VMS
Version 5.3 Release Notes

Order Number: AA–MG16B–TE

October 1989

This document describes software problems, corrections, and restrictions, as well as documentation changes, that pertain to Version 5.3 of the VMS operating system.

Revision/Update Information:	This manual supersedes the *VMS Version 5.2-1 Release Notes*. It does not replace the *VMS Version 5.2 Release Notes*.
Software Version:	VMS Version 5.3

digital equipment corporation
maynard, massachusetts

October 1989

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

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

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

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

© Digital Equipment Corporation 1989.

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

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

The following are trademarks of Digital Equipment Corporation:

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

The following are third-party trademarks:

PostScript is a registered trademark of Adobe Systems, Inc.

MS-DOS is a registered trademark of Microsoft Corporation.

ZK5328

Production Note

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

Contents

PREFACE xiii

CHAPTER 1 GENERAL USER RELEASE NOTES 1–1

1.1 CHANGES IN ERROR CONDITIONS FOR PRINT AND SUBMIT COMMANDS 1–1

1.2 SUBMIT/DELETE COMMAND CHANGE 1–1

1.3 INITIALIZING UNFORMATTED DISKETTES WITHOUT VOLPRO PRIVILEGE 1–2

1.4 CHANGES TO VAXTPU 1–2
1.4.1 Changed DECwindows VAXTPU Initialization 1–2
1.4.2 Display Manager Definition Restriction 1–2

1.5 CHANGES TO EVE 1–3
1.5.1 Obsolete Keywords for EVE Message Constants 1–3
1.5.2 Known Restrictions for EVE 1–4

1.6 BATCH/PRINT FACILITY PERFORMANCE IMPROVEMENTS 1–5

1.7 VMS DECWINDOWS RELEASE NOTES 1–6
1.7.1 Window Manager (Icon Box) Corrections 1–6
1.7.2 FileView Changes 1–6
1.7.3 DECterm Corrections and Notes 1–8
1.7.3.1 Corrected DECterm Problems • 1–8
1.7.3.2 Initializing DECterm • 1–9
1.7.3.3 Keyboards and Languages Information • 1–10
1.7.3.4 Fonts and User Interface Information • 1–11
1.7.3.5 Text Information • 1–11
1.7.3.6 Graphics Information • 1–11
1.7.3.7 DECterm PC Interoperability Restrictions • 1–13
1.7.3.8 DECterm Memory Problem • 1–14
1.7.4 Help Widget Enhancement 1–14
1.7.5 Desktop Application Notes 1–14

1.7.5.1 Calendar Restrictions and Limitations • **1–14**
1.7.5.2 Mail Corrections and Restrictions • **1–14**
1.7.5.3 Paint Restrictions • **1–15**
1.7.5.4 PrintScreen Restrictions • **1–15**

1.8 VMS VERSION 5.2-1 RELEASE NOTES 1–16
1.8.1 Console Carrier Error Handling 1–16
1.8.2 SYSGEN INVVEC Error Fixed 1–16
1.8.3 RMS Journaling—Automatic Recovery Performance 1–16
1.8.4 Restriction on Number of Files in Magnetic Tape Volume Set 1–17

CHAPTER 2 SYSTEM MANAGER RELEASE NOTES 2–1

2.1 REDUCTION IN CLUSTER RECONFIGURATION TIME WITH QUORUM DISK 2–1

2.2 ETHERNET CIRCUIT LOOPBACK TESTS 2–1

2.3 XMDRIVER POWERFAIL RECOVERY 2–2

2.4 PAGE FAULT WITHIN SYSLOA PROBLEM FIXED 2–2

2.5 DEVICE NAMING CHANGE FOR DSSI DISKS CONNECTED TO A KFQSA CONTROLLER 2–2

2.6 MODIFYING LICENSE UNITS WITH THE LICENSE MANAGEMENT FACILITY 2–4

2.7 VMS SUPPORT FOR VAX PROCESSORS 2–8

2.8 VAXSTATION 3100 MODEL NAME CHANGE 2–8

2.9 VMSD2 SYSGEN PARAMETER AND SCSI DEVICES 2–8

2.10 VMS/RZ55 DATA INTEGRITY PROBLEM FIXED 2–9

2.11 USING THE RZ23 AS A SYSTEM DISK 2–12

2.12 LAT TERMINAL SERVICE PROBLEM 2–12

2.13 VMS DECWINDOWS RELEASE NOTES 2–13
2.13.1 New Template File 2–13
2.13.2 Adding DECwindows at a Later Date 2–13
2.13.3 ULTRIX Connection (UCX) and TCP/IP Transport Notes 2–13
2.13.3.1 Problem Starting UCX Before DECnet • 2–13
2.13.3.2 Sequence Lost Errors • 2–14

2.14 VMS VERSION 5.2-1 RELEASE NOTES 2–14
2.14.1 DECnet–VAX Notes 2–14
2.14.1.1 Fix for Repeated Maintenance Operations • 2–14
2.14.1.2 Downline Task Loading to RSX–11S Systems • 2–14
2.14.1.3 XDDRIVER Retransmit Timer Correction • 2–14
2.14.1.4 Using NETACP$BUFFER_LIMIT Logical Name to Override Default BYTLM Quota • 2–14
2.14.1.5 NML Checks for Illegal Address Configurations • 2–15
2.14.1.6 NCP/NML Requires OPER Privilege to Obtain Service Passwords • 2–15
2.14.1.7 NETACP/NETDRIVER—Synchronization Problem Fixed • 2–15
2.14.2 Executing CLUSTER_CONFIG.COM when Modifying VAXCluster Configurations 2–15
2.14.3 New Batch-Oriented Procedure for AUTOGEN Feedback 2–15
2.14.4 SYSMAN Utility—DECnet TASK Object 2–18
2.14.5 Running UETP 2–18
2.14.6 Error Messages During Installation 2–18

CHAPTER 3 PROGRAMMER RELEASE NOTES 3–1

3.1 MTH$SINCOS ROUTINE CORRECTION 3–1

3.2 MATH RUN-TIME LIBRARY NOTES 3–2

3.3 VMS DEBUGGER RESTRICTIONS 3–2
3.3.1 GO Command Limitation 3–2
3.3.2 Pascal Subrange Expressions Restriction 3–2
3.3.3 SET IMAGE Command Limitation 3–2
3.3.4 Requesting Help About Diagnostic Messages 3–2

