pertain to DECwindows server and driver.

Programmer Release Notes

3.10 DECwindows Notes

3.10.2.1 Misspelled Cursor Screen Boundary \$QIO Constant

V5.3-1 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 \$DECWGGBL 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.

3.10.2.2 Problems and Restrictions

V5.3 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 might be time-stamped with the current time during replay instead of the time the event was collected.

3.10.2.3 Setting a Cursor Pattern \$QIO Call

V5.3-1 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*.

3.10.2.4 Virtual Memory Space Problem with Large Pixmaps

V5.4 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.

3.10.2.5 VAXstation Configurations

V5.1

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.
- 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 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 this type of problem, it sends an “Implementation” error to the client or 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 might 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 can 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 might 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.

Programmer Release Notes

3.10 DECwindows Notes

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.

3.10.3 DECwindows X11 Display Server

This section contains corrections, problems and changes pertaining to the DECwindows X11 Display Server.

3.10.3.1 PostScript (DPS) Extension Correction

V5.4-3

Prior to VMS Version 5.4-3, the server DPS extension did not return error status to the CDA Viewer when a corrupt PostScript file was sent to the extension. Consequently, the CDA Viewer would appear to hang.

This problem has been corrected. When the server DPS extension detects a corrupt PostScript file as input, it notifies the client application with an error status. Then the client application can display the diagnostic message on the screen.

3.10.3.2 Problem Updating Cursor Colors

V5.4-3

On a GPX system, if you run a new server with an old driver, the cursor colors are not updated. To update the cursor colors, you must reboot your system to load the new driver.

3.10.3.3 Problem Accessing Layered Products Fonts

V5.4-3

If you want to use any layered products that supply their own X fonts, your system manager must invoke the DECW\$MKFONTDIR command file in order for the DECwindows X11 Display Server to be able to access the layered products' fonts. The system manager will need to invoke this command file only for layered products installed after DECwindows is installed because the DECwindows installation automatically invokes DECW\$MKFONTDIR.

The system manager should enter the following command:

```
$ @SYSSUPDATE:DECW$MKFONTDIR
```

After the system manager enters this command, you must end the current session and start a new session before the server can access the fonts.

This is necessary because the DECwindows X11 Display Server now supports font directory files for faster startup. Fonts supplied by the DECwindows kit come with prebuilt font directory files. However, layered products do not supply font directory files with their fonts.

This restriction will be lifted in the future as layered products are modified to invoke the DECW\$MKFONTDIR command file in their kit installation procedures.

3.10.3.4 Narrow Terminal Fonts—Increased Height

V5.4-3

The Terminal family of fonts has been modified to be more consistent in its spacing of characters. In particular, the font ascent of the 14-point 75 dots/inch narrow (condensed) and 10-point 100 dots/inch narrow fonts has been increased by one pixel. This means that DECterm can display fewer lines of text for the same amount of space than before when using the Condensed and Little Font settings together.

If you have an application that depends on the previous vertical spacing of the Terminal fonts (for instance, to fill in the entire screen with text), the workaround is to copy the Terminal fonts from VMS Version 5.4, 5.4-1, or 5.4-2 (depending upon which version you are upgrading from) to the USER_75DPI and USER_100DPI font directories. Execute the following commands before upgrading:

```
$ COPY SYS$COMMON:[SYSFONT.DECW.75DPI]TERM*.DECW$FONT -  
_ $ SYS$COMMON:[SYSFONT.DECW.USER_75DPI]  
$ COPY SYS$COMMON:[SYSFONT.DECW.100DPI]TERM*.DECW$FONT -  
_ $ SYS$COMMON:[SYSFONT.DECW.USER_100DPI]
```

If you copy the fonts to the user font directories after upgrading, you must also execute the command file SYS\$UPDATE:DECW\$MKFONTDIR in order to use the new fonts. This command file is automatically executed by the VMS Version 5.4-3 update procedure.

3.10.4 Display PostScript Notes

The following release notes pertain to Display PostScript.

3.10.4.1 Display PostScript Documentation

V5.4-1

Actual colors are colors that the Display PostScript (DPS) server allocates as a result of either of two actions:

- Passing a non-zero **actual** parameter to the CREATE CONTEXT routine or to the setXgcdrawablecolor PostScript operator
- Using the setXrgbactual PostScript operator

DPS always allocates actual colors as shared. However, if you create a colormap with the AllocAll flag, all entries in the colormap are allocated as nonshareable. Therefore, you cannot use actual colors in DPS with a colormap that you created with the AllocAll flag.

3.10.4.2 Ada Bindings Not Available

V5.4

With VMS Version 5.4 and subsequent versions, Ada bindings are not available for Display PostScript.

3.10.4.3 VAX Calling Standard Bindings—DPS\$PRINTF Routine Not Implemented

V5.4

In the VAX Calling Standard Bindings, the DPS\$PRINTF routine is documented but not implemented.

3.10.5 SET DISPLAY Command—Restriction

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, enter the following command:

```
$ SET DISPLAY/CREATE/TRANSPORT=TCPIP/NODE="uuu"
```

The quotation marks are optional for a node name composed of all uppercase characters.

Programmer Release Notes

3.10 DECwindows Notes

3.10.6 User Interface Language (UIL) Compiler Notes

The following release notes contain information about the User Interface Language (UIL) compiler.

3.10.6.1 Built-In Tables—Additions

V5.4 The built-in UIL arguments listed in Table 3–1 were added for VMS Version 5.4.

Table 3–1 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, hls_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
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

3.10.6.2 Corrected Problems

V5.4 With VMS Version 5.4, the following UIL problems were 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 DRMFatchWidget call to return with a DRMNNotFound 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.
- 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.

3.10.6.3 Specifying Multiline Compound Strings

V5.3

In VMS Version 5.3 and subsequent versions, 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 might 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 about the COMPOUND_STRING function. See the *VMS DECwindows User Interface Language Reference Manual* for more information about the /VERSION qualifier.

Programmer Release Notes

3.10 DECwindows Notes

3.10.7 XUI Toolkit Notes

This section contains release notes that pertain to the XUI Toolkit.

3.10.7.1 DRM Routines—Unavailable VAX Bindings

V5.4 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*.

3.10.7.2 Font-Unit Values—Change in Properties

V5.3 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.

3.10.7.3 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.

3.10.7.4 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—can, 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.

3.10.7.5 Problems and Restrictions in the XUI Toolkit

The following DECwindows problems and restrictions exist:

- V5.3
- 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.

- 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 could 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 can still be selected by clicking the mouse button on the item text.
- Pop-up dialog boxes with no icon button (DwtNnoIconify set true) are initially created as icons (DwtNiconic true). Not only does the pop-up icon not have an icon box (and therefore cannot be popped up), but operations such as SetInputFocus on the pop-up icon 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);
```

3.10.7.6 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.

Programmer Release Notes

3.10 DECwindows Notes

3.10.7.7 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.

3.11 Disk Forced Error Counting Implemented

V5.5 Changes have been made in the VMS operating system's response to forced errors on disks. In previous releases of VMS, the forced error was only logged into the errorlog at the time that the error and corresponding bad block replacement originally took place. VMS now places an ERL\$LOGSTATUS entry into the errorlog and increments the error count for that device every time a QIO request for the file that contains the forced error is sent to the disk.

This behavior makes it easier to identify files that have forced errors in them. The first word of the first longword of the CDRP\$Q_IOSB field, at the bottom of the ERL\$LOGSTATUS entry, indicates whether a forced error was encountered. You can then run ANALYZE/DISK on the affected disk to identify the file with the forced error. Example 3-1 shows a sample ERL\$LOGSTATUS entry with the CDRP\$Q_IOSB field noted.

Example 3-1 Sample ERL\$LOGSTATUS Entry

```
***** ENTRY   1152. *****
ERROR SEQUENCE 3991.                   LOGGED ON:        SID 12000003
DATE/TIME 29-AUG-1991 08:58:23.38       SYS_TYPE 02200101
SYSTEM UPTIME: 0 DAYS 14:43:02
SCS NODE: URQUEL                        VAX/VMS T5.4-3

ERL$LOGSTATUS ENTRY KA65 CPU FW REV# 3. CONSOLE FW REV# 2.0
                                      XMI NODE # 1.

I/O SUB-SYSTEM, UNIT _HOST$DUA27:
      MSLG$L_CMD_REF   5E79002F
      ORB$L_OWNER      00000000           OWNER UIC [000,000]
      UCB$L_CHAR       1C4D4108
                                      DIRECTORY STRUCTURED
                                      FILE ORIENTED
                                      SHARABLE
                                      AVAILABLE
                                      MOUNTED
                                      ERROR LOGGING
                                      CAPABLE OF INPUT
                                      CAPABLE OF OUTPUT
                                      RANDOM ACCESS
      UCB$L_OPCNT      0001BDA0           114080. QIO'S THIS UNIT
      UCB$W_ERRCNT     000B
                                      11. ERRORS THIS UNIT
      UCB$W_STS        1810
                                      ONLINE
                                      SOFTWARE VALID
                                      UNLOAD AT DISMOUNT
      CDRP$L_MEDIA     0008C139
      CDRP$W_FUNC      000C
                                      READ PHYSICAL BLOCK
```

(continued on next page)

Programmer Release Notes

3.11 Disk Forced Error Counting Implemented

Example 3-1 (Cont.) Sample ERL\$LOGSTATUS Entry

CDRP\$L_BCNT	00002E00	TRANSFER SIZE 11776. BYTE(S)
CDRP\$W_BOFF	0170	368. BYTE PAGE OFFSET
CDRP\$L_PID	820619F6	REQUESTOR "PID"
CDRP\$Q_IOSB	08002144 00000039	(2144 Indicates a forced error) IOSB, 2048. BYTE(S) TRANSFERRED

The CDRP\$Q_IOSB longword provides the status of the transfer in the low word (2144 for forced error) and the number of bytes transferred in the high word.

Condition 2144 translates into:

```
%SYSTEM-F-FORCEDERROR, forced error flagged in last sector read
```

3.12 DSA Devices

The information in this section pertains to DSA Devices.

3.12.1 DSA Disk Driver—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 accessed through the Mass Storage Control Protocol (MSCP) server. This update reflected the most recent server that had become available.

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 about MSCP server load balancing, see the *VMS VAXcluster Manual*.

3.12.2 I/O Preferred Path Interface for DSA Devices—Problem Corrected

V5.4-1

The VMS Version 5.4 preferred path interface does not let you specify the local node as the preferred path. VMS Version 5.4-1 corrects this problem.

Starting with VMS Version 5.4-1, if you specify the local node as the preferred path, the disk class driver clears any preferred path information for the disk, allowing default path selection. For disks that have dual paths between VMS systems, specifying the local node as the preferred path causes the disk class driver to try to locate the disk on a local controller before trying Mass Storage Control Protocol (MSCP) server paths.

Programmer Release Notes

3.13 I/O Device Driver Notes

3.13 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.

3.13.1 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 about logical end-of-volume detection, see the Magnetic Tape Drivers chapter in the *VMS I/O User's Reference Manual: Part I*.

3.13.2 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 about lookups by file ID, see the ACP-QIO Interface chapter in the *VMS I/O User's Reference Manual: Part I*.

3.13.3 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).

3.14 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.

3.15 LAT Release Notes

The following sections include information for programmers working with the new LAT software included in this release of the VMS operating system. See also Section 2.22 for related information.

For a description of the new features associated with this LAT software, see the *VMS Version 5.5 New Features Manual* and the revised *VMS LAT Control Program (LATCP) Manual*.

The following changes have been made to the LAT QIO interface:

- LAT SETMODE and SENSEMODE QIO functions have been added.
- Forward connects and disconnects have been added.

- Some QIO functions have been superseded.
- Some Port Modifier QIO Function items no longer have meaning, but they are accepted by the driver and ignored.

The following sections describe these changes.

3.15.1 Change to Disconnect Processing

V5.5

In previous versions of the VMS operating system, an LTA device was unavailable for a second or two after a disconnect. With this version of the VMS operating system, the LTA device is available immediately.

You can now set the DISCONNECT QIO flush flag so data is flushed at disconnect time to prevent sessions from hanging in the “Disconnecting” state. The flush flag is the LAT\$M_FLUSH_DATA bit of the P2 argument of the DISCONNECT QIO.

3.15.2 FORTRAN Programs Using \$LATDEF

V5.5

Any FORTRAN source code that refers to \$LATDEF to obtain LAT error message codes needs to be modified to refer to \$LATMSGDEF. The new \$LATDEF module contains LAT programming definitions not including the LAT message codes. This only applies to FORTRAN source code modules. Any images already produced by the FORTRAN compiler before VMS Version 5.5 will run under Version 5.5.

3.15.3 New SETMODE and SENSEMODE QIO Functions

V5.5

The LAT SETMODE \$QIO function (IO\$_TTY_PORT!IO\$_M_LT_SETMODE) is used to create and delete LAT entities, such as nodes, services, ports, and links. It is also used to modify parameters of these entities.

The LAT SENSEMODE \$QIO function (IO\$_TTY_PORT!IO\$_M_LT_SENSEMODE) is used to obtain information about LAT entities, such as nodes, services, ports, and links.

For a complete description of these functions, see the *VMS Version 5.5 New Features Manual*.

3.15.4 Forward Port Connections and Disconnections

V5.5

Connections and disconnections on forward ports (outbound LAT) are now possible. You can get a forward port by assigning a channel to the LAT template device LTA0. You can map the port to a remote service (remote node and remote port are optional) by using the LAT SETMODE QIO function on a port and specifying the following items:

```
LAT$_ITM_TARGET_SERVICE_NAME  
LAT$_ITM_TARGET_NODE_NAME  
LAT$_ITM_TARGET_PORT_NAME
```

The IO\$_M_LT_CONNECT function modifier is supplied with the IO\$_TTY_PORT QIO function to perform the LAT CONNECT function.

The IO\$_M_LT_DISCON function modifier is supplied with the IO\$_TTY_PORT QIO function to perform the LAT DISCONNECT function.

3.15.5 LTA0 Now a Template Device

V5.5

LTA0 is now a template device. Assigning a channel to it returns a cloned LT device (or copy of it).

Programmer Release Notes

3.15 LAT Release Notes

3.15.6 Port Modifier QIO Functions Ignored

V5.5 Two Port Modifier (IO\$M_LT_MAP_PORT) QIO function items are obsolete. Although existing programs that use these function items will still run on this version of the VMS operating system, the functions are ignored by LTDRIVER. Therefore, Digital recommends that you modify existing programs accordingly, since these two function items might not be supported in future releases.

- IO\$V_LT_MAP_LNKNAM

You can no longer map an application port onto a specific link because of the changes to link processing.

- IO\$V_LT_MAP_NETADR

This function item was originally implemented as a temporary debug item code when host-initiated connects were first being prototyped. Now, the LAT protocol calls for the node address to be solicited, thus overwriting whatever is specified with this item code.

(Both function items are documented in Section 8.4.4 of the *VMS I/O User's Reference Manual: Part I*.)

3.15.7 QIO Completion Status

V5.5 The R0 QIO completion status now indicates only the success or failure of the execution of the QIO itself (VMS-specific checks, such as BYTLM). The first word of the IOSB contains the completion status of the LAT function.

3.15.8 QIO Functions Superseded

V5.5 Two QIO functions have been replaced. Although existing programs that use these functions are still supported in this release, Digital recommends that you modify those existing programs accordingly and begin using the new QIO functions whenever you create a new program.

The functions which have been superseded are as follows:

- Map Port Function

Instead of using the Map Port function code and modifier (FUNC=#IO\$_TTY_PORT!IO\$_M_LT_MAP_PORT) to associate a specific port on a terminal server with a local LTA device, you can now accomplish that task by using the LAT SETMODE QIO function on a port and specifying the following items:

```
LAT$_ITM_TARGET_NODE_NAME
LAT$_ITM_TARGET_PORT_NAME
LAT$_ITM_TARGET_SERVICE_NAME
```

- Set Rating Function

Instead of using the Set Rating function code and modifier, (FUNC=#IO\$_TTY_PORT!IO\$_M_LT_RATING) to set a static rating for a VMS service, you can now accomplish that task by using the LAT SETMODE QIO function on a service and specifying the item, LAT\$_ITM_RATING.

(Both functions are documented in Section 8.4.4 of the *VMS I/O User's Reference Manual: Part I*.)

3.16 Lock Manager

The release notes in this section pertain to the Lock Manager.

3.16.1 Dynamic Remastering

V5.5 The VMS distributed lock manager has been modified to dynamically redistribute the management of resource trees based on locking activity. The resource tree is moved to the most active node. This reduces latencies resulting from sending messages between nodes. Lock value blocks are not invalidated when the management of the tree is moved to a new node.

This behavior is influenced by the SYSGEN parameter LOCKDIRWT as follows:

- When several nodes with different values for LOCKDIRWT have locks on a resource tree, the tree is managed by the node with the largest value for LOCKDIRWT.
- When several nodes with identical values for LOCKDIRWT have locks on a resource tree, the tree is managed by the most active node.
- When only one node has locks on a resource tree, that node will manage the resource tree.

3.16.2 Extended Lock ID Table

V5.5 The VMS distributed Lock Manager has been modified to support up to 4 million Lock IDs.

3.17 Linker Utility

Release notes in this section pertain to the Linker Utility.

3.17.1 Analyzing Images Created with the VMS Version 5.5 Linker

V5.5 Because of changes to the VMS Version 5.5 image header, versions of the DCL command ANALYZE/IMAGE earlier than VMS Version 5.5 report an error when used to analyze an image created by VMS Version 5.5 Linker. The error, "Reserved flag bit 7 is set," appears at the end of the Fixed Header Information section of the analysis and can be ignored. The error is also included in the analysis summary line. Other errors reported by the analysis can indicate problems and should be examined.

3.17.2 Linker Now Combines Psects with the GBL, OVR, and NOSHR Attributes Set

V5.5 In the VMS Version 5.5 Linker, a problem was corrected that sometimes prevented the linker from combining two psects with the same name, one in a shareable image and the other in the main image, that had the GBL, OVR, and NOSHR attributes set. According to documented behavior, the linker should combine a psect in a shareable image with a psect in the main image if they have the same name and have the GBL and OVR attributes set. The SHR attribute should have no affect on this operation (it affects the sharing of memory in writable global sections).

The problem appeared with psects in shareable images that were specified in a shareable image library. In this case, the linker was incorrectly considering the NOSHR attribute when deciding which psects to combine. If the shareable image was specified in the link operation instead of in the shareable image library, the linker combined the psects correctly because it processed the shareable image before processing the object file. When specified in a shareable image library, the

Programmer Release Notes

3.17 Linker Utility

linker processes the shareable image after processing the object file, causing the problem.

In addition, the linker now reports the MULSHRPSC error if two or more shareable images with psects with the same name are referenced. The linker never allowed this type of operation; however, earlier versions simply ignored it. If you encounter this error, examine the psect declarations in your source files and link option files, especially the psect attributes. If the psects should be combined, then the definitions must be adjusted so that at most one shareable image contains the definition.

3.18 Mailbox Driver Write

V5.5 The mailbox driver buffer quota (Mailbox UCB BUFQUO) is now charged one byte for both zero length write requests and write end-of-file requests. Formerly, there was no charge to BUFQUO for such requests.

Any applications that depend on BUFQUO not being charged for zero length write requests and write end-of-file requests must be recoded.

3.19 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 might see the following error messages:

```
DDS-E-OPSYS, Operating system interface error  
LIB-E-BADBLOADR, bad block address
```

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 might 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.

Programmer Release Notes

3.19 Message Router Version 3.0 Installation

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.

3.20 Modem Signal Control Change

V5.4-3

For modem lines connected to DZ-type terminal interfaces (usually port TTA2 on workstations), modem signals are now checked for any changes before being sent. Previously, modem signals were checked at a rate determined by the SYSGEN parameter TTY_SCANDelta; the default value of this parameter was once every second. This rate of checking resulted in problems with high speed (9600 baud) modems, which could output up to 1000 characters before detecting that the CTS (clear to send) signal had been dropped. The terminal driver now checks if the modem signals have changed before each character is output. If the CTS signal has been dropped, the character is not sent.

3.21 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.

Programmer Release Notes

3.21 Processor Register Definition Symbols

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 might 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.

3.22 Programming Languages—Reinstallation Required

V5.2 To most easily use a VMS system routine from a given high level programming 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 want to use these routines.

3.23 Run-Time Library (RTL) Notes

The release notes in this section pertain to the Run-Time Library.

3.23.1 `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 3-2 have been added to the **flags** argument.

Table 3–2 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 about the **flags** argument, see the *VMS RTL Library (LIB\$) Manual*.

3.23.2 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.

3.23.3 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.

3.23.4 Specifying Input Format Mnemonics

V5.4-3 To generate a prompt for the user that differs from those listed in the table in the description of LIB\$INIT_DATE_TIME_CONTEXT, first call LIB\$INIT_DATE_TIME_CONTEXT to initialize the component LIB\$K_FORMAT_MNEMONICS. Then, call LIB\$GET_DATE_FORMAT and pass the context returned by LIB\$INIT_DATE_TIME_CONTEXT.

Using the component LIB\$K_FORMAT_MNEMONICS in the LIB\$INIT_DATE_TIME_CONTEXT routine does not affect the input string format. The only effect of LIB\$K_FORMAT_MNEMONICS is to change the string returned by LIB\$GET_DATE_FORMAT.

Programmer Release Notes

3.24 SET HOST/DTE/DIAL Command—DMF-32 Controller Problem

3.24 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 `SYSS$EXAMPLES:DT_DF03.MAR` to perform an `IO$_SENSEMODE!IO$_M_RD_MODEM $QIO` to check for a carrier signal. If set, the autodialer should assume success and continue.

3.25 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, BBBS, BBCC
- FFS, FFC
- EXTV, EXTZV
- CMPV, CMPZV
- INSV

3.26 VAX Ada Run-Time Library Notes

The release notes in this section pertain to the VAX Ada run-time library.

3.26.1 VAX Ada Version 2.2—Superseded

V5.4-3 The VMS bundled VAX Ada Run-Time Library included with VMS Version 5.4-3 supersedes the version of the VAX Ada Run-Time Library that is delivered as part of VAX Ada Version 2.2. To prevent the VMS Version 5.4-3 of the VAX Ada Run-Time Library from being overwritten by the older version shipped with VAX Ada Version 2.2, you must save the VAX Ada Run-Time Library images before installing VAX Ada Version 2.2, and then restore the VAX Ada Run-Time Library images after completing the VAX Ada Version 2.2 installation. For example:

```
#! Save VAX Ada Run-Time Library images.
#!
$ RENAME SYS$COMMON:[SYSLIB]ADARTL.EXE ADARTL.VMS
$ RENAME SYS$COMMON:[SYSMSG]ADAMSG.EXE ADAMSG.VMS
$
#! DO INSTALLATION OF VAX Ada V2.2.
$
#! Restore VAX Ada Run-Time Library images.
#!
$ RENAME SYS$COMMON:[SYSLIB]ADARTL.VMS ADARTL.EXE
$ RENAME SYS$COMMON:[SYSMSG]ADAMSG.VMS ADAMSG.EXE
$ LIBRARY/REPLACE/SHARE SYS$COMMON:[SYSLIB]IMAGELIB.OLB -
  SYS$COMMON:[SYSLIB]ADARTL.EXE
$ INSTALL REPLACE SYS$COMMON:[SYSLIB]ADARTL.EXE -
  /OPEN/SHARED/HEADER_RESIDENT
```

3.26.2 CALENDAR.SPLIT Improvement

V5.4 With VMS Version 5.4, the accuracy of the values returned by the procedure CALENDAR.SPLIT is improved.

3.26.3 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.

3.26.4 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.

3.26.5 '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.

Programmer Release Notes

3.26 VAX Ada Run-Time Library Notes

3.26.6 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.

3.26.7 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.)

3.27 VAX C Notes

The release notes in this section pertain to VAX C.

3.27.1 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.

3.27.2 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 these 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 file produced.

Note

Digital strongly recommends linking against the file VAXCRTL.EXE instead of VAXCRTLG.EXE. Libraries linked against the file VAXCRTLG.EXE can not be usable by programs that use D_FLOAT variables and library routines.

3.27.3 Run-Time Library Correction

V5.4-3 In Version 5.4 of the VMS operating system, a problem with the difftime() routine was introduced. The difftime() routine returned incorrect results if the second time value was larger than the first time value.

This problem has been corrected in VMS Version 5.4-3.

3.27.4 Sample Programs

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 will not work. To correct this problem, reinstall VAX C and extract the .h files.

3.28 VAX MACRO Notes

The release notes in this section pertain to VAX MACRO.

3.28.1 VAX MACRO Corrections

V5.4-3 These sections describe corrections to VAX MACRO in VMS Version 5.4-3. The following problems have been fixed.

3.28.1.1 Debug Symbol Table (DST) Corruption

The Debug Symbol Table (DST) was sometimes corrupted. For VMS Version 5.4-3, this problem has been corrected.

The problem was found to be an invalid Debug Symbol Table (DST) record. The fault was with the logic in the assembler code responsible for creating a DST for an ASCII directive. DSTs are created only when the /DEBUG qualifier is specified during assembly.

This problem affected users of the VMS Symbolic Debugger, the VMS Patch Utility, and the Performance Coverage Analyzer (PCA).

Programmer Release Notes

3.28 VAX MACRO Notes

Patch Utility Symptoms

The message "Premature End-of-DST Encountered" was displayed when attempting to PATCH an executable image created from MACRO source code. The following procedure shows this behavior:

```
$ CREATE test.mar
TIM:      .ASCII 'R EVTEST2'
          .ENTRY DECGO, ^M<>
          RET
          .END DECGO
^Z
$ MACRO/DEBUG test.mar
$ LINK/DEBUG test.mar
$ PATCH test.exe

    PATCH Version 4-00    15-Sep-1984
Premature End-of-DST Encountered.
PATCH>
```

Symbolic Debugger Symptoms

The instructions in the following MACRO program produced access violations when executed under the control of the debugger:

```
$ CREATE test.mar
          .BLKB 130560 ;
A:        .ASCII 'a' ; Omit this line and no error occurs
          ; Change to .ASCID and no error occurs
          .ENTRY test,0
          .END test
^Z
$ MACRO/DEBUG test.mar
$ LINK/DEBUG test.obj
$ RUN test
    VAX DEBUG Version V5.3-17
%SYSTEM-F-ACCVIO, access violation, reason mask=00, virtual
                    address=00179800, PC=0004C698, PSL=03C00000
```

The error message or program counter (PC) and virtual address varied depending on which version of VMS was running.

Other programs containing the .ASCII directive produced different error messages when they were compiled and linked with the /DEBUG qualifier.

PCA Symptoms

When a user application includes a MACRO module, the Performance Coverage Analyzer (PCA) loads a subset of the DSTs into the PCA datafile. If the application encountered an invalid DST record in PCA Versions 2.1 and 2.2, the problem appeared as an internal error, as indicated by the following messages:

```
%PCA-F-INTERR, internal PCA error in PCASYMTAB\PCA$RST_BUILD 09
%PCA-I-NODATACOL, exiting with no data collection phase
```

In PCA Version 3.0, the internal error PCASYMTAB\PCA\$RST_BUILD 09 now produces a new message, as follows:

```
Error INCDST - incorrect DST nesting in module xxx. Datafile not set.
```

If PCA encounters an error when loading the DSTs, PCA will not set the datafile.

3.28.1.2 Debug Symbol Table (DST) Corruption Caused by a Repeat Loop in a Macro

Commonly called the source-line correlation problem, this problem affected the debugging of MACRO programs that invoked either the \$FAO or \$FAOS system service macro or an application-defined macro that contained any of the following directives: .IRP, .IRPC, .REPT, or .REPEAT.

In such cases, the DST information about source lines and line numbers could be unreliable. During a debugging session, the problem occurred as soon as the macro was invoked and persisted as long as execution was within the module in which the macro was invoked.

The screen-mode source display (SRC) was not updated correctly when the debugger suspended execution. The pointer in the source display was typically 1 or 2 lines off the line at which execution was actually suspended.

The problem showed up also when the debugger issued a "stepped to...", "break at...", or "trace at..." message, which might indicate the wrong source line.

Specifying a line number in a command could also give incorrect results. The difference in behavior before and after the fix is shown in three examples:

- Example 3-2 shows a debug session before the fix was applied.
- Example 3-3 shows a debug session after the fix was applied.
- Example 3-4 shows the sample code used to generate these sessions.

Example 3-2 Sample Debug Session Before the Fix Is Applied

```
$ run test
      VAX DEBUG Version V5.4-019
%DEBUG-I-INITIAL, language is MACRO, module set to .MAIN.