3.3.5	**RUN/DETACHED Command Entered After LINK/DEBUG Command**	3–3
3.3.6	**Specifying Search List for SYS$LIBRARY**	3–3
3.3.7	**$WAKE Call Followed by $HIBER Call**	3–3
3.3.8	**Register Window in DECwindows Interface**	3–3
3.3.9	**Watchpoints in Installed Writable Shareable Images**	3–3
3.4	**SMALL COMPUTER SYSTEM INTERFACE (SCSI) RELEASE NOTES**	3–4
3.4.1	**Third-Party Device Support**	3–4
3.4.2	**SCSI Class Drivers and the DPT$V_NO_IDB_DISPATCH Bit**	3–4
3.5	**PPL$TRIGGER_EVENT ROUTINE MEMORY PROBLEM**	3–5
3.6	**USE OF TIMEDWAIT MACRO ON MICROVAX AND VAXSTATION 3100 SERIES SYSTEMS**	3–5
3.7	**APPENDING TO SHARED SEQUENTIAL FILES — RESTRICTION REMOVED**	3–5
3.8	**VMS DECWINDOWS RELEASE NOTES**	3–5
3.8.1	**Xlib Enhancements**	3–5
3.8.2	**DECwindows Server Notes**	3–6
3.8.2.1	Server Modifications • **3–6**	
3.8.2.2	Server Problems and Restrictions • **3–6**	
3.8.3	**SET DISPLAY DCL Command Default Setting**	3–7
3.8.4	**XUI Toolkit Changes and Restrictions**	3–7
3.8.4.1	Internal Format of Compound Strings • **3–7**	
3.8.4.2	Performance of INIT GET SEGMENT • **3–7**	
3.8.4.3	Font Unit Values • **3–7**	
3.8.4.4	List Box Dynamic Sizing • **3–7**	
3.8.4.5	FileSelection Callback Correction • **3–8**	
3.8.4.6	Selection Push Buttons • **3–8**	
3.8.4.7	Known Restrictions • **3–8**	
3.8.5	**User Interface Language (UIL) Compiler**	3–10
3.8.5.1	Specifying Multiline Compound Strings • **3–10**	
3.8.5.2	Arguments Not Directly Supported in UIL • **3–10**	
3.8.5.3	Width Argument for Work Area Menu • **3–11**	
3.8.5.4	Color Name Strings • **3–12**	
3.8.5.5	References to XmATTACH_NONE in the *VMS DECwindows User Interface Language Reference Manual* • **3–12**	
3.8.6	**Font Quality Cleanup**	3–12
3.8.7	**Terminal Font Changes**	3–12

CHAPTER 4 DOCUMENTATION RELEASE NOTES 4–1

4.1 VMS VERSION 5.3 DOCUMENTATION 4–1

4.2 CORRECTION TO THE *VMS VERSION 5.3 NEW FEATURES MANUAL* 4–2

4.3 CORRECTION TO THE *VMS VERSION 5.1 RELEASE NOTES* 4–2

4.4 CORRECTION TO THE *VMS VERSION 5.2 RELEASE NOTES* 4–2

4.5 CORRECTION TO THE *VMS VERSION 5.2 NEW FEATURES MANUAL* 4–2
4.5.1 Changes Made in NETCONFIG.COM for VMS Version 5.2 4–2
4.5.2 Default DECnet Access Options by Means of NETCONFIG.COM 4–3
4.5.3 Recommendation on the Use of FAL 4–3
4.5.4 Related Documentation 4–4

4.6 $MOUNT SYSTEM SERVICE ADDITION 4–4

4.7 LINKER IMAGE-ID FIELD CORRECTION 4–5

4.8 VAXSTATION 3100 MODEL INFORMATION 4–5

4.9 CORRECTIONS TO THE *VMS DCL DICTIONARY* 4–5
4.9.1 CONVERT/DOCUMENT Command 4–5
4.9.2 VIEW Command 4–11

4.10 ADDITIONS AND CORRECTIONS TO THE *VMS DEVELOPER'S GUIDE TO VMSINSTAL* 4–14
4.10.1 System Disk Directory Structure 4–14
4.10.2 Layered Product Installation 4–14
4.10.3 Command Line Qualifiers 4–15
4.10.4 Get Save Set Option 4–15
4.10.5 Release Notes Option 4–15
4.10.6 Alternate Root Option 4–15
4.10.7 Booting Option 4–15
4.10.8 Developer's Options 4–16
4.10.9 VMIMARKER.DAT 4–16

4.10.10	Referencing Other Products	4–16
4.10.11	Request Codes	4–16
4.10.12	SYSMGR Directory	4–16
4.10.13	KITINSTAL.COM	4–16
4.10.14	CHECKTRAN.COM	4–17
4.10.15	VMSINSTAL Steps	4–17
4.10.16	ADD_IDENTIFIER Callback	4–18
4.10.17	ASK Callback	4–18
4.10.18	CHECK_NETWORK Callback	4–18
4.10.19	CHECK_NET_UTILIZATION Callback	4–18
4.10.20	CHECK_PRODUCT_VERSION Callback	4–19
4.10.21	CHECK VMS VERSION Callback	4–19
4.10.22	COMPARE_IMAGE Callback	4–19
4.10.23	CONTROL_Y Callback	4–20
4.10.24	NETUAF.DAT	4–20
4.10.25	CREATE_DIRECTORY Callback	4–20
4.10.26	DELETE_FILE Callback	4–20
4.10.27	FIND_FILE Callback	4–20
4.10.28	GET_IMAGE Callback	4–20
4.10.29	GET_SYSTEM_PARAMETER Callback	4–20
4.10.30	MODIFY_STARTUP_DB Callback	4–21
4.10.31	MESSAGE Callback	4–21
4.10.32	PROVIDE_FILE Callback	4–21
4.10.33	PROVIDE_IMAGE Callback	4–21
4.10.34	SET IVP Option	4–22
4.10.35	SET POSTINSTALL Option	4–22
4.10.36	SET PURGE Option	4–22
4.10.37	SET REBOOT and SET SAFETY Options	4–22
4.10.38	SET STARTUP Option	4–23
4.10.39	Appendix B Corrections	4–23
4.10.40	Product Registration Information	4–23

APPENDIX A VMS DECWINDOWS PERFORMANCE CONSIDERATIONS A–1

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

INDEX

FIGURE

2–1 VMSD2 System Parameter 2–9

TABLES

1–1 Keywords No Longer Defined in EVE 1–3

2–1 License Unit Requirement Table (LURT) 2–6

2–2 VMS Versions that First Supported VAX Processors 2–8

Preface

This document describes software problems, corrections, and restrictions, as well as documentation changes, that pertain to Version 5.3 of the VMS operating system.

These release notes supersede the *VMS Version 5.2-1 Release Notes*, and include or update any Version 5.2-1 notes that are still pertinent to the Version 5.3 release. Please note also that all release notes in the *VMS Version 5.2 Release Notes* still apply, unless explicitly modified by notes in this document.

Note: The VMS Version 5.2-1 release provided hardware support for the VAXstation 3100 Model 38 and Model 48, and selected peripheral support for the MicroVAX and VAXserver 3100. VMS Version 5.2-1 was supplied only to those customers who purchased this new hardware.

For information about the new features included in VMS Version 5.3, see the *VMS Version 5.3 New Features Manual.*

Intended Audience

VMS Version 5.3 Release Notes is intended for all system users. Read the release notes before you use your VMS Version 5.3 system.

Document Structure

This manual contains the following chapters:

- Chapter 1 contains release notes intended for general users of the VMS operating system.
- Chapter 2 contains release notes intended for system managers.
- Chapter 3 contains release notes intended for application and system programmers.
- Chapter 4 contains additions and corrections to the VMS documentation set.
- Appendix A provides information on VMS DECwindows performance considerations.

Associated Documents

For additional information on VMS Version 5.3, see the following documents:

- *VMS Version 5.3 Upgrade and Installation Procedures*—Provides installation and upgrade information for VMS Version 5.3.
- *VMS Version 5.3 New Features Manual*—Provides information on new features included in VMS Version 5.3.

Conventions

The following conventions are used in this manual:

mouse	The term *mouse* 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: • 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 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.
italic text	Italic text represents information that can vary in system messages (for example, Internal error *number*).
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 radixes—binary, octal, or hexadecimal—are explicitly indicated.

1 General User Release Notes

This chapter contains information about the VMS Version 5.3 operating system that is of interest to general users.

1.1 Changes in Error Conditions for PRINT and SUBMIT Commands

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

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

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

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

1.2 SUBMIT/DELETE Command Change

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

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

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

1.3 Initializing Unformatted Diskettes Without VOLPRO Privilege

In previous versions of the VMS operating system, an unformatted diskette could not be initialized by the INITIALIZE/DENSITY command from an account that did not have the volume protection (VOLPRO) privilege. A fatal drive error would be reported when an account without VOLPRO privilege attempted to initialize the diskette.

Version 5.3 of the VMS operating system corrects this problem. VOLPRO privilege is no longer needed to initialize an unformatted diskette.

1.4 Changes to VAXTPU

This section contains release notes applicable to the VAX Text Processing Utility (VAXTPU).

1.4.1 Changed DECwindows VAXTPU Initialization

This section discusses new VAXTPU initialization processing features that are incompatible with previous versions of VMS DECwindows VAXTPU.

All code that compiled and executed properly on the versions of VAXTPU released with VMS Version 5.0, Version 5.1, and Version 5.2 will continue to compile and execute properly, unless the code depends on displaying information or prompting during startup of DECwindows VAXTPU. To display information or prompts, you must modify the initialization code to include a statement containing any of the following built-in procedures:

- READ_KEY built-in procedure
- READ_CHAR built-in procedure
- READ_LINE built-in procedure
- MAP built-in procedure

Also note that the use of new or incompatible Extensible VAX Editor (EVE) features might make rebuilding existing section files desirable.

For more information about new initialization programming procedures, see the *VMS Version 5.3 New Features Manual.*

1.4.2 Display Manager Definition Restriction

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

1.5 Changes to EVE

This section contains release notes applicable to the Extensible VAX Editor (EVE).

1.5.1 Obsolete Keywords for EVE Message Constants

The keywords listed in Table 1–1 are no longer defined in EVE.

Table 1–1 Keywords No Longer Defined in EVE

EVE Keyword Message Constant	Numeric Value
EVE$_NODOCMSG	35959266
EVE$_SHOWBUFSSTATUS	35971419
EVE$_SHOWSTATUS	35971403
EVE$_HELPSTATUSDOWN	35971363
EVE$_HELPSTATUSUPDN	35971371
EVE$_HELPSTATUSN	35971339
EVE$_STATUSLINE	35971291
EVE$_PATSTATBIG	35971523
EVE$_DCLSTATUS	35971323
EVE$_HELPSTATUSNP	35971347
EVE$_HELPSTATUSUP	35971379
EVE$_HELPSTATUSP	35971355
EVE$_HELPSTATUS	35971331
EVE$_PATSTATBIGUPDN	35971507

These keywords were message constants used as parameters for the EVE$MESSAGE procedure. When writing your own procedures, do not use any such EVE message constants because these constants are not supported as part of the EVE public interface. The only exception to such use is any message constant used by EVE as part of a call to an EVE$ routine or as a return status from an EVE$ routine. Digital will ensure that such constants remain totally upward compatible.

The integer values of the message constants shown in Table 1–1 are listed so that any code you have written that depends on these messages can continue to use the messages for this release. Use these integers in place of the constants that previously defined the messages.

1.5.2 Known Restrictions for EVE

The following restrictions and limitations are known to exist in this version of EVE.

- If you have a primary EVE selection, move to another buffer, and copy the selection to another DECwindows application by doing a MoveTo operation (Ctrl/Click MB3); the selection is correctly copied to the other application. However, the selection is not pending deletion from EVE as expected. EVE outputs an error message stating the selection is in another buffer.
- If the variable EVE$X_DEBUG is set to 1, entering a WILDCARD FIND command of "12\>\<ab" ("12" on the end of a line followed by "ab" on the start of next line) outputs an error message that can be ignored.
- The EVE command SELECT ALL changes the cursor position to the end of the buffer if in the forward direction, or to the beginning of the buffer if in the reverse direction. It should not move the cursor.
- Clicking a mouse button in a buffer, and then dragging the mouse into another buffer positions the cursor in the other buffer. This procedure moves the current position in those buffers from where it had been. Also, merely moving the mouse into another buffer does not reposition the mouse in the new buffer. You must also click a mouse button to reposition the mouse.
- Dragging the mouse through the command window incorrectly displays the command window prompt. Only clicking the mouse in the command window should display the prompt.
- The first attempt to define a Gold key combination fails. For example, with the DIGITAL Standard Editor (EDT) keypad set, enter the DEFINE KEY command. When prompted for a command, enter the command TOP. When prompted for the key to define, press Gold/KP7. EVE outputs an error message stating there is an undefined procedure called GOLD. Subsequent attempts to define the key succeed.
- When you resize the DECwindows EVE display to less than 10 screen lines below the menu bar, EVE displays the message buffer in the available line or lines, and also pops up a message box that you must acknowledge. This is expected behavior. However, a subsequent attempt to resize the display to 10 or more lines may not work. When this happens, first resize the display to a number smaller than 10 lines but greater than one line, and then resize it again to 10 or more lines.