DBG> examine @pc
.MAIN.\START\FAO_LOOP: SOBGEQ  B^.MAIN.\LOOP_COUNT,.MAIN.\START\DOIT_AGAIN
DBG> Step
stepped to .MAIN.\START\DOIT_AGAIN
      21:      blbs r0,fao_ok
DBG> examine @pc
.MAIN.\START\DOIT_AGAIN:      PUSHL  B^.MAIN.\RECEIVE_COUNT
DBG> Step
stepped to .MAIN.\START\%LINE 22
      22:      pushl r0
DBG> examine @pc
.MAIN.\START\%LINE 22:  BLBS      R0,.MAIN.\START\FAO_OK
DBG> Step
stepped to .MAIN.\START\FAO OK
      26:      calls #1,g^lib$put_output
DBG> examine @p
.MAIN.\START\FAO_OK:      PUSHAB  W^.MAIN.\FAODESC
DBG> Step
stepped to .MAIN.\START\%LINE 27
      27:      incl receive_count
```

Programmer Release Notes

3.28 VAX MACRO Notes

Example 3-3 Sample Debug Session After Fix Is Applied

```
$ run test
      VAX DEBUG Version V5.4-019
%DEBUG-I-INITIAL, language is MACRO, module set to .MAIN.

DBG> examine @pc
.MAIN.\START\FAO_LOOP: SOBGEQ  B^.MAIN.\LOOP_COUNT,.MAIN.\START\DOIT_AGAIN
DBG> Step
stepped to .MAIN.\START\DOIT_AGAIN
      19:          p1 = receive_count
DBG> examine @pc
.MAIN.\START\DOIT_AGAIN:      PUSHL  B^.MAIN.\RECEIVE_COUNT
DBG> Step
stepped to .MAIN.\START\%LINE 21
      21:          blbs r0,fao_ok
DBG> examine @pc
.MAIN.\START\%LINE 21:  BLBS    R0,.MAIN.\START\FAO_OK
DBG> Step
stepped to .MAIN.\START\FAO_OK
      25:          pushab faodesc
DBG> examine @pc
.MAIN.\START\FAO_OK:      PUSHAB  W^.MAIN.\FAODESC
DBG> Step
stepped to .MAIN.\START\%LINE 26
      26:          calls #1,g^lib$put_output
```

Example 3-4 Source Code to Generate Sample Sessions

```
.library "sys$library:lib.mlb"

faodesc: .long 80
         .address faobuf
faobuf:  .blkb 80
faolen:  .blkl 1
rx_countstr: .ascid "RECEIVE FRAME !UL"
receive_count: .blkl
loop_count: .word 10

         .entry start,^m<>

fao_loop:
    sobgeq loop_count,doit_again
    brw end
doit_again:
    $fao_s ctrstr = rx_countstr, -
           outbuf = faodesc,-
           outlen = faolen,-
           p1 = receive_count

    blbs r0,fao_ok
    pushl r0
    calls #1,g^lib$stop
fao_ok:
    pushab faodesc
    calls #1,g^lib$put_output
    incl receive_count
    brw fao_loop
end:
    $exit_s
    .end start
```

3.28.2 Known Problems in VAX MACRO

V5.4-3

The following is a list of all known problems in VAX MACRO:

1. When concatenated source files are assembled using the /ANALYSIS_DATA command qualifier, incorrect source record numbers are reported to VAX SCA for other than the first concatenated file. These incorrect source record numbers are exactly one fewer than the correct numbers.
2. 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 any error other than the first error.
3. 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 can also occur in the generated DST, causing VAX DEBUG to display incorrect source records. To work around this problem for debugging purposes, manually concatenate the source files (forming a single input file) before assembly.
4. Assembly errors can occur if the string ".REPT" is found in comments included within a .REPT loop.
5. The %EXTRACT macro string operator is parsed by the assembler even if it is found in an unsatisfied conditional block or in a comment.
6. If the total length of an 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.
7. No diagnostic message is given if a keyword actual argument is associated with a created local label. To work around 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*.)
8. For branch instructions located in the body of a macro, the assembler does not always generate a diagnostic for a branch out of range.
9. The assembler can abort with "%SYSTEM-F-RADRMOD, reserved addressing fault..." if the register mask operator used with the .ENTRY directive is mistyped as "M^" or "M" instead of "^M".
10. When evaluating a large expression, the assembler may incorrectly generate a "store word" instruction instead of the correct "store longword" instruction. This causes the linker to generate a truncation error.
11. 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.
12. The %LENGTH operator does not work within .REPEAT blocks.
13. 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.

Programmer Release Notes

3.28 VAX MACRO Notes

14. When `.DISABLE` is specified with multiple items, some of the items might be ignored. Use individual `.DISABLE` directives for all the items to avoid this problem.
15. The assembler does not correctly evaluate expressions containing the arithmetic shift operator. For example, the expression:

```
<1@30+1@31-2>
```

is evaluated as follows:

```
<<<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, force the order of evaluation to be correct by placing angle brackets around subexpressions containing the arithmetic shift operator.
16. A "branch to subroutine" instruction immediately followed by a directive that computes an expression containing the ASCII operator (^A) can yield a "MACRO-W-DATATRUNC, Data truncation error" diagnostic message. To avoid this problem, place a NOP (or any other) instruction immediately after the "branch to subroutine" instruction.
17. The assembler might 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.
18. If the last character in a `.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).
19. Attempting to assemble a VAX MACRO program using a command line in excess of 512 characters results in a corrupt object file.
20. The assembler cannot display listing line numbers greater than 64K. To avoid this problem, do not assemble modules containing more than 64K source records. (Pay particular attention to the total size of a module when concatenating source files.)

The problems listed in this section will be corrected in a future release.

3.29 VAX Math Run-Time Library (MTHRTL)—Improved Accuracy

V5.4-3

The accuracy for `D_float` and `G_float` `ASIN` functions in MTHRTL has been improved for input arguments having values slightly less than 1.0. You might notice a change in the results of calculations from these math routines, and the new answers should be considered correct.

3.30 VMS Convert/Reclaim Utility—Problem

V5.4-3

A problem exists with the VMS Convert/Reclaim Utility (`CONVERT/RECLAIM`). During reclamation of a VMS RMS prolog 3 Indexed file containing multiple keys, `CONVERT/RECLAIM` incorrectly handles records that have been marked for deletion yet still have an associated alternate key. Such records can be created as the result of a `$DELETE` operation by an application program using the Fast Delete Record Processing Option (bit `RAB$V_FDL` in field `RAB$L_ROP` set). Such records can also exist inadvertently as the result of a failed record

Programmer Release Notes

3.30 VMS Convert/Reclaim Utility—Problem

operation, for example, due to a fatal hardware failure, lack of software resources, or process deletions (\$DELPRC, \$STOP/ID).

Failure of CONVERT/RECLAIM to handle such records correctly can cause future, normal RMS record-processing operations for a file, which had such records, to fail unexpectedly. Possible errors returned are RMS\$_RNF and RMS\$_DUP.

You can avoid the problem by taking one of the following actions:

1. Use the VMS Convert Utility instead of the VMS Convert/Reclaim Utility.
2. Avoid using the Fast Delete Record Processing Option if CONVERT/RECLAIM is expected to be used.
3. Prior to using the VMS Convert/Reclaim Utility, open the file for write and access every record along each secondary key that allows duplicates.

Because action 3 is generally very resource intensive, Digital suggests using the VMS Convert Utility (action 1) as the preferred action. It has the advantage of providing internal file reorganization in addition to reclaiming data buckets. However, its usage might be prohibited by operational requirements such as a lack of free disk space.

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

3.31 VMS Record Management Services (RMS) Notes

The release notes in this section pertain to the VMS Record Management Services (RMS).

3.31.1 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 about the XAB\$_NORECORD XABITM, see the *VMS Version 5.5 New Features Manual*.

Programmer Release Notes

3.31 VMS Record Management Services (RMS) Notes

3.31.2 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.

3.31.3 RMS Corrections

V5.4-3 This section lists RMS problems that have been corrected in VMS Version 5.4-3.

- Prior to this release, a potential race condition could develop when a process opened a sequential data file for shared-write access to append data to the file, but then closed it just as another process tried to access it. The race condition infrequently would result in records being overwritten (effectively deleting the records). This release has corrected this condition.

If you are running applications under previous versions of VMS, you can avoid this problem by having one of the accessing processes keep the file open as long as data is being appended to the file.

- Prior to Version 5.4-3, if you followed a \$OPEN call with a \$DISPLAY call, the protection XAB (XABPRO) ACL fields were null instead of containing appropriate protection data. This behavior resulted when the XQP erroneously returned the value -1 in the context pointer. To correct the problem, VMS now clears the context pointer in this situation.
- Prior to Version 5.4-3, the routine to lock the file while obtaining end-of-file information during XAB processing of sequential files did not preserve the AP register over a stall.

Note

Stalling is an elaborate internal RMS version of WAITING. If a user program enters a wait state and RMS has to wait for an IO to finish or a lock to become available, RMS will stall on behalf of the user program. Stalling is most commonly used in the context of threads.

When RMS resumed operations, the corrupted AP register caused an access violation. To correct the problem, VMS now saves the AP register before calling the routine to lock the file.

- Prior to Version 5.4-3, Convert/Reclaim operations on index files using key compression could terminate with an access violation when the key following the deleted key was expanded such that it required more space than was formerly required for itself and the deleted key. This usually occurs when key sizes are relatively large with respect to data size and where significant trail compression has occurred.

Note

Trail compression refers to the second part of key and index compression in indexed file keys where RMS stores the number of identical trailing characters in a key as a space-saving measure. It is internal to RMS and the user has no control over it other than enabling or disabling the key and/or index compression.

To correct the problem, when the reconstructed key is larger than the space reclaimed through compression, the reclamation process is terminated and Convert/Reclaim looks for the next empty data bucket to continue its operations.

3.31.4 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 systems that are not tuned properly. 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.

3.31.5 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.

Programmer Release Notes

3.31 VMS Record Management Services (RMS) Notes

3.31.6 RMS Journaling Notes

The release notes in this section pertain to VAX RMS journaling.

3.31.6.1 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 VMS Convert Utility. You can then transfer the converted copy of the file to the system running VMS Version 4.7 or earlier. For more information refer to the *VMS Convert and Convert/Reclaim Utility Manual*.