Defining Mouse Button Clicks

EVE lets you define mouse button clicks with the DEFINE KEY= command. However, EVE does not always properly respond to those mouse button clicks. Instead, EVE executes the command bound to the corresponding mouse UP button if one exists.

In DECwindows, VAXTPU converts CLICK buttons to mouse UP buttons under certain circumstances. If you define an UP mouse button but not the corresponding CLICK mouse button in the same key_map, VAXTPU passes an UP button to EVE each time you input a CLICK. VAXTPU does this for compatibility with older section files written before DECwindows support (and mouse CLICK buttons) were introduced.

EVE filters user mouse actions by defining programs in a key_map set on each VAXTPU window. EVE looks up programs in the user key_map that correspond to buttons intercepted by its own window key_map. EVE often incorrectly looks up the user program for an UP mouse button when you have actually input a CLICK.

For example, assume the following button definitions:

```
DEFINE_KEY ("EVE_RESET", M2DOWN, "RESET")
DEFINE_KEY ("EVE_BOTTOM", M2UP, "BOTTOM")
DEFINE_KEY ("EVE_TOP", M2CLICK, "TOP")
```

These definitions are put into EVE's buffer key_map EVE$X_USER_KEYS. Assume you then press and release M2 within the same character. This should execute the M2DOWN and then the M2CLICK definitions. However, since EVE has no definition for M2CLICK in the window key_map (EVE$X_MOUSE_KEYS), VAXTPU executes the definition for the user M2UP button, and not the user definition for M2CLICK.

This problem will be fixed in a future version of VAXTPU. At that point, defining CLICK buttons will work properly. For now, if you want to define a CLICK button in the user key_map EVE$X_USER_KEYS, then also define the corresponding CLICK button in the EVE$X_MOUSE_KEYS key_map on the EVE command line. The following example shows you how to define M2CLICK:

```
TPU DEFINE_KEY(EVE$$KT_RETURN+"EVE$UNDEFINED_MOUSE_KEY(LAST_KEY)",
               M2CLICK,"M2CLICK",EVE$X_MOUSE_KEYS);
```

The following example shows you how to define ALT/M2CLICK:

```
TPU DEFINE_KEY(EVE$$KT_RETURN+"EVE$UNDEFINED_MOUSE_KEY(LAST_KEY)",
               KEY_NAME(M2CLICK, ALT_MODIFIED),"ALT-M2CLICK",
               EVE$X_MOUSE_KEYS);
```

These commands should be entered as a single line. Enter the commands first into a buffer, then cut the entire line, press the Do key, paste the entire line, and press the Return key to finish the command.

1.6 BATCH/PRINT Facility Performance Improvements

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

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

- Issue many single-file SUBMIT and PRINT command requests.
- Use the SPAWN command to create subprocesses that require job controller action on termination.
- Have a high number of interactive or detached process, or both, terminations.
- Use the /NOTIFY option on SUBMIT and PRINT commands.

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

1.7 VMS DECwindows Release Notes

The following sections describe release notes applicable to VMS DECwindows.

1.7.1 Window Manager (Icon Box) Corrections

The following corrections have been made to the Icon Box:

- The Icon Box is now centered on the screen, based upon 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.

1.7.2 FileView Changes

The following release notes are of general interest to FileView users.

- FileView's user interface definition has been converted to User Interface Language (UIL). Customization of FileView through XDefaults (DAT) files remains unsupported. (Previously, FileView used an XDefaults resource file for its user interface.) Because of this change, attempts to change fonts and the use of DAT files will no longer be as effective. The name of FileView's application class was changed from VUE$DEFAULTS.DAT to VUE$MASTER.DAT. These changes may affect users who are doing advanced workstation customization.

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

Saving a View

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

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

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

Customizing FileView

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

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

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

1 Open a window that contains the DCL prompt ($).

2 If necessary, direct your display to your workstation with the DCL command SET DISPLAY.

3 Enter the DCL command DEFINE/JOB VUE$PROFILE NL:.

4 Enter the DCL command RUN SYS$SYSTEM:VUE$MASTER.

A system-default FileView window will be displayed on your workstation.

5 Choose **Logical Names...** from the **Utilities** menu.

6 In the Logical Names dialog box, deassign VUE$PROFILE.

7 Choose **Save Startup View** from the **Customize** menu.

8 Choose **Quit** from the **Control** menu.

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

1.7.3 DECterm Corrections and Notes

This section contains release notes for the DECterm terminal emulator.

1.7.3.1 Corrected DECterm Problems

The following DECterm problems have been fixed:

- The null terminated string problem and the $CANTIM problem have been fixed in DECwTermPort.
- DECterm now redraws text correctly when it is scrolling in a partially obscured window.

- The problem that made TBC (tabulation clear) set the conformance level to zero has been fixed.
- This version fixes the problem where part of the transcript scrolls into the window when you select SET TERM/WIDTH=132.
- Resetting the terminal now enables the text cursor.
- DECterm no longer terminates abnormally when you resize it to zero lines (the minimum terminal size is now 1 line).
- Sixels do not overwrite the border, are drawn in the correct colors, and scroll when they reach the bottom of the scrolling region.
- The pixmap backing store is deallocated when the graphics picture scrolls off the screen.
- Many improvements have been made in ReGIS and ANSI locator mode. ReGIS character sizes are now more compatible with the VT330 and VT340 terminals, and there have been several other problem fixes. Many of the remaining differences between DECterm ReGIS and VT330/VT340 ReGIS are due to the different bitmap size on DECterm, or, in the case of single-plane systems, due to the lack of color or gray shades.
- DECterm does not try to write to the status line unless the status line is enabled.
- DECterm now conforms to the Inter-Client Communication specification, and supports TARGETS, STRINGS, and PRIMARY as selection targets.
- A problem where fonts were being loaded but never freed has been fixed.
- Colormap support for true color systems (such as the VAXstation 3520 /3540) has been fixed.
- The icon pixmap problem on screens other than zero (for example, the second screen of a multiheaded workstation) has been fixed.

1.7.3.2 Initializing DECterm

The following information is specific to DECterm initialization:

- To avoid having your DECterm windows shrink to the default size of 80 characters by 24 lines unexpectedly, Digital suggests that system-wide and user login command procedures (SYS$SYLOGIN.COM and LOGIN.COM) not execute the SET TERMINAL/INQUIRE command on DECterm windows.

  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 on both DECterm windows and non-DECterm windows, use the following commands in a system-wide or user login command procedure:

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

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

1.7.3.3 Keyboards and Languages Information

The following information is specific to DECterm keyboards and languages:

- To create a compose character in DECwindows, you must press the Compose key and space bar at the same time, not the Compose key alone as on character cell keyboards. The Compose key is a modifier and is used like a Shift key.
- Keyclick, auto-repeat, keyboard dialects (national keyboards), keyboard usage mode (data processing/typewriter mode), and caps lock/shift lock should be changed in the Session Manager on a workstationwide basis, rather than through DECterm.
- National replacement character sets (NRCS) are selected in the Customize / 7-bit NRCS Selection menu in DECterm. Their selection is independent of both the keyboard dialect and the keyboard usage mode. For example, you must change both the Session Manager and DECterm to use the French NRCS with the French keyboard.
- The Dutch NRCS is implemented, even though its use is no longer recommended. It is likely to be removed from future versions of DECterm. The florin character is drawn as a currency sign, ¤ (character 164 in ISO Latin 1 or 168 in DEC Multinational Character Set).
- When the keyboard becomes locked (for example, because the input silo is full), DECterm rings the bell for each character entered until the lock condition is cleared, rather than disabling keyclick as is done on the VT320, VT330, and VT340 terminals.
- In some situations DECterm can continue to auto-repeat a key even after the key has been released; this is a problem with WPS-PLUS under ALL-IN-1. To avoid this problem, activate the Lock User Features toggle button in the Customize General dialog box.

1.7.3.4 Fonts and User Interface Information

The following information is specific to DECterm fonts and user interface:

- The Big font/Little font radio buttons in the Customize Window have no effect when using the 100-dot-per-inch (dpi) fonts. This occurs because the 75-dpi big font is used for both the big and little fonts on 100-dpi monitors.
- The Dismiss button in DECterm's dialog boxes is equivalent to the Cancel function in other DECwindows applications.

1.7.3.5 Text Information

The following information is specific to DECterm text:

- User-loadable characters (DRCS), print screen, and local mode and control representation mode are not implemented.
- RIS (an <ESC> c sequence) sets certain parameters to their factory defaults instead of to their saved settings. Therefore, its use is discouraged.
- The checkerboard character (character 97 in the special graphics set) is used as an error character instead of as the reverse question mark, and, in some cases, it is not displayed at all.

1.7.3.6 Graphics Information

The following information is specific to DECterm graphics:

- The dialog box will appear black on a 4-plane color system if you bring up a dialog box in the Customize window while DECterm is using its private color map. To avoid this problem, either bring up all the dialog boxes you plan to use before displaying any graphics or exit from the graphics, clear, and reset the window (this restores the default colormap) before bringing up a dialog box.
- There are various problems when displaying ReGIS or sixel graphics while recording lines off the top of the screen. In some cases, recorded lines can overprint each other when you use the vertical scroll bar, and ReGIS pictures can be scaled to the wrong size. To avoid these problems, turn off the "Record lines off top" button in the Customize Window menu when displaying ReGIS or sixel pictures.
- Only graphics, not text, are written to the graphics backing store. When part of the window has to be redrawn, DECterm will draw the graphics portion first and then overlay it with text. As a result, the window might not look the same when redrawn as it did when it was occluded.
- ReGIS standard character sizes are different from those in the VT340. The VT340 uses a character size of 8 pixels by 20 pixels and a display size of 9 pixels by 20 pixels, while the ReGIS character size in DECterm depends on the font being used. This means text labels are not aligned correctly in applications such as DECgraph and DECslide. The standard character sizes are likely to change in future versions of DECterm in an attempt to correct this problem.

- ReGIS addresses the entire window, unless the window height is exactly 24 rows. This causes pictures to be scaled differently on terminals other than the VT330 or the VT340.
- ReGIS pictures may be scaled incorrectly when the window is resized, or when the font size is changed.
- Terminate one-shot input mode (R(I0)) from the ReGIS locator reporting by clicking a mouse button. Pressing one or more keys on the keyboard will send information to the application, but it will not be followed by ReGIS coordinates. You cannot control the locator position with the arrow keys, you must use a mouse (unless you use Ctrl/F3 to do this for the entire workstation). In addition, this version of DECterm does not support DECLKB (called DECLBD in the *VT330/VT340 Programmer Reference Manual*), so clicking a mouse button sends an escape sequence followed by the coordinates of the mouse position.

  Application writers modifying their applications to support DECterm should consider using the text locator mode.
- The following are not implemented: ReGIS command display mode, scrolling, hard copy, and output cursors. The mouse is used as the input cursor, but its shape cannot be changed with the S(C(I)) command.
- Both monochrome and color systems emulate a 16-entry colormap, as on the VT340, rather than a 4-entry colormap as on a VT330 (or VT240).
- ReGIS always uses the DEC Multinational Character Set, regardless of whether ISO Latin 1 was chosen in the Customize/General menu.
- There is no Customize option to disable the reporting of macrograph contents, as on the VT330/VT340. Proper care should be taken if sensitive information is stored in macrographs.
- Sixel files can overwrite the window borders. To restore the borders, shrink the window to an icon and then expand the icon to a window. There is also a problem on single-plane systems (such as the VS2000) where sixel pictures are redrawn with reversed colors when they are refreshed.
- Sixels do not scroll when they reach the bottom of the window, so it is best to position the cursor at the top of the window before displaying a sixel file.
- The vertical position after leaving sixel mode is the same as it was on entry, and does not account for the length of the sixel picture.
- The background parameter in the sixel device control string (DCS) has no effect; lines actually overwritten by the picture are erased, but the rest of the window is unaffected, regardless of the setting of the parameter.
- The horizontal and vertical extent parameters in the set attributes command (", ASCII code 34) are not implemented.