3.31.6.2 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)
```

3.31.6.3 Synchronous Recovery

V5.4-3 Section 4.11.2.1 of the *VAX RMS Journaling Manual* describes the synchronous and asynchronous behavior of Detached Recovery. Asynchronous recovery is the default. The section states: “In asynchronous recovery, the detached recovery process acquires the record locks that were held within orphaned transactions. Since this makes the file transaction consistent, other processes can have immediate access to nonlocked records.”

This means that, in the case of a \$OPEN, the open can complete before Detached Recovery of orphaned transactions has completed. This, in turn, means that some records might still be locked by Detached Recovery when the application is resumed following the \$OPEN completion. If the application tries to access these locked records before Detached Recovery releases them, then a record locked (RMS\$_RLK) status will be returned as if the transaction to be recovered were still active.

Experience has shown that some applications will fail as a result of this new feature; for example, an application that relies on a synchronization method other than an RMS record lock will fail. To return to behavior before VMS Version 5.4, a logical name (RMSREC\$LOG) has been provided that, when it is defined, forces Detached Recovery to recover all files synchronously. Defining this logical name will ensure that there are no more locked records when a \$OPEN completes.

Programmer Release Notes

3.31 VMS Record Management Services (RMS) Notes

RMSREC\$LOG must be defined as an EXECUTIVE_MODE logical in the system logical name table LNM\$SYSTEM with value 401. For example:

```
$ DEFINE/SYSTEM/EXECUTIVE_MODE RMSREC$LOG 401
```

However, under some circumstances, using this feature can cause the recovery server to hang. A correction for this problem can be obtained through your Digital Services Group.

3.31.6.4 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.

3.32 VMS System Services Notes

The notes in this section pertain to VMS system services.

3.32.1 Error Messages Referring to Shared Memory

V5.4 When using 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 issue these error messages:

```
$ASCEFC  
$CREMBX  
$CRMPSC  
$DGBLSC  
$UPDSEC
```

3.32.2 Mailbox Driver Error Checking

V5.5 The mailbox driver now checks to see if any illegal function modifier bits are set in a QIO request issued to the driver. If any illegal bits are set, an SS\$_ILLIOFUNC error is returned. In previous releases of VMS, the mailbox driver ignored these bits.

For example, if a \$QIO IO\$_WRITEVBLK!IO\$_M_NOWAIT request was erroneously issued to the mailbox driver in a previous release of VMS, the mailbox driver would have ignored the erroneous function modifier and treated it as if \$QIO IO\$_WRITEVBLK had been issued. In this release of VMS, the mailbox driver returns the error SS\$_ILLIOFUNC.

Programmer Release Notes

3.32 VMS System Services Notes

As a result of this new error checking, any \$QIO requests that contained function modifiers not recognized by the mailbox driver will generate errors. This programming problem will need to be fixed.

3.32.3 \$GETQUI—QUI\$_QUEUE_STATUS Longword

V5.5 In VMS Version 5.4, a problem existed with QUI\$_QUEUE_STATUS longword for the SYS\$GETQUI system service. The stopped bit could be set even if the queue was not stopped. Programmers could have worked around this behavior by checking both the stopped and idle bits to make sure a queue was stopped.

VMS Version 5.5 fixes this problem. A subset of the status bits, including the stopped and idle bits, has been designated as state bits. A queue can now be in only one state at any time. To determine if a queue is stopped, a program need only check the QUI\$_M_QUEUE_STOPPED bit in the QUI\$_QUEUE_STATUS longword.

The following bits, available in Version 5.4, are now state bits:

- QUI\$_V_QUEUE_IDLE
- QUI\$_V_QUEUE_PAUSED
- QUI\$_V_QUEUE_PAUSING
- QUI\$_V_QUEUE_RESUMING
- QUI\$_V_QUEUE_STALLED
- QUI\$_V_QUEUE_STARTING
- QUI\$_V_QUEUE_STOPPED
- QUI\$_V_QUEUE_STOPPING

Several other changes were made to the QUI\$_QUEUE_STATUS longword in Version 5.5. For a complete description of the QUI\$_QUEUE_STATUS item code, see the *VMS System Services Reference Manual*.

3.32.4 \$GETSYI System Service: VAX Computer Model Numbers Added to Item Codes

V5.5 The following sections list VAX computer models defined for the SYI\$_CPU, SYI\$_XCPU, and SYI\$_HW_NAME item codes.

3.32.4.1 Model Numbers for the SYI\$_CPU Item Code

When you specify SYI\$_CPU, the \$GETSYI system service returns the CPU processor type for the local node. The \$PRDEF macro defines the following symbols to support new processor types:

Processor	Symbol
VAX 4000-200	PR\$_SID_TYP660
VAX 4000-500	PR\$_SID_TYP690
VAX 6000-500 series	PR\$_SID_TYP1202
VAX 6000-600 series	PR\$_SID_TYP1302
VAXstation 4000-60	PR\$_SID_TYP46
VAXstation 4000 VLC	PR\$_SID_TYP440

Programmer Release Notes 3.32 VMS System Services Notes

Processor	Symbol
MicroVAX 3100-30, 3100-40	PR\$_SID_TYP440
MicroVAX 3100-80	PR\$_SID_TYP46
VAXft Model 410	PR\$_SID_TYP550

3.32.4.2 Model Numbers for the SYI\$_XCPU Item Code

When you specify SYI\$_XCPU, the \$GETSYI system service returns the extended CPU processor type for the local node. (You should obtain the general processor type value first by using the SYI\$_CPU item code.) The \$PRDEF macro defines the following symbols to support new processor types:

VAX Processor Type Symbol	Extended Processor Type	Extended Processor Symbol
PR\$_SID_TYP46	MicroVAX 3100-80	PR\$_XSID_V12_46
PR\$_SID_TYP46	VAXstation 4000-60	PR\$_XSID_V12_46
PR\$_SID_TYP440	VAXstation 4000 VLC	PR\$_XSID_V14_440
PR\$_SID_TYP440	MicroVAX 3100-30, 3100-40	PR\$_XSID_V14_440
PR\$_SID_TYP550	VAXft Model 410	PR\$_XSID_V14_550
PR\$_SID_TYP660	VAX 4000-200	PR\$_XSID_V14_660
PR\$_SID_TYP690	VAX 4000-500	PR\$_XSID_V13_690
PR\$_SID_TYP1202	VAX 6000-500 series	PR\$_XSID_V12_1202
PR\$_SID_TYP1302	VAX 6000-600 series	PR\$_XSID_V13_1302

3.32.4.3 Model Numbers for the SYI\$_HW_NAME Item Code

When you specify SYI\$_HW_NAME, the \$GETSYI system service returns the VAX model name string of the node. The VAX model name is a character string that describes the model of the VAX node (such as VAX 8800, MicroVAX II). The VAX model name usually corresponds to the nameplate that appears on the outside of the CPU cabinet. The following table lists new VAX model processor names and the corresponding model types:

VAX Model Processor Name	VAX Model Type
MicroVAX 3100-30, 3100-40 (one user)	VAX\$K_VPV2_S
MicroVAX 3100-30, 3100-40 timeshare	VAX\$K_VPV2_T
MicroVAX 3100-80 (one user)	VAX\$K_VKA46_S
MicroVAX 3100-80 timeshare	VAX\$K_VKA46_T
VAX 4000-200 timeshare	VAX\$K_V660
VAX 4000-200 server	VAX\$K_V690_S
VAX 4000-500 timeshare	VAX\$K_V690
VAX 4000-500 server	VAX\$K_V690_S
VAX 6000-510 timeshare	VAX\$K_1202_1T
VAX 6000-520 timeshare	VAX\$K_1202_2T
VAX 6000-530 timeshare	VAX\$K_1202_3T

Programmer Release Notes

3.32 VMS System Services Notes

VAX Model Processor Name	VAX Model Type
VAX 6000-540 timeshare	VAX\$K_1202_4T
VAX 6000-550 timeshare	VAX\$K_1202_5T
VAX 6000-560 timeshare	VAX\$K_1202_6T
VAX 6000-510 server	VAX\$K_1202_1S
VAX 6000-520 server	VAX\$K_1202_2S
VAX 6000-610 timeshare	VAX\$K_1302_1T
VAX 6000-620 timeshare	VAX\$K_1302_2T
VAX 6000-630 timeshare	VAX\$K_1302_3T
VAX 6000-640 timeshare	VAX\$K_1302_4T
VAX 6000-650 timeshare	VAX\$K_1302_5T
VAX 6000-660 timeshare	VAX\$K_1302_6T
VAX 6000-610 server	VAX\$K_1302_1S
VAX 6000-620 server	VAX\$K_1302_2S
VAX 6000-630 server	VAX\$K_1302_3S
VAXft Model 410	VAX\$K_V550FT
VAXstation 4000 VLC timeshare	VAX\$K_VKA48W_T
VAXstation 4000 VLC server	VAX\$K_VKA48W_S
VAXstation 4000-60 monochrome, timeshare	VAX\$K_VKA46M_T
VAXstation 4000-60 monochrome, (one user)	VAX\$K_VKA46M_S
VAXstation 4000-60 color, timeshare	VAX\$K_VKA46C_T
VAXstation 4000-60 color, (one user)	VAX\$K_VKA46C_S
VAXstation 4000-60 ScanProc, timeshare	VAX\$K_VKA46S_T
VAXstation 4000-60 ScanProc, (one user)	VAX\$K_VKA46S_S

3.32.5 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 might be returned or unexpected errors might 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. 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.

3.32.6 SJC\$_START_QUEUE_MANAGER Function Code for \$SNDJBC Service

V5.5 The behavior of the SJC\$_START_QUEUE_MANAGER function code for \$SNDJBC has changed. For more information, see Section 2.7.8.1.

3.32.7 \$SNDJBC Service—Obsolete Item Codes

V5.5 The following item codes for the \$SNDJBC service are obsolete:

SJC\$_BUFFER_COUNT
SJC\$_EXTEND_QUANTITY
SJC\$_QUEUE_FILE_SPECIFICATION
SJC\$_QUEMAN_RESTART
SJC\$_NO_QUEMAN_RESTART

3.33 XDELTA Notes

The release notes in this section describe procedures for booting the XDELTA debugger and requesting an interrupt for XDELTA for all computers available with VMS Version 5.5. The DELTA/XDELTA debuggers are used to monitor the execution of user programs and the VMS operating system. For more information about the XDELTA debugger, see *VMS Delta/XDelta Utility Manual*.

3.33.1 Invoking XDELTA

V5.4 To invoke the XDELTA debugger, perform the following steps:

1. Boot the system using a console command or a command procedure that includes XDELTA.
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 booting 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 booting the computers, requesting an interrupt, and setting breakpoints in program code.

Programmer Release Notes

3.33 XDELTA Notes

Some boot 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 boot program. Table 3-3 contains a description of valid values for the flag.

Table 3-3 Valid XDELTA Values for the BOOT Command Qualifier

Value	Description
0	Normal, nonstop boot (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.

3.33.1.1 Booting VMS Computer Systems Post Version 5.4

V5.5 Table 3-4 provides references to appropriate sections for booting XDELTA on VMS computer systems post Version 5.4.

Table 3-4 Booting XDELTA on VMS Computer Systems Post Version 5.4

Computer System	Section Reference
VAXstation 4000 Series Computer	Section 3.33.1.11
VAXft-410	Section 3.33.1.10
VAXft-610	Section 3.33.1.10
VAXft-612	Section 3.33.1.10
VAXft-110	Section 3.33.1.10

3.33.1.2 Booting a VAX-11/730 or VAX-11/750 Computer Using the VMS Console TU58

V5.4 In addition to the normal system boot command files, the VMS console TU58 for a VAX-11/730 computer contains the following command files that boot the system with XDELTA:

- DQAXDT
- DQ0XDT
- DL0XDT
- DUAXDT

- DU0XDT

To boot 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 boot 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 booting process, enter CONTINUE.

To boot 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, DMAXDT, and DBAXDT, which contain the command procedures that boot the system from DU, DM, and DB devices, respectively.

3.33.1.3 Booting XDELTA on a MicroVAX 2000, VAXstation 2000, MicroVAX 3400 Series, VAXstation 3520, VAXstation 3540, MicroVAX/VAXstation 3600 Series, MicroVAX 3900, VAX 4000 Series, MicroVAX II, or VAX-11/750 Computer

V5.4 To boot 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 boot 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 3-3.

For example, the following command boots 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 booting operation, enter CONTINUE.

Programmer Release Notes

3.33 XDELTA Notes

See the upgrade and installation supplement for your computer for more information about the B command.

3.33.1.4 Booting XDELTA on a VAX-11/780 or a VAX-11/785 Computer

V5.4 In addition to the normal system boot command files, the VMS console RX01 for a VAX-11/780 or VAX-11/785 computer contains the following command files that boot the system with XDELTA:

- DUAXDT.CMD
- DMAXDT.CMD
- DBAXDT.CMD

To boot 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 booting operation, enter CONTINUE.

3.33.1.5 Booting XDELTA on a VAX 6000-Series Computer

V5.4 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. Press 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
```

Table 3-5 lists the qualifiers to the BOOT command.

Table 3-5 XDELTA BOOT Command Qualifiers for VAX 6000-Series Computers

Qualifier	Function
/R5:a	Deposits a value (<i>a</i>) into R5. For valid values for the flag, refer to Table 3-3.
/XMI:b	Specifies the XMI node number (<i>b</i>) of the node being accessed. Defaults to the lowest numbered I/O device.

(continued on next page)

Table 3–5 (Cont.) XDELTA BOOT Command Qualifiers for VAX 6000-Series Computers

Qualifier	Function
/BI:c	Specifies the BI node number (<i>c</i>) 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>
```

3.33.1.6 Booting XDELTA on a VAX 8200, 8250, 8300, or 8350 Computer

V5.4

To boot 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 boot program. Refer to Table 3–3 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
```

3.33.1.7 Booting XDELTA on a VAX 8530, 8550, 8810 (8700), 8820, 8820-N (8800), 8830, or 8840 Computer

V5.4

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 3–3 for a description of the valid values for this flag.

Programmer Release Notes

3.33 XDELTA Notes

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
```

3.33.1.8 Booting XDELTA on a VAX 8600 or 8650 Computer

V5.4

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*OXDT format, where *dd* is the device code of the system disk.

If DUOXDT is present, follow the standard boot procedure except in the following two steps:

- When you specify the boot 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 DUOXDT:

```
>>> @DUOXDT
```

The command procedure boots the processor and prompts you from SYSBOOT. When the SYSBOOT> prompt appears, enter any SYSBOOT command.

To continue the booting operation, enter CONTINUE.

If the console media does not have the DUOXDT file, perform a normal boot procedure using an available *ddu*GEN.COM, *ddu*BOO.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 3-3 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
```

3.33.1.9 Booting 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.COMD) and the unit number 30. Use the parameter CI30.

Table 3-6 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.COMD 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
```

Table 3-6 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 3-3 for a description of the valid values for the flag.
/R3:shadow	Specifies shadowing information. Required only 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 about /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.

(continued on next page)

Programmer Release Notes

3.33 XDELTA Notes

Table 3–6 (Cont.) BOOT Command Qualifiers for VAX 9000 Computers

Qualifier	Function
/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 <i>xy</i> , where <i>x</i> is the XMI number and <i>y</i> 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
```

3.33.1.10 Booting XDELTA on a VAXft 3000 Computer

V5.4 To boot a VAXft 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, release the LOCAL CONSOLE DISABLE switch on the front of the system cabinet that your console terminal is connected to. Note that the VAXft computer has two system cabinets and two system consoles. Make sure you deactivate the appropriate LOCAL CONSOLE DISABLE switch.

3.33.1.11 Booting 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 devname
```

For *devname*, enter the device name of the system disk. For *f*, enter a value from Table 3–3.

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
```

3.33.2 Requesting an Interrupt

V5.4 If you set the boot control flag in R5 to 7, as described in Section 3.33.1, XDELTA stops at an initial breakpoint during the system boot 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.

Programmer Release Notes 3.33 XDELTA Notes

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 E_{16} into IPR 14_{16} .

For a VAX 8530, 8550, 8600, 8650, 8810 (8700), 8820, 8820-N (8800), 8830, 8840, 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
```

For a VAX 9000 computer, enter the following commands at the console terminal to request the interrupt:

```
$ Ctrl/P
>>> HALT/CPU=ALL
>>> D/I 14 E
>>> C/CPU=ALL
```

On a VAX 6000-series, VAX 8200, VAX 8250, VAX 8300, VAX 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.
- Enter the following commands at the console terminal:

```
>>> D/I 14 E
>>> C/ALL
```

For a VAXft 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 Series, 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.

Programmer Release Notes

3.33 XDELTA Notes

- Enter the following commands at the console terminal:
 >>> D/I 14 E
 >>> C

Documentation Release Notes

This chapter contains additions and corrections to the VMS documentation set.

4.1 VMS Version 5.5 Specific Release Notes Pertaining to VMS Documentation

These release notes are cumulative from VMS Version 5.0. The following sections contain documentation release notes that pertain specifically to VMS Version 5.5:

- Section 4.2—*Using VMS BACKUP*
- Section 4.3—*Building Dependable Systems: The VMS Approach*
- Section 4.4—*Guide to Maintaining a VMS System*
- Section 4.8.1—COPY/TRUNCATE Command
- Section 4.8.3—DELETE/QUEUE Command
- Section 4.8.5—F\$GETDVI Lexical Function
- Section 4.8.6—F\$GETQUI Lexical Function
- Section 4.8.7—INITIALIZE/MEDIA_FORMAT=[NO]COMPACTION Command
- Section 4.8.8—PRINT/PRIORITY Command
- Section 4.8.9—REPLY Command
- Section 4.8.10—RUN (Process) Command
- Section 4.8.11—SET CLUSTER/EXPECTED_VOTES Command
- Section 4.8.12—SET DEVICE/SERVED Command
- Section 4.8.13—SET ENTRY/AFTER Command
- Section 4.8.14—SET ENTRY/PRIORITY Command
- Section 4.8.15—SET ACL/OBJECT_TYPE Command
- Section 4.8.16—SHOW SYSTEM/INTERACTIVE Command
- Section 4.8.17—SUBMIT/PRIORITY Command
- Section 4.23—Corrections to the *VMS System Services Reference Manual*
- Section 4.27—Corrections to the *VMS Version 5.4-3 Release Notes*

Documentation Release Notes

4.2 Using VMS BACKUP

4.2 Using VMS BACKUP

V5.5 A new manual, *Using VMS BACKUP*, is available to help users complete common tasks with the VMS Backup Utility (BACKUP). Intended as a companion to the *VMS Backup Utility Manual*, *Using VMS BACKUP* includes information about disk and tape operations; backing up and restoring files, directories, and disks; troubleshooting; and creating your own BACKUP command procedures.

Using VMS BACKUP is available on your VMS system disk (SYS\$EXAMPLES:USING_VMS_BACKUP.*) in DECW\$BOOK, LINE, and PS format.

4.3 Building Dependable Systems: The VMS Approach

V5.5 The VMS Version 5.5 documentation set includes a new handbook, *Building Dependable Systems: The VMS Approach*. This handbook addresses the building blocks of dependable systems and explains basic dependability principles. It also provides practical techniques for utilizing the dependability features of VAX systems with those of the VMS operating system and layered software products to help you form a dependable computing system.

Building Dependable Systems: The VMS Approach is included with the VMS Version 5.5 Base Documentation Set, and it can be ordered separately by using order number AA-PH62A-TE.

4.4 Guide to Maintaining a VMS System

V5.5 In Section 6.2 of the *Guide to Maintaining a VMS System*, the classes of security events audited by default should be extended to include the AUTHORIZATION class. Thus, all changes to the system authorization file (SYSUAF), the network proxy authorization file (NETPROXY), and the rights database (RIGHTSLIST) are audited.

4.5 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 C of this manual, which contains updated information.

4.6 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.

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 5.5 New Features Manual*.

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 5.5 New Features Manual*.

4.7 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.

4.8 VMS DCL Dictionary

The release notes in this section pertain to the *VMS DCL Dictionary*.

4.8.1 COPY/TRUNCATE Command

- V5.5 The /TRUNCATE qualifier is useful only with sequential files. In previous versions of the *VMS DCL Dictionary*, this restriction was not stated.

4.8.2 DEFINE/FORM Command

- V5.4-3 The documentation for *form-number* states that the parameter assigns a number in the range 0 through 2,147,483,647 to the form being defined.
- The correct range is 0 through 9999.

4.8.3 DELETE/QUEUE Command

- V5.5 The description of the DELETE/QUEUE command incorrectly states that the DELETE/QUEUE command requires OPER privilege. This command can be executed with either OPER privilege or EXECUTE access to the queue.

4.8.4 F\$FAO Lexical Function

- V5.4-3 In the description of the F\$FAO lexical function, the last example is incorrect. The corrected example appears below:

Example

```
$ OFFSPRING = 1
$ REPORT = F$FAO -
  $ ("There !OUL!1%Cis!%Eare!%F !-!UL !-!OUL!1%Cchild!%Echildren!%F here",1)
$ SHOW SYMBOL REPORT
$ REPORT ="There is 1 child here"
```

Documentation Release Notes

4.8 VMS DCL Dictionary

4.8.5 F\$GETDVI Lexical Function

V5.5 A sentence has been added to the F\$GETDVI command description which says, "Unless otherwise stated in the description of the item argument, F\$GETDVI returns device information about the local node only."

4.8.6 F\$GETQUI Lexical Function

V5.5 The table of keywords for the F\$GETQUI flags argument is incorrect. The table lists the GENERIC keyword as valid for only the DISPLAY_QUEUE function code. The GENERIC keyword is also valid for the DISPLAY_ENTRY function code.

Also, the following sentence has been added to the F\$GETQUI lexical function item code LOG_SPECIFICATION description: "Use the JOB_LOG_NULL item code to determine whether a log file will be produced."

4.8.7 INITIALIZE/MEDIA_FORMAT=[NO]COMPACTION Command

V5.5 In previous versions of VMS, the /MEDIA_FORMAT=[NO]COMPACTION qualifier for the INITIALIZE command controlled whether data records were automatically compacted and blocked together on a TA90E tape drive.

With VMS Version 5.5, there are many devices that support the /MEDIA_FORMAT qualifier. Support is no longer restricted to the TA90E tape drive. The qualifier controls whether data records are automatically compacted and blocked together on any device that supports data compaction.

4.8.8 PRINT/PRIORITY Command

V5.5 The description of the PRINT/PRIORITY command states that its use requires OPER (operator) or ALTPRI (alter privilege) privilege to raise the job scheduling priority above the value of the SYSGEN parameter MAXQUEPRI.

The description has been changed to read "... above the value of the queue's maximum scheduling priority."

In addition, the last sentence of the description, which states that no privilege is needed to set the job scheduling priority lower than the MAXQUEPRI value, now reads "... lower than the queue's maximum scheduling priority."

4.8.9 REPLY Command

V5.5 The *message-text* parameter for the REPLY command incorrectly states that the text of a message can be from 1 to 128 characters in length.

The description has been changed to read "... correct length is 1 to 511 characters."

4.8.10 RUN (Process) Command

V5.5 The description of the RUN (Process) command in the *VMS DCL Dictionary* states that a detached process is created if you specify the /UIC qualifier.

The RUN command creates a detached process if you specify either the /UIC or the /DETACHED qualifier.

The following changes affect the RUN/[NO]AUTHORIZE command:

- When you specify the /NOAUTHORIZE qualifier, quotas are derived from the SYSGEN parameters that set process quota default limits (parameters prefixed by PQL_D).

- When you specify the /AUTHORIZE qualifier, quotas are derived from the user authorization file (UAF) record of the process' owner. Any qualifiers to the RUN command that specify other quotas are ignored in favor of these UAF-based quotas.

4.8.11 SET CLUSTER/EXPECTED_VOTES Command

V5.5 The following sentence has been added to the SET CLUSTER/EXPECTED_VOTES command description:

Note that no matter what value you specify for the SET CLUSTER/EXPECTED_VOTES command, you cannot increase quorum to a value that is greater than the number of the votes present, nor can you reduce quorum to a value that is half or fewer of the votes present.

4.8.12 SET DEVICE/SERVED Command

V5.5 A warning has been added to the description of the SET DEVICE/SERVED command that reads "SET DEVICE/SERVED cannot be used with VMS Volume Shadowing."

Using SET DEVICE/SERVED in the wrong manner can cause deadlocks and jeopardize the cluster.

4.8.13 SET ENTRY/AFTER Command

V5.5 The description of the SET ENTRY/AFTER command should state that, when you specify /AFTER for a job on hold, you must also specify /NOHOLD in order to cause the job to be held only until the specified time.

4.8.14 SET ENTRY/PRIORITY Command

V5.5 The description of the SET ENTRY/PRIORITY command states that its use requires OPER (operator) or ALTPRI (alter privilege) privilege to raise the job scheduling priority above the value of the SYSGEN parameter MAXQUEPRI.

The description has been changed to read "... above the value of the queue's maximum scheduling priority."

In addition, the last sentence of the description, which stated that no privilege is needed to set the job scheduling priority lower than the MAXQUEPRI value, has been changed to read "... lower than the queue's maximum scheduling priority."

4.8.15 SET ACL/OBJECT_TYPE Command

V5.5 The description of the LOGICAL_NAME_TABLE keyword of the SET ACL/OBJECT_TYPE command states that the object is a system logical name table.

The description should read that the object is a *shareable* logical name table.

4.8.16 SHOW SYSTEM/INTERACTIVE Command

V5.5 The description of the SHOW SYSTEM command should include the /INTERACTIVE qualifier. The SHOW SYSTEM/INTERACTIVE command displays all interactive processes on the system.

Documentation Release Notes

4.8 VMS DCL Dictionary

4.8.17 SUBMIT/PRIORITY Command

V5.5 The description of the SUBMIT/PRIORITY command states that its use requires OPER (operator) or ALTPRI (alter privilege) privilege to specify a job scheduling priority greater than the value of the SYSGEN parameter MAXQUEPRI.

The description is changed to read “. . . greater than the value of the queue’s maximum scheduling priority.”

In addition, the last sentence of the description, which states that no privilege is needed to set the job scheduling priority lower than the MAXQUEPRI value, now reads “. . . lower than the queue’s maximum scheduling priority.”

4.9 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 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.

Example 4-1 is the new callback routine for the Hello World! sample application. Replace the existing Example 2-18 with Example 4-1.

- 1 The variable *newlabel* holds the address of the compound string used as the new push button label.
- 2 The new push button label, "Goodbye World!", is created using the compound string creation routine LATIN1 STRING.
- 3 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.

- 4 The memory allocated by LATIN1 STRING is freed using the FREE intrinsic routine.

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.)

Documentation Release Notes
4.9 VMS DECwindows Guide to Application Programming

Example 4-1 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 ;
    }
}

```

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 4-2.

Documentation Release Notes

4.9 VMS DECwindows Guide to Application Programming

Example 4-2 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_ARGNAMES (  
    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 ARGNAMES routine frees the memory allocated by the VMS SET ARG routine.

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.

4.10 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 4-1 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.

Table 4-1 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.

4.11 VMS DECwindows Xlib Programming Volume

V5.4 In Table 8-5 (section 8.3) 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: -Adobe-Helvetica-Bold-R-Normal--10-100-75-75-P-60-ISO8859-1
X\$C_XA_FAMILY_NAME	Atom	Name of the font family. For example: Helvetica
X\$C_XA_FULL_NAME	Atom	Full name of the font. For example: Helvetica Bold

In Table 8-5 (section 8.3) 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.

4.12 VMS Device Support Manual

The following sections describes changes to the *VMS Device Support Manual*.

4.12.1 DMA I/O Operations—Driver Code Reference

V5.4-3 Chapter 1 of the *VMS Device Support Manual*, Version 5.4, contains a section describing DMA I/O operations performed by device drivers. In the Direct-Memory-Access I/O section (1.7.2), two sentences at the end of the first paragraph provide references to driver code examples as follows:

“The VMS drivers DLDRIVER and XADRIVER are examples of DMA drivers. They appear in full in appendixes of the *VMS Device Support Reference Manual*.”

For the last sentence, substitute the following:

“They appear in full in Appendixes C and D of this manual (*VMS Device Support Manual*).”

Documentation Release Notes

4.12 VMS Device Support Manual

4.12.2 UNIBUS and Q22 Bus CSR Address Assignment

V5.4-3 Chapter 12 of the *VMS Device Support Manual*, Version 5.4, specifies the range of the floating CSR space for UNIBUS and Q22 bus customer third-party devices to be 766000₈ to 767776₈. The upper value (767776₈) is incorrect.

In the third paragraph, change the third sentence of Section 12.4.1, SYSGEN Device Table, to read:

“Reserved addresses 766000₈ to 766776₈ are recommended for customer third-party devices.”

4.12.3 VAX 8600/8650 I/O Offset Value

V5.4-3 Chapter 19 of the *VMS Device Support Manual*, Version 5.4, describes mapping to I/O space and lists symbolic addresses and offsets to I/O space of various systems supported. In Table 19-1, where it lists the symbols and their address/offset values on the VAX 8600/8650 system, the value for symbol IO790\$AL_UB0SP is incorrect. In the table under subhead VAX 8600/8650, change the value 24000000 for IO790\$AL_UB0SP to 100000 as follows:

IO790\$AL_UB0SP	Offset to start of adapter address space for first UNIBUS	100000
-----------------	---	--------

4.13 VMS Device Support Reference Manual

The following sections describe changes to the *VMS Device Support Reference Manual*.