1.7.3.7 DECterm PC Interoperability Restrictions

The following interoperability restrictions apply when using DECterm on a PC system:

- The DECterm window is not sized properly on the PC screen. Initially, the DECterm window is located partially off the screen, and you must manually center it. You must also manually resize it, as 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, go into the Window dialog box from the Customize menu, select "Condensed font (132 columns)" and then select OK. To save the change, select "Save Current Settings" in the Customize menu. DECterm then 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 system.

  To circumvent this problem, select the Delete setting of this toggle button.

- The "Comma Key Sends ,<" option does not work on the PC. Pressing the comma key produces a comma, but pressing Shift/, produces the number nine (9).

  To circumvent this problem, select the "Comma Key Sends ,," setting of this toggle button.

- The "Tilde Key Sends ESC" option under the Customize Keyboard dialog box does not work on the PC server. When this option is enabled, pressing the tilde (~) key gives no response, whereas it should act as an escape key (and does on the native platform).

  To circumvent this problem, select the "Tilde Key Sends '~" setting of this toggle button.

- DECterm does not support function, editing, and keypad keys on non-LK201 keyboards. This is a problem from PC- and MAC-based X servers. These servers do not generate all the KEYSYMs the LK201 supports; therefore, not all DECterm functions are accessible. For example, you cannot run EDT in screen mode from DECterm given a non-LK201 keyboard.

  Users with keyboards that do not include some of the LK201 keys will not be able to run programs that depend on those keys easily. There are often functional equivalents for the keys, though; for example, EDT has a line editing mode in which commands can be entered in alphanumeric form rather than with 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, where <ESC> is an escape, which can be entered as Ctrl/3 or Ctrl/[. The escape sequences sent for each key are listed in the *DECterm Text Programming Manual.*

1.7.3.8 DECterm Memory Problem

When you log out of a DECterm window, about 64 pages (32 kilobytes) of virtual memory are not freed. This means that if you create and delete many DECterm windows, the DECterm controller process may run out of virtual memory. If the controller runs out of virtual memory it will crash, and you will lose all of your DECterm windows. To prevent this you should periodically exit from all your DECterm windows, so that a new controller will be created the next time you create a DECterm window.

1.7.4 Help Widget Enhancement

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

1.7.5 Desktop Application Notes

The following sections provide notes about the DECwindows desktop applications.

1.7.5.1 Calendar Restrictions and Limitations

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

If your cluster 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 will need to run Calendar remotely if the version on your workstation does not match your Calendar database version. Instructions for running DECwindows applications remotely can be found in the *VMS DECwindows User's Guide*.

1.7.5.2 Mail Corrections and Restrictions

This section provides information on DECwindows Mail.

The following problems have been fixed:

- Delete Drawers... would not delete the actual mail file. This problem has been fixed.
- When you used a dialog box to move or copy messages selected in the main window, making a selection would cause the main window selection to be lost. The main window selection is no longer lost.
- Using Pick From Selected Folder in the paned interface on the deliver folder prevented any future delivering or reading of new mail. Now, future delivering or reading of mail is not affected.
- Entering an illegal personal name in the Create-Send window no longer makes that window unusable.

- The icon color for both small and large icons is now changed when new mail arrives.
- The pane widget (used in the read window and in the paned main window) now complies with the *XUI Style Guide*. The knob that was previously grabbed to adjust the panes is gone, and the panes are now separated by a double line. To adjust a pane, move the cursor over the double line (it should change to the double-arrow pane cursor), press and hold MB1, and drag the mouse to the desired position.

The following restrictions exist:

- When you read a Mail message while the Auto Refile option is set, the message is moved immediately to the MAIL folder. If you delete or move the message, the action does not take place until the deliver folder is closed. Therefore, if you read a message, delete it, and open the MAIL folder without first closing the deliver folder, you will see the deleted message listed in the MAIL folder. When you eventually close the deliver folder, the deleted message will be deleted from the MAIL folder.
- Input focus is not always assigned to the Create-Send window when it is mapped.
- On the In Window submenu of the main window Read menu, menu entries of the form "Window #" will be enabled improperly even if no messages are selected. Choosing one of these entries with no messages selected may cause an access violation.
- The Options... button in the Print... dialog box may intermittently cause an access violation in certain circumstances. To avoid this possibility, change your print settings by selecting Customize Print Attributes... before you attempt to print.

1.7.5.3 Paint Restrictions

Due to limits on the size of pixmaps, Paint cannot read pictures in X11 bitmap format that are larger than the size of the screen in pixels. Paint also cannot write pictures larger than screen size into an X11 bitmap file. To save large pictures, use the DDIF format.

If the picture is smaller than the paint window (that is, the borders of the picture are visible), and you attempt to use the paint bucket tool on the right or bottom border of the picture, an access violation results. To avoid this problem, do not attempt to use the paint bucket tool on the right or bottom border of the picture.

1.7.5.4 PrintScreen Restrictions

The following restrictions apply to PrintScreen:

- 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 circumvent this problem, choose the No Rotate option.
- On small memory systems, print screens of large areas may cause memory failures.

- Performance is slower than in previous versions for color systems, especially when the output is gray scale or black and white.
- PrintScreen on VAXstation 3520/3540 systems only works if one colormap is installed on the system. If any software that installs a second colormap is running, PrintScreen returns an error.

1.8 VMS Version 5.2-1 Release Notes

This section contains general user release notes that originated with VMS Version 5.2-1.

1.8.1 Console Carrier Error Handling

The console interface is invoked when an NCP CONNECT NODE command is issued; only one user is allowed to use this facility at any given time. In previous versions of VMS, when a second user attempted to use the console carrier facility on a given system, the second user would receive the following error message:

```
Bad parameter value
```

This message indicated that the console carrier protocol type was not available. As of VMS Version 5.2-1, the second attempt to use console carrier produces the following error message:

```
Console carrier protocol not available - already in use
```

1.8.2 SYSGEN INVVEC Error Fixed

Certain hardware configurations have had a problem with the following command in VMS Version 5.2:

```
SYSGEN>
```

The configuration of the first floating vector device would report incorrectly the following error message:

```
%SYSGEN-E-INVVEC, invalid or unspecified interrupt vector
```

As of VMS Version 5.2-1, the configuration of the first floating vector device no longer reports this error message.

1.8.3 RMS Journaling—Automatic Recovery Performance

VMS Version 5.2-1 corrects some RMS Journaling problems with automatic recovery after a failure. In some cases, this correction significantly reduces the time required for recovery. No changes in programs are required.