4.13.1 COM\$SETATTNAST Routine

V5.4-3 Chapter 3 of the Version 5.4 *VMS Device Support Reference Manual* documents many of the operating system routines used by drivers. In the description of the COM\$SETATTNAST Routine under “synchronization,” it states that the routine raises IPL to the device IPL and returns control to the caller at the caller’s IPL. Change the entire synchronization text to read:

“COM\$SETATTNAST must be called from code executing at IPL\$_ASTDEL. It acquires the corresponding device lock to insert the AST into the AST queue. COM\$SETATTNAST returns control to the caller at IPL\$_ASTDEL.”

4.13.2 EXE\$ALLOCBUF and EXE\$ALLOCIRP Routine

V5.4-3 Chapter 3 of the *VMS Device Support Reference Manual*, Version 5.4, documents many of the operating system routines used by drivers. The description of EXE\$ALLOCBUF/EXE\$ALLOCIRP routine under “synchronization” states that the routines return control to their callers at the caller’s IPL. Change the text to read:

“They return control at IPL\$_ASTDEL”

4.13.3 SPI\$CONNECT Macro Return Values

V5.4-3 Chapter 2 of the *VMS Device Support Reference Manual*, Version 5.4, documents the function and operations of various driver macro routines. The macro SPI\$CONNECT (page 2-71) lists return values in various registers. This macro is used in SCSI device drivers. The return value for R1 is omitted. Add a return for R1 in the macro return table as follows:

R1	Maximum byte count allowed (SPDT\$L_MAXBYTECNT) for a data transfer.
----	--

4.13.4 SPI\$GET_CONNECTION_CHAR and SPI\$SET_CONNECTION_CHAR Macro Buffer Format

V5.4-3 Chapter 2 of the *VMS Device Support Reference Manual*, Version 5.4, documents the function and operations of various driver macro routines. The macros SPI\$GET_CONNECTION_CHAR and SPI\$SET_CONNECTION_CHAR (pages 2-75 and 2-89) describe the structure of the characteristics buffer listing each longword in positional order. The order of two longwords listed in the table shows number 7 as the select-retry-count longword and number 8 as the arbitration-retry-count longword. Reorder the longword entries in the table as follows:

Longword	Contents
.	.
.	.
.	.
7	Arbitration retry count. Maximum number of retries allowed on this connection while waiting for the port to win arbitration of the bus.
8	Select retry count. Maximum number of retries allowed on this connection while waiting for the port to be selected by the target device.
.	.
.	.

4.14 VMS Installation and Operations Manuals—Title Changes

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 *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*.

4.15 VMS I/O User's Reference Manual: Part I

V5.4 In Section 1.3.5.1 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:

Documentation Release Notes

4.15 VMS I/O User's Reference Manual: Part I

A Ctrl/S is sent if the type-ahead buffer contains less than TTY_ALTALARM bytes

4.16 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, 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.

4.17 VMS Monitor Utility Manual

- V5.2 In the MONITOR command section of the *VMS Monitor Utility Manual*, the Network Control Program (NCP) example of the MONITOR CLUSTER command is no longer valid. Refer to the following example instead:

```
$ 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=NOSYSRV
```

- V5.0 In the MONITOR Examples section, 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 might also want to consider raising the working set extent from 512 to 1024 or higher.

4.18 VMS Networking Manual

- V5.2 In Section 3.6.3 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 example:

```
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`:

```
NCP> DEFINE LINE TT-0-0 RECEIVE BUFFERS 6
```

V5.0

In 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.

4.19 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)
```

Documentation Release Notes

4.19 VMS Record Management Services Manual

- 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 ...

4.20 VMS RTL Library (LIB\$) Manual

- 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—Invoke \$TRNLOG System Service to Translate Logical Name

The Invoke \$TRNLOG System Service to Translate Logical Name routine uses the system service \$TRNLOG to translate a logical name. LIB\$SYS_TRNLOG returns the logical name's translation using the semantics of the caller's string.

Format

```
LIB$SYS_TRNLOG logical-name [,word-integer-dest-length] ,destination-string
                [,byte-integer-table] [,access-mode] [,byte-integer-disable-mask]
```

Returns

VMS Usage: cond_value
 type: longword (unsigned)
 access: write only
 mechanism: by value

Arguments

logical-name

VMS Usage: logical_name
 type: character string
 access: read only
 mechanism: by descriptor

Logical name. The **logical-name** argument contains the address of a descriptor pointing to this logical name string.

word-integer-dest-length

VMS Usage: word_unsigned
 type: word (unsigned)
 access: write only
 mechanism: by reference

Number of characters written into **destination-string**, not counting padding in the case of a fixed-length string. The **word-integer-dest-length** argument contains the address of an unsigned word integer that is this number.

If the input string is truncated to the size specified in the **destination-string** descriptor, **word-integer-dest-length** is set to this size. Therefore, **word-integer-dest-length** can always be used by the calling program to access a valid substring of **destination-string**.

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.

LIB\$SYS_TRNLOG

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

4.21 VMS RTL Screen Management (SMG\$) Manual

4.21 VMS RTL Screen Management (SMG\$) Manual

V5.4 In Section 1.1 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*.

4.22 VMS System Generation Utility Manual

V5.4-1 The CONNECT/ADAPTER=*adapter-spec* command description in the *VMS System Generation Utility Manual* defines adapter specification as:

“...the name of the Unibus or mass bus adapter to which the device is attached. The value can be expressed as an integer or as one of the names listed by the SYSGEN command SHOW/ADAPTER.”

Substitute the following definition for adapter specification:

“The adapter specification is the nexus number assigned to the device. You can find the nexus number by executing the SYSGEN SHOW/ADAPTER or SHOW/BUS commands.”

VMS does not support the use of named equivalents for nexus numbers such as UB0 on all machines.

4.23 VMS System Services Reference Manual

V5.5 The description of the \$ENQ system service states that there is an optional **range** argument. Since the **range** argument has not been implemented, ignore all discussion about it within this system service.

4.24 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.

4.25 VMS Upgrade and Installation Supplement: VAX 8600, 8650

V5.4 In Section 4.1.4 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 CSI
```

4.26 VMS Utility Routines Manual

V5.4 In Section 2.3 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.
-------------------------	---

4.27 VMS Version 5.4-3 Release Notes

V5.5 Chapter 8 of the VMS Version 5.4-3 Release Notes describes coding and linking requirements for VMEbus device drivers. In Section 8.1.8, the following paragraph is in error:

The VME support routines described in *VMS Device Support Reference Manual* are supplied in a separate object library to which the driver must link. Place the PSECT (\$\$\$112_VME_SUPPORT_ROUTINES) containing the VME support routines after the prologue PSECT and just ahead of the main driver code. For information on other PSECTs, see the *VMS Device Support Manual*.

Replace the above paragraph with the following:

The VME support routines described in *VMS Device Support Reference Manual* are supplied in a separate object library to which the driver is linked. When the driver is linked to include the VME support routines (as described in Section 8.1.9), the routines will reside in PSECT \$\$\$112_VME_SUPPORT of the resulting image. For information on other PSECTs, see the *VMS Device Support Manual*.



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.

A.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.

A.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 is ultimately limited by the amount of workstation memory available for this purpose.

VMS DECwindows Performance Considerations

A.2 Running VMS DECwindows Applications Remotely

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 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.

A.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 SYS$SYSTEM:VUE$MASTER
```

Run the previous command procedure as a noninteractive detached process, using the following command:

```
$ RUN SYS$SYSTEM: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.

A.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 might 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:

```
$ @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT CHECK_FEEDBACK
```

See the *Guide to Setting Up a VMS System* for more details about AUTOGEN.

A.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 or GBLSECTIONS, or both, with these additional images installed as shareable images. See the *VMS Install Utility Manual* for further information.



Building and Copying a VMS System Disk

V5.4 This appendix contains replacement text for the section “Building and Copying a VMS System Disk” in the *Guide to Setting Up a VMS System*.

B.1 Introduction

The command procedure `SYS$UPDATE:VMSKITBLD.COM` is used for building and copying a VMS system disk. The procedure gives you 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.

B.2 Building the Operating System on Another Disk

You might want to create a clean, new system disk without having to install the operating system. You can use the `VMSKITBLD BUILD` option to copy the operating system files from an existing system disk to another disk, without copying layered products, user files, UAF modifications, and other changes you may have made to the disk. For example, suppose your operating system is stored on an RA60 disk together with some of your user files. You have purchased an RA81 disk, and 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 copying any additional 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.

Building and Copying a VMS System Disk

B.2 Building the Operating System on Another Disk

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.
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 give you information about 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]:  Return
* Enter TARGET disk name (ddcu): DUA0:
* Enter the TARGET disk's label [default = VAXVMSRL5]:  Return
* Enter TARGET disk top level system directory [default = SYS0]:  Return
  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
```

Building and Copying a VMS System Disk

B.2 Building the Operating System on Another Disk

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
```

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.

B.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.

Building and Copying a VMS System Disk

B.3 Copying the Operating System Files to Another Disk

The following example shows a typical message sequence for a single-node system:

```
* Enter mounted SOURCE disk name (ddcu:): SYSSYSDEVICE:
* 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.

B.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.

Caution

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:

```
$ @SYSSUPDATE: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 give you information about the procedure's status.

Building and Copying a VMS System Disk B.4 Adding an Alternate System Root Directory

7. When you are prompted for the mounted source disk name, type `SYS$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's use.

A typical example using the `VMSKITBLD ADD` option might look like this:

```
* Enter mounted SOURCE disk name (ddcu):  SYS$SYSDEVICE:
* Enter SOURCE top level system directory [default = SYS0]:  [Return]
* Enter TARGET disk name (ddcu):  SHEMP$DUA5:
* Enter TARGET disk top level system directory [default = SYS0]:
%DCL-I-ALLOC,  SHEMP$DUA5: allocated
%MOUNT-I-MOUNTED, VAXVMSRL5  mounted on  SHEMP$DUA5:
  Creating system specific directories ...
  Creating SYSGEN files ...
%SYSGEN-I-CREATED,  SHEMP$DUA5:<SYSA.SYSEXE>SWAPFILE.SYS;1 created
%SYSGEN-I-CREATED,  SHEMP$DUA5:<SYSA.SYSEXE>PAGEFILE.SYS;1 created
%SYSGEN-I-CREATED,  SHEMP$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:

```
$ @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT CHECK_FEEDBACK
```

See the *Guide to Setting Up a VMS System* for detailed information about `AUTOGEN`.



File Design Attributes

V5.4 This appendix contains replacement text for Section 3.2.1 of the *Guide to VMS File Applications*.

C.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.

C.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$L_ALQ`.

C.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

C.1 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.

C.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 can increase file fragmentation.

C.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 C.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
- 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.

C.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.

File Design Attributes

C.1 File Design Attributes

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.

- 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.

C.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.

File Design Attributes

C.1 File Design Attributes

- 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.

C.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 can 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.

C.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.

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 can include up to 63 blocks.

C.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.

C.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.

File Design Attributes

C.1 File Design Attributes

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

- Access
 - required for DELETE/QUEUE command, 4-3
 - required for SJC\$_DELETE_QUEUE, 4-18
- Accounting record
 - page count, 1-3
- Address
 - NML address check, 2-15
- AllocAll flag
 - creating colormap with, 3-15
- Alternate root directory
 - adding to an existing system disk, B-4
- Asynchronous option
 - VMS RMS support, 4-14
- Authorize Utility (AUTHORIZE)
 - adding proxy accounts, 2-3
 - ADD/PROXY and MODIFY/PROXY commands, 2-3
 - restricted flag modifications, 2-3
- AUTOGEN
 - privileges required to run, 2-4
 - running DECW\$STARTUP.COM command
 - procedure, 2-16
 - switching window systems, 2-5
- AUTOGEN operations
 - on a VAX 3100 Model 76, 2-60
- Autostart feature, 2-49, 2-50

B

- BACKUP command
 - with the TA90E tape drive, 2-52
- Backup Utility (BACKUP), 2-6
 - documentation, 4-2
 - effect on ACLs, 2-6
 - files with active transactions, 2-6
 - image save operation restriction, 2-7
 - problems and restrictions, 2-6
 - using with compound document files, 2-7
- Bad block support for SCSI disks, 2-37
- BASIC
 - See VAX BASIC
- Batch job
 - deleting, 1-2, 1-4
- Batch/Print changes
 - supported versions, 2-7

- Batch/Print changes (Cont.)
 - system startup and shutdown, 2-49
- Batch/Print Facility, 2-9
 - supported versions, 2-7
- Boot procedures for XDELTA
 - See entries for specific computers
- Building dependable systems, 4-2
- BYTLM, 2-13

C

- Calendar
 - See DECwindows applications
- CALENDAR.SPLIT improvement, 3-31
- CAPTIVE flag (UAF), 2-45
 - new interpretation, 2-44
- CDA Base Services
 - get and put routines, 3-10
- CDA toolkit
 - new qualifier, 1-16
- CDA Viewer
 - See DECwindows
- Changing print characteristics, 2-9
- Changing print forms, 2-9
- Changing print stock, 2-9
- Character
 - printable, 1-1
- Characteristic
 - changing, 2-9
- Circuit cost, 2-16
- CIXCD, 2-56
- Clock
 - See DECwindows applications
- CLOSE procedures (VAX Ada), 3-31
- Cluster
 - adding proxy accounts, 2-3
- COBRTL installation, 2-13
- Command verb and qualifier length, 1-16
- Compound document
 - sending, 3-10
- CONNECT/ADAPTER command
 - change to syntax, 4-18
- COPY/TRUNCATE command, 4-3
- CREATE PORT command, 2-33
- CREATE PORT/LOGICAL command, 2-33
- CREATE SERVICE command, 2-33
- Cursor colors

Cursor colors (Cont.)
 updating, 3-14
Cursor patterns, 3-12
Cursor screen boundaries, 3-12

D

DAP (DECnet file access protocol)
 extensions, 3-10
Data record compaction
 TA90E support, 2-52
DCL (DIGITAL Command Language)
 CALL command, 3-2
 command verb and qualifier length, 1-16
 Debugger commands, 3-5
 DEFINE/FORM command, 1-2
 ENABLE AUTOSTART/QUEUES, 2-49
 expiration of RMS disk files, 3-39
 F\$CONTEXT, 1-18
 IF-THEN-ELSE construct, 3-22
 label scoping in, 3-2
 OPEN command, 1-17
 PRINT command
 /PAGES qualifier, 1-2
 SET DISPLAY command, 3-15
 SET HOST/DTE command, 1-17, 3-30
 SHOW DEVICES/FULL command, 1-18
 SHOW ENTRY, 2-11
 SHOW MAGTAPE command, 1-18
 SHOW QUEUE, 2-11
 SUBMIT/DELETE command, 1-3
 SUBROUTINE command, 3-2
 subroutine entry points, 3-2
 substring assignment, 3-3
DEBNI Ethernet/802 controller, 2-25
Debugger
 commands used in DCL command procedures,
 3-5
 corrected problems and restrictions, 3-3
 examining LABEL[n], 3-5
 problems with DECwindows interface, 3-6
 using ABORT key after SPAWN command, 3-6
 using concealed rooted-directory logical names,
 3-5
 using on VAXstation workstation, 3-6
 using Stop button after SPAWN command, 3-6
DEC\$SYLOGIN.COM file, 2-17
DEC\$SYLOGIN.TEMPLATE file, 2-17
DECburger sample application
 corrections, 4-6
DECdtm files
 required for the batch and print queuing
 system, 2-8
DECnet
 links, 2-14
 retransmission timeout period, 2-15
 starting in a local area VAXcluster, 2-15
DECnet file access protocol
 DECnet file access protocol (Cont.)
 See DAP
 DECnet software
 default account, 2-40
 starting during DECwindows session, 1-9
DECnet-VAX
 SYS\$CLUSTER_NODE logical, 2-16
DECterm
 See DECwindows
DECthreads, 3-10
DECwindows, 1-4
 applications
 Calendar restrictions, 1-9
 CDA Viewer restrictions, 1-11
 Mail, 1-12
 performance, A-1
 running remotely, A-1
 cutting and pasting, 1-4
 debugger problems, 3-6
 DECterm
 conformance level, 1-4
 corrected color table report problem, 3-11
 graphics, 1-5
 initializing, 1-6
 memory, 1-7
 negative values correction, 3-11
 ReGIS locator report, 3-11
 text, 1-8
 VT52-mode cursor addressing, 3-11
 window scrolling problem, 1-8
 DECW\$COLOR guidelines, 1-9
 DECW\$STARTUP.COM procedure
 running AUTOGEN.COM command
 procedure, 2-16
 font properties, 4-9
 minimum memory, A-1
 multihead system support, 2-17
 Print Screen function, 1-12
 server, 3-11
 startup, 2-16
 startup problem, 1-9
 tailoring on RD53 system disk, 2-17
 terminal emulator
 See DECterm
 UIL built-in tables, 3-16
 UIL corrections, 3-16
 ULTRIX authorization requirements, 1-14
 Window Manager icon box, 1-15
 DECwindows cut and paste operation, 1-4
 DECwindows Mail, 3-10
 DECwindows server logical switch for server
 connect/disconnect messages, 2-16
 DECwindows X11 Display Server, 3-14
 PostScript (DPS) Extension, 3-14
 updating cursor colors, 3-14
DEFAULT print form
 deleting, 2-8
DEFINE/CHARACTERISTIC command, 2-9

DEFINE/FORM command, 2-9, 4-3
 DEFINE LINE command, 4-12
 DEFPRI SYSGEN parameter, 2-12
 DEFQUEUEPRI SYSGEN parameter, 2-9, 2-12
 DELETE/ENTRY command, 1-2
 DELETE/QUEUE command
 privilege or access required for, 4-3
 Deleting batch and print job entries, 1-2
 Deleting files
 after printing or submitting, 1-4
 Deleting jobs on queues set /RETAIN qualifier, 1-2
 Deleting the DEFAULT print form, 2-8
 Delta/XDelta Utility (DELTA/XDELTA)
 invoking XDELTA, 3-47
 DEMNA controller
 firmware revision level, 2-73
 DEMNA Ethernet/802 controller, 2-26
 Department of Defense (DoD) erase pattern, 2-39
 Dependability, 4-2
 DEQTA Ethernet/802 controller, 2-26
 Device
 terminal configuration, 2-39
 DEVICE_SCAN, 3-28
 Digital Small Computer System Interface (SCSI)
 devices, 2-36
 Digital Storage Architecture devices
 See DSA devices
 DIRECTORY command, 1-16
 DISABLE_AUTOSTART option
 in SHUTDOWN.COM, 2-50
 Disk
 DSA drivers, alternate host information, 3-21
 large-capacity disks, header space problem, 2-18
 Disk forced error counting, 3-20
 DISMOUNT command
 processing open files, 2-20
 Display
 change in SHOW QUEUE and SHOW ENTRY commands, 2-11
 Display PostScript documentation, 3-15
 DNS\$CVT_TO_USERNAME_STRING routine, 2-23
 DNS\$PARSE_USERNAME_STRING routine, 2-23
 DNS (Distributed Name Service)
 RTL routines, 2-23
 Documentation
 new, 4-2
 DQS
 See VAX Distributed Queuing System (DQS)
 DRM routines
 unavailable VAX bindings for, 3-18
 DSA devices
 preferred path support, 3-21
 DSSI (DIGITAL Storage System Interconnect)

DSSI (DIGITAL Storage System Interconnect)
 (Cont.)
 device naming, 2-23
 DUMP command, 1-18

E

EFDRIIVER OPCOM messages, 2-58
 ENABLE AUTOSTART/QUEUES command, 2-49
 Entries
 deleting for batch and print jobs, 1-2
 Erase pattern
 Department of Defense, 2-39
 Ethernet
 DEBNI controller, 2-25
 DEMNA controller, 2-26
 DEQTA controller, 2-26
 SGEC controller, 2-27

F

F\$CONTEXT, 1-18
 F\$GETDVI lexical function
 documentation corrections, 4-4
 F\$GETQUI lexical function
 documentation corrections, 4-4
 FAB\$V_ASY option
 documentation change, 4-14
 File
 deletion after printing or submitting, 1-4
 File design attributes, C-1
 Files required for the batch and print queuing system, 2-8
 Flag
 restricted modifications in Authorize Utility, 2-3
 Folder name parameters
 in Mail Utility, 1-21
 Fonts
 for layered products, 3-14
 narrow terminal, 3-15
 Form
 deleting the default, 2-8
 Forms
 changing, 2-9

G

\$GETQUI system service
 QUI\$_QUEUE_STATUS longword, 3-44
 \$GETSYI, 3-44 to 3-46

H

Hello World! sample application
 corrections, 4-6
 Hierarchical Storage Controller
 see HSC

HSC (hierarchical storage controller)
revision levels required, 2-27

I

I/O device drivers

ACP-QIO interface, 3-22
opening a sequential-media file, 3-22

I/O driver

logical end-of-volume detection, 3-22

Icon box, 1-15

IF-THEN-ELSE construct (DCL)

setting \$STATUS symbol, 3-22

Image data records

login echo problem, 1-19
verification, 1-19

INITIALIZE/MEDIA_

FORMAT=[NO]COMPACTION command, 4-4

INKEY\$ function, 3-2

Internal processor register

See IPR

Interrupt request for XDELTA, 3-54

See also entries for specific processors

Invalid qualifiers

removed from print commands, 1-3

IPC (Interprocess Communication) files required

for the queuing system, 2-8

IPR (internal processor register)

definition symbols, 3-27

J

Job controller

starting queue manager, 2-10

Job scheduling priority, 2-9

L

Label scoping in DCL, 3-2

Language

reinstalling, 3-28

LAT (local area transport), 3-22

LAT\$SYSTARTUP.COM, 2-29

LTLOAD.COM, 2-29

setting up, 2-29

starting XTERMINAL software, 2-29

SYSTARTUP_V5.COM, 2-29

LATCP (LAT Control Program)

commands replaced, 2-29

installing, 2-29

LATSYM print symbiont, 2-31

new qualifiers for SET NODE command, 2-30

obsolete command, 2-31

obsolete qualifiers, 2-30, 2-31

privileges required, 2-29

setting LTA MAX units, 2-32

starting LATACP, 2-31

stopping LATACP, 2-31

LATCP (LAT Control Program) (Cont.)

stopping LTDRIVER, 2-31

using FDDI Controller, 2-33

using SET NODE command, 2-31

LAT QIO interface, 3-22

Layered products

accessing fonts, 3-14

upgrade caution, 2-56

Lexical function

F\$CONTEXT, 1-18

LIB\$DECODE_FAULT

use with vector processor, 3-29

License Management Facility

See LMF

Link

problem with dropping, 2-14

Linker

combining psects, 3-25

image-ID field, 4-12

Linking image

against system table of different VMS version,
3-1

Local area VAXclusters

starting DECnet, 2-15

Lock manager, 3-25

LTDRIVER, 2-33

M

MACRO

See VAX MACRO

Magnetic tape

ancillary control process, 2-2

automatic loading, 2-2

Magnetic tape ACP

correction to I/O, 2-33

Mailbox driver error checking, 3-43

Mailbox driver write, 3-26

Mail Utility (MAIL)

folder name parameter, 1-21

multiple copies of message to same recipient,
1-21

PRINT/QUEUE command changes, 1-19

use of quotes in forwarding, 1-22

Mass storage device

DSSI device naming change, 2-23

MAXQUEPRI SYSGEN parameter, 2-9, 2-12

Memory

minimum recommended for DECwindows, A-1

Message Router

installation restriction, 3-26

Messages

EFDRIVER OPCOM messages, 2-58

MicroVAX/VAXstation 3600-series computer

boot procedure for XDELTA, 3-49

requesting interrupt, 3-55

MicroVAX 2000 computer

boot procedure for XDELTA, 3-49

MicroVAX 2000 computer (Cont.)
 requesting interrupt, 3-55
 MicroVAX 3400-series computer
 boot procedure for XDELTA, 3-49
 requesting interrupt, 3-55
 MicroVAX 3900-series computer
 boot procedure for XDELTA, 3-49
 requesting interrupt, 3-55
 MicroVAX II computer
 boot procedure for XDELTA, 3-49
 requesting interrupt, 3-55
 Modem signal control, 3-27
 Modifying print characteristics, 2-9
 Modifying print forms, 2-9
 Monitor Utility (MONITOR), 2-35, 4-12
 MONITOR CLUSTER command, 2-34
 Motif Developer Kit
 linker incompatibility, 2-57
 MOUNT/AUTOMATIC command, 2-2
 Mounting of queue file disk, 2-10
 Mouse Key Combination Change, 1-13

N

NCP (Network Control Program), 2-14
 DEFINE LINE command, 4-12
 number of receive buffers, 4-12
 Remote Buffer Errors counter, 4-12
 SET LINE command, 4-12
 SHOW CIRCUIT command, 4-12
 SHOW LINE command, 4-12
 NETACP\$BUFFER_LIMIT logical name, 2-15
 NETACP (network ancillary control process), 2-15
 NETCONFIG.COM command procedure
 security enhancements, 2-40
 NETCONFIG_UPDATE.COM command procedure,
 2-40
 NETDRIVER, 2-14
 Network ancillary control process
 See NETACP
 Network Control Program
 See NCP
 Network default access
 controlling access to your system, 2-40
 Network management listener
 See NML
 /NEW_VERSION qualifier
 of START/QUEUE/MANAGER command, 2-10
 NML (network management listener)
 check for illegal address, 2-15
 object, 2-15
 NUMERIC_ERROR exception (VAX Ada), 3-31

O

Obsolete qualifiers
 START/QUEUE/MANAGER command, 2-11
 OPCOM (operator communication manager)
 EFDRIIVER OPCOM messages, 2-58
 SECURITY class messages, 2-42
 OPEN command, 1-17
 Operating system
 copying files to another disk, B-3
 Operator Communication Manager
 See OPCOM

P

Page count
 in accounting record, 1-3
 Parameter
 setting in AUTOGEN, 2-4
 Password
 service, 2-14
 Password dictionary, 2-39
 PGFIPLHI crash, 2-52
 PHY_IO privilege
 for shadowed disks, 2-76
 PIOPAGES parameter, 3-43
 PPL\$TRIGGER_EVENT
 memory correction, 3-29
 Preferred path support for DSA devices, 3-21
 Print characteristics
 changing, 2-9
 PRINT command
 accounting record page count, 1-3
 PRINT/DELETE command, 1-4
 Printers
 LJ250, 1-12
 Print forms
 changing, 2-9
 deleting the default, 2-8
 Printing files
 accounting record page count, 1-3
 with the /DELETE qualifier, 1-4
 Print jobs
 deleting, 1-2, 1-4
 PRINT/PRIORITY command, 4-4
 Print Screen function (DECwindows), 1-12
 Print stock
 changing, 2-9
 Print symbiont
 purging working set, 2-10
 Priority
 scheduling, 2-9
 PRIORITY_OFFSET parameter (SYSGEN), 2-51
 Privilege
 required for DELETE/QUEUE command, 4-3
 required for SJC\$_DELETE_QUEUE, 4-18
 required to run AUTOGEN, 2-4