Automatic recovery is required after a system crash, or a process deletion ($DELPRC rather than normal process rundown). The recovery may occur immediately when other shared users of recovery unit journaled files exit after the failure, or it may be delayed until the next time the file is accessed. The changes made in Version 5.2-1 reduce the time for each

individual file recovery, and they also reduce total recovery time when several failure events occur within a short time.

1.8.4 Restriction on Number of Files in Magnetic Tape Volume Set

Because the ordinal number of a file on an ANSI-labeled magnetic tape volume set is stored in the file header as four decimal digits, an ANSI-labeled volume set is limited to 9999 files.

In previous versions of VMS, no check against this limit was made when creating a new file. The ten thousandth file and subsequent files would be created with an invalid file header.

In VMS Version 5.2-1, an attempt to create more than the supported number of files fails and the following error message is reported:

```
%SYSTEM-W-FILENUMCHK, file identification number check
```

This error message is expected to change in a future release of the VMS operating system.

2 System Manager Release Notes

This chapter contains information about the VMS Version 5.3 operating system that is of interest to system managers.

2.1 Reduction in Cluster Reconfiguration Time with Quorum Disk

Quorum disk processing has been enhanced to ensure that quorum disk votes continue to be counted toward quorum whenever VAXcluster software determines that a removed node has shut down or has performed a bugcheck.

Prior to VMS Version 5.3, when a node was removed from a VAXcluster, by any means, the votes contributed by the quorum disk were not included in quorum calculations performed by the remaining nodes, for 4 times the value of QDSKINTERVAL seconds. Thus, a VAXcluster could lose quorum for this period if the quorum disk votes were necessary to meet quorum.

In VMS Version 5.3, when a node is removed from a VAXcluster, either by means of a shutdown or a bugcheck, the votes contributed by the quorum disk are included without delay in quorum calculations performed by the remaining nodes. However, if a node leaves the VAXcluster by any other means, such as a console halt operation or a powerfail, the pre-Version 5.3 behavior is maintained. As a result, State Transition delays associated with a quorum disk no longer occur when a node performs a bugcheck or is shut down.

Note that the new mechanism is effective only if the remaining nodes can determine that the removed node has shut down or has performed a bugcheck. This condition is indicated by receipt of a "last gasp" datagram from the failed node. However, because the datagram communications mechanism used is a best effort only, the datagram might fail to arrive. For these very rare occasions, the pre-Version 5.3 behavior is maintained.

System managers who have previously lowered the SYSGEN parameter QDSKINTERVAL may wish to restore the default value (10 seconds), thereby reducing quorum disk I/O.

2.2 Ethernet Circuit Loopback Tests

In VMS Versions 5.0 to 5.2, Ethernet circuit loopback tests would fail if the node address, rather than the node name, was given in the command specification. For example:

```
$
%NCP-W-PRMMS, Parameter missing, Physical address
```

This problem has been corrected in VMS Version 5.3.

Also, the *VMS Network Control Program Manual* indicates that all parameters for circuit loopback tests are optional. In fact, the Ethernet multicast loopback assistance (as in NCP LOOP CIRCUIT QNA-0) is an unimplemented feature of the DIGITAL Network Architecture (DNA) architecture. Therefore, the Network Control Program (NCP) online HELP information has been updated to reflect that Ethernet circuit loopback operations require at least one parameter to define the loopback destination.

2.3 XMDRIVER Powerfail Recovery

XMDRIVER powerfail recovery did not always work properly. The DMR11 device would sometimes hang in an ON-SYNCHRONIZING state, which required a reboot operation to clear. VMS Version 5.3 corrects this problem.

2.4 Page Fault within SYSLOA Problem Fixed

In previous versions of VMS, a PGFIPLHI (page fault with IPL too high) crash would occasionally occur when a program counter (PC) was within an area of memory allocated to SYSLOA/MOUNTVER. VMS Version 5.3 corrects this problem.

2.5 Device Naming Change for DSSI Disks Connected to a KFQSA Controller

Starting with VMS Version 5.3, the device names assigned to Digital Storage System Interconnect (DSSI) disks attached to a KFQSA controller have changed. In previous versions of VMS, the device name was of the form DI*cu*, 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 of the form $*n*$DI*cu*. This scheme was inconsistent with the naming used for DSSI disks attached to an integrated (SII-based) adapter, such as that 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 representing the disk. Instead, all DSSI disks use the controller letter A. Thus, device names are now of the form $*n*$DIA*u*, where *n* is the nonzero allocation class of the DSSI disk, or *nodename*$DIA*u* if the allocation class is zero. Note that nodename is the node name of the DSSI disk, and not the same as the VMS parameter SCSNODE.

In order to alleviate some of the problems anticipated with this type of a change, the new naming scheme is dependent on the new SYSGEN parameter VMS5. With VMS5 set to 1, the old (prior to Version 5.3) device naming will continue to be used, while setting VMS5 to zero will enable the new behavior. Systems installing Version 5.3 for the first time will have VMS5 set to zero by default, while systems being upgraded from a previous version of VMS will have VMS5 set to 1.

2.5 Device Naming Change for DSSI Disks Connected to a KFQSA Controller

For example, a single KFQSA with three DSSI disk drives attached would have the following device names for ports/disks:

	Allocation Class=0		Allocation Class=4	
	Port	**Disk**	**Port**	**Disk**
Old scheme:	PUA0	DIA0	PUA0	4DIA0
(VMS5=1)	PUB0	DIB1	PUB0	4DIB1
	PUC0	DIC2	PUC0	4DIC2
New scheme:	PUA0	FRED$DIA0	PUA0	4DIA0
(VMS5=0)	PUB0	BARNEY$DIA1	PUB0	4DIA1
	PUC0	WILMA$DIA2	PUC0	4DIA2

Note: In a future release of VMS, the SYSGEN parameter VMS5 will no longer be used to determine the device naming scheme; at that time, only the new naming scheme will be used. It is recommended that VMS5 be set to zero as soon as practical after upgrading to Version 5.3.

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 KFQSAs per system could result in inconsistent device names across the two systems if the common DSSI was incorrectly attached (KFQSA #1 on MicroVAX A is attached to KFQSA #2 on MicroVAX B). The old scheme also precluded dual-hosting with mixed adapter types (Integral SII and KFQSA).

With the new scheme, all systems (regardless of adapter type) use device names $*n*$DIA*u* or *nodename*$DIA*u*, where the only variables are the allocation class, node name and unit number of the DSSI disk. As 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 KFQSAs per system and mixed adapter types.