Process-permanent files
 VMS RMS asynchronous support, 4-14
PROCESS_SCAN, 3-28
Proxy account
 changes, 2-3

Q

Q-bus
 DEQTA Ethernet/802 controller support for,
 2-26
\$QIO call requirements
 setting multiscreen cursor pattern, 3-12
Qualifier
 invalid, 1-3
 obsolete, 2-11
Queue
 deleting jobs on queues set /RETAIN, 1-2
 deletion of print and batch files, 1-4
 execution, 2-9
 generic, 2-9
 print job accounting records, 1-3
 retrieving status of, 3-44
Queue database
 creating a new version, 2-10
Queue file
 mounting of disk holding, 2-10
Queue manager
 creating a new queue database, 2-10
Queue status
 change in behavior, 3-44
Queuing system
 required files, 2-8
 supported versions, 2-7
QUI\$_QUEUE_STATUS longword, 3-44

R

RAB\$_ASY qualifier
 process-permanent file support, 3-40
Record Management Services
 See VMS RMS
Reinstalling languages, 3-28
Remote buffer errors, 4-12
REPLY command, 4-4
RESTRICTED flag (UAF)
 new flag, 2-44
Retained jobs
 deleting, 1-2
Retransmission timeout period
 determination of, 2-15
Rights database, 2-3
RMS
 See Record Management Services
Root directory
 adding to an existing system disk, B-4
RTL (Run-Time Library), 3-29

RTL (Run-Time Library) (Cont.)
 LIB\$CREATE_VM_ZONE routine, new flags
 added, 3-28
 LIB\$SYS_TRNLOG routine, 4-14
RUN (Process) command, 1-17, 4-4
RUN/AUTHORIZE command, 1-17, 4-4
Run-Time Library
 See RTL

S

SCSI command, 2-36
SCSI notes, 2-36
SCS SYSGEN/TIMVCFAIL parameter, 2-72
SDA (System Dump Analyzer), 2-52
Second Generation Ethernet Controller (SGEC),
 2-27
Security
 auditing failure mode setting, 2-41
Security enhancements to NETCONFIG.COM
 for new systems, 2-40
Service passwords, 2-14
SET ACL/OBJECT_TYPE command, 4-5
Set cursor pattern \$QIO call, 3-12
SET DEVICE/SERVED command, 4-5
SET ENTRY/AFTER command, 4-5
SET ENTRY/PRIORITY command, 4-5
SET FILE/AI_JOURNAL command
 errors when creating duplicate journals, 3-42
SET FILE/BI_JOURNAL command
 errors when creating duplicate journals, 3-42
SET LINE command, 4-12
SET QUEUE /DEFAULT command, 2-12
SET VERIFY command, 1-19
SHADBOOTFAIL bugcheck message, 2-76
Shared memory
 error messages, 3-43
SHOW CIRCUIT command, 4-12
SHOW ENTRY command
 change in display, 2-11
SHOW LINE command, 4-12
SHOW QUEUE command
 change in display, 2-11
SHOW SYSTEM/INTERACTIVE command, 4-5
SHUTDOWN\$DISABLE_AUTOSTART logical
 name, 2-50
SHUTDOWN.COM
 changes, 2-50
Shutdown procedure
 changes, 2-50
SJC\$_DELETE_QUEUE function code
 privilege or access required for, 4-18
SJC\$_START_QUEUE_MANAGER function code,
 3-47
SMP
 See Symmetric Multiprocessing
SPAWN command, 2-41
Standalone BACKUP, 2-49

Standalone BACKUP (Cont.)
 command to boot from an RL02 disk, 4-19
 STARLET library symbols, 2-45
 START/QUEUE/MANAGER command
 change in behavior of /NEW_VERSION
 qualifier, 2-10
 obsolete qualifiers, 2-11
 Startup
 mounting of queue file disk, 2-10
 Startup procedure
 changes, 2-49
 Status
 of queues, 3-44
 Status bits
 changed, 3-44
 \$STATUS symbol
 set by IF-THEN-ELSE construct, 3-22
 Stock
 changing, 2-9
 Stopped bit
 in queue status longword, 3-44
 Stopped queues
 checking status of, 3-44
 STOP/QUEUE/RESET command, 2-11
 SUBMIT/DELETE command, 1-4
 SUBMIT/PRIORITY command, 4-6
 Submitting files
 with the /DELETE qualifier, 1-4
 Subroutine entry points in DCL, 3-2
 Substring assignments, 3-3
 SYCONFIG.COM
 mounting queue file disk, 2-10
 Symbol names
 making assignments, 1-18
 Symmetric Multiprocessing (SMP), 2-51
 SYNCHRONIZE command, 1-18
 SYS\$CLUSTER_NODE logical, 2-16
 SYS\$CREATE_RDB
 creation of rights database, 2-3
 SYS\$ERAPAT, 2-39
 SYS\$GETDVI
 using to obtain FREEBLOCK count, 2-76
 SYSGEN (System Generation Utility)
 PIOPAGES parameter, 3-43
 PQL_MPRCLM parameter, 1-14
 PRIORITY_OFFSET parameter, 2-51
 SYSLOA symbol, 2-52
 SYSTARTUP_V5.TEMPLATE
 changes, 2-49
 System disk
 building and copying, B-1
 shutdown for volume shadowing, 2-76
 System disk size, recommendation, 2-52
 System Dump Analyzer Utility
 See SDA
 System Generation Utility
 See SYSGEN
 System Management Utility (SYSMAN)

System Management Utility (SYSMAN) (Cont.)
 used by AUTOGEN, 2-4
 System services
 SYS\$GETQUI, 3-44
 SYS\$GETSYI, 3-44 to 3-46
 System Services error messages, 3-43
 System shutdown
 changes, 2-50
 System startup
 changes, 2-49
 mounting of queue file disk, 2-10
 System symbol table
 linking against, 3-1
 System User Authorization File
 See UAF

T

TA90E tape drive
 support for, 2-52
 using BACKUP command, 2-53
 using /MEDIA_FORMAT qualifier, 2-52
 Tape support
 for ANSI initialized magnetic tapes, 1-20
 TLZ04 tape drive
 performance, 1-20
 TLZ04 tape driver, 2-53
 TU58 console bootstrap procedures, 3-48

U

UAF (user authorization file)
 flags, 2-44
 UAF parameter
 changes for volume shadowing Phase II, 2-77
 UAI\$V_CAPTIVE symbol (STARLET), 2-45
 UAI\$V_RESTRICTED symbol (STARLET), 2-45
 UETP Load Phase Failure in Vector Processing
 Systems, 2-53
 UIL compiler, 3-16
 valid tables changes, 3-16
 UNA circuit
 change in default cost, 2-16
 Upgrading VMS systems
 VAXstation 8000 unsupported, 2-58
 User Authorization File
 See UAF
 User EOT mode, correction, 3-22
 User Interface Language Compiler
 See UIL compiler

V

VAX-11/730 computer
 boot procedure for XDELTA, 3-48
 requesting interrupt, 3-55
 VAX-11/750 computer
 boot procedure for XDELTA, 3-49

- VAX-11/750 computer (Cont.)
 - boot procedure for XDELTA with TU58 console, 3-49
 - requesting interrupt, 3-55
- VAX-11/780 computer
 - boot procedure for XDELTA, 3-50
 - requesting interrupt, 3-55
- VAX-11/785 computer
 - boot procedure for XDELTA, 3-50
 - requesting interrupt, 3-55
- VAX 6000 computer
 - boot procedure for XDELTA, 3-50
 - requesting interrupt, 3-55
- VAXstation 3100-series computer
 - boot procedure for XDELTA, 3-54
- MicroVAX 3100-series computer
 - boot procedure for XDELTA, 3-54
- VAX 3100 Model 76
 - AUTOGEN operations, 2-60
- VAX 4000 Model 300 computer
 - boot procedure for XDELTA, 3-49
 - requesting interrupt, 3-55
- VAX 6000
 - console tapes and tape serving devices, 2-60
 - Model 500 configuration of CPUs, 2-62
- VAX 8000-series systems
 - SET TIME/CLUSTER command, 2-63
 - SET TIME command, 2-63
 - VAXBI restriction, 2-64
- VAX 8200 computer
 - boot procedure for XDELTA, 3-51
 - requesting interrupt, 3-55
- VAX 8250 computer
 - boot procedure for XDELTA, 3-51
 - requesting interrupt, 3-55
- VAX 8300 computer
 - boot procedure for XDELTA, 3-51
 - requesting interrupt, 3-55
- VAX 8350 computer
 - boot procedure for XDELTA, 3-51
 - requesting interrupt, 3-55
- VAX 8530 computer
 - boot procedure for XDELTA, 3-51
 - requesting interrupt, 3-55
- VAX 8550 computer
 - boot procedure for XDELTA, 3-51
 - requesting interrupt, 3-55
- VAX 8600 computer
 - boot procedure for XDELTA, 3-52
 - requesting interrupt, 3-55
- VAX 8650 computer
 - boot procedure for XDELTA, 3-52
 - requesting interrupt, 3-55
- VAX 8700 computer
 - See VAX 8810
- VAX 8800 computer
 - See also VAX 8820-N
 - deadlock situation, 2-63
- VAX 8810 computer
 - boot procedure for XDELTA, 3-51
 - requesting interrupt, 3-55
- VAX 8820 computer
 - boot procedure for XDELTA, 3-51
 - requesting interrupt, 3-55
- VAX 8820-N computer
 - boot procedure for XDELTA, 3-51
 - requesting interrupt, 3-55
- VAX 8830 computer
 - boot procedure for XDELTA, 3-51
 - requesting interrupt, 3-55
- VAX 8840 computer
 - boot procedure for XDELTA, 3-51
 - requesting interrupt, 3-55
- VAX 9000 computer
 - AUTOGEN parameter calculations for, 2-64
 - BI device driver requirement, 3-30
 - boot procedure for XDELTA, 3-53
 - requesting interrupt, 3-55
- VAX Ada
 - CLOSE procedures change, 3-31
 - restrictions, 3-32
- VAX Ada Run-Time Library
 - release notes for, 3-30
- VAX BASIC
 - INKEY\$ function change, 3-2
- VAXBI bus
 - DEBNI Ethernet/802 controller support, 2-25
 - VAXBI restriction, 2-64
- VAX C
 - Run-Time Library error checking, 3-32
- VAXcluster
 - boot disk path, 2-72
 - DEMNA firmware revision level for, 2-73
 - FDDI adapter support, 2-72
 - LAN adapter support, 2-72
 - multiadapter support, 2-72
 - reconfiguration time reduction, 2-72
 - support, 2-70
- VAXcluster environment
 - deletion of print and batch files, 1-4
 - supported versions of the batch and print queuing system, 2-7
 - SYSGEN parameter recommendation, 2-12
- VAX computers, VMS support for, 2-64
- VAX C Run-Time Library, 3-33
- VAX Distributed Queuing System (DQS)
 - recommended version, 2-13
- VAXft 3000 computer
 - boot procedure for XDELTA, 3-54
 - EFDRIVER OPCOM messages, 2-58
 - requesting an interrupt for VAXft 3000, 3-55
- VAX MACRO, 3-33
- VAX Math Run-Time Library, 3-38
- VAXstation 2000 computer
 - boot procedure for XDELTA, 3-49
 - requesting interrupt, 3-55

- VAXstation 3520 and 3540 computers
 - Ctrl/F2 key sequence, 2-65
 - Print Screen restriction, 1-12
 - supported software products, 2-65
- VAXstation 3520 and 3540 screen blank out, 1-8
- VAXstation 3520 computer
 - boot procedure for XDELTA, 3-49
 - requesting interrupt, 3-55
- VAXstation 3540 computer
 - boot procedure for XDELTA, 3-49
 - requesting interrupt, 3-55
- VAXstation 4000 series computer
 - changing font size, 2-65
- VAXstation 8000 computer
 - upgrade information, 2-58
- VAXTPU (VAX Text Processing Utility)
 - /WORK and /NOWORK qualifiers, 1-21
- Verification
 - image data records, 1-19
- VIRTUALPAGECOUNT parameter
 - adjustment for Volume Shadowing (Phase II), 2-76
- VMSKITBLD procedure, B-1
- VMS License Management Facility (LMF), 2-66
- VMS RMS
 - CONVERT/RECLAIM Utility, 3-38
 - new local buffers default, 3-41
 - RAB\$_ASY qualifier, 3-40
 - statistics restrictions, 3-41
 - XAB\$_NORECORD, 3-39
- VMS RMS Journaling, 3-42
 - synchronous recovery, 3-42
- VMS System Generation Utility Manual, 4-18
- VMS versions
 - computer support, 2-64
- Volume shadowing, 2-35
 - PHY_IO privilege, 2-76
 - shutdown with shadowed system disk, 2-76

- Volume shadowing (Phase II)
 - changing device names, 2-74
 - implications for batch and print jobs, 2-74
 - overview, 2-74
 - SHADBOOTFAIL bugcheck message, 2-76
 - UAF parameter changes, 2-77
 - VIRTUALPAGECOUNT parameter adjustment, 2-76

W

- Widget
 - redrawing in XUI Toolkit, 3-19
- Window Manager
 - See DECwindows
- Window systems
 - switching with AUTOGEN, 2-5
- Workstation
 - with dual-head support, 1-15

X

- XDELTA
 - invoking, 3-47
- XMI bus
 - DEMNA Ethernet/802 controller support, 2-26
- X servers
 - interoperability with other vendors' X servers, 2-17
- XUI Toolkit
 - redrawing widgets, 3-19
 - unavailable VAX bindings for DRM routines, 3-18

Y

- YFDRIVER terminal port driver, 2-78



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).