Note: These configurations are only supported in Version 5.3 with the VMS5 parameter set to zero. Dual-host configurations with multiple KFQSAs per system or with mixed adapter types are not supported with VMS5=1.

However, all DSSI disks must have unique device names. Therefore, it is recommended 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 SET/SHOW commands to examine and change the values of device parameters. Then enter the WRITE command to write any new parameter values to nonvolatile storage in the device. (Changing ALLCLASS or NODENAME requires that the controller be initialized.)

For more information on the console command SET HOST/DUP, see the section "Configuring RF30 and RF71 Devices in a VAXcluster" in the *VMS Installation and Operations: MicroVAX, VAXstation, and VAXserver 3400, 3600, 3900 Series*, or the hardware information for your system.

2.6 Modifying License Units with the License Management Facility

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

For VMS Version 5.3, many products require a Product Authorization Key (PAK) that includes license data to be registered in the LICENSE database. Some PAKs provide a MOD_UNITS option, which lets you modify the size of the registered licenses. If you have registered a license with the MOD_UNITS option, you can modify the size of the license to match the product to your VAX computer or VAXcluster environment, subject to your software license terms and conditions. You can modify all licenses with the MOD_UNITS option, including those that specify the following:

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

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

```
LICENSE LIST/FULL FORTRAN
```

The following display of a license list appears:

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


 License Management Facility

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

 ----------------------------------------------------
 Issuer:                      DEC
 Authorization:               USA-10
 Product Name:                FORTRAN
 Producer:                    DEC
 Units:                       900
 Version:                     5.3
 Date:                        (none)
 Termination Date:            21-DEC-1991
 Availability:                F (Layered Products)
 Activity:                    0
 Options:                     MOD_UNITS
 Hardware ID:

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

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

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

1. Enter the LICENSE LIST/FULL *product-name* command to display the license. Examine the display. Look for a code A through F, next to either the Availability label or the Activity label in the LICENSE LIST display. The codes, which designate license type, determine the appropriate license size for your VAX computer.
2. Find the name of your VAX computer in the first column of the VMS Version 5.3 License Unit Requirement Table (LURT) (see Table 2–1).

Table 2–1 License Unit Requirement Table (LURT)

	License Types by Code					
	VMS				SIP	LP
System Marketing Model	A	B	C	D	E	F
VAX 11/730	10	NA	NA	NA	230	50
VAX 11/750	12	NA	NA	NA	230	100
VAX 11/780,785	13	NA	NA	NA	230	100
VAX 6000-210,6000-310	58	NA	NA	NA	230	300
VAX 6000-220,6000-320,6000-410	69	NA	NA	NA	230	600
VAX 6000-230,6000-330	81	NA	NA	NA	400	900
VAX 6000-240,6000-340,6000-350,6000-420	93	NA	NA	NA	400	1200
VAX 8200,8250	20	NA	NA	NA	230	100
VAX 8300,8350	25	NA	NA	NA	230	200
VAX 8530	65	NA	NA	NA	230	400
VAX 8550,8700,8810	72	NA	NA	NA	400	600
VAX 8600,8650	28	NA	NA	NA	230	400
VAX 8800,8820	93	NA	NA	NA	400	1200
VAX 8830,6000-360,6000-430	119	NA	NA	NA	600	1800
VAX 8840,6000-440,6000-450,6000-460	143	NA	NA	NA	600	2400
MicroVAX II	18	NA	100	NA	230	50
MicroVAX 2000	18	NA	100	NA	230	20
MicroVAX 3100	18	NA	100	NA	230	20
MicroVAX 3500,3600,3800, 3900	60	NA	100	NA	230	300
MicroVAX 3300,3400	60	NA	100	NA	230	100
VAXstation II,II/GPX	NA	NA	NA	100	50	10
VAXstation 2000,2000/GPX	NA	NA	NA	100	50	10
VAXstation 3100,3200,3500, 3520,3540,8000	NA	NA	NA	100	50	10

Key to License Type Codes

A–VMS capacity
B–VMS server
C–VMS concurrent user
D–VMS workstations
E–System integrated products
F–Layered products

(continued on next page)

Table 2–1 (Cont.) License Unit Requirement Table (LURT)

	License Types by Code					
	VMS				SIP	LP
System Marketing Model	A	B	C	D	E	F
VAXserver 2000	NA	52	NA	NA	50	10
VAXserver 3100,3300,3400, 3500,3600,3602,3800,3900	NA	100	NA	NA	50	10
VAXserver 6000-210,6000-310	NA	1443	NA	NA	230	200
VAXserver 6000-220,6000-320,6000-410,6000-420	NA	1737	NA	NA	230	400

Key to License Type Codes

A–VMS capacity
B–VMS server
C–VMS concurrent user
D–VMS workstations
E–System integrated products
F–Layered products

3 Find the row with the appropriate name and the column with the code corresponding to the type of license you have. Note the value at the intersection of the row and column. For example, find the intersection of VAX 8800 with column F (the value shown is 1200). Unless NA (meaning a value is not applicable for this license type) appears, the value that you find is the number of units required for your VAX computer and type of license.

4 Modify your license by entering the value found in Table 2–1 as the parameter in a LICENSE MODIFY/UNITS=*number* command. For example, to modify a FORTRAN license running on a VAX 8800, enter the following:

```
$
```

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

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

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

The license unit requirements provided by the License Unit Requirement Table (LURT) are subject to change.

2.7 VMS Support for VAX Processors

Table 2–2 lists the versions of the VMS operating system that first supported available VAX processors.

Table 2–2 VMS Versions that First Supported VAX Processors

VMS Version	Processors Supported
Version 5.0	VAX 6000 Series 2XX, VAX 8820-8840
Version 5.0-2A	MicroVAX 3300/3400
Version 5.1	VAX 6000 Series 3XX, MicroVAX 3800/3900
Version 5.1-B	VAXstation 3100, Model 30/40
Version 5.1-1	VAXstation 3520/3540
Version 5.2	VAXstation 6000 Series 4XX, MicroVAX 3100
Version 5.2-1	VAXstation 3100, Model 38/48

2.8 VAXstation 3100 Model Name Change

Prior to VMS Version 5.3, the lexical function F$GETSYI using the "hw_name" item code would return the following string for the VAXstation 3100 Model 38 and 48:

```
$
VAXstation 3150
```

In VMS Version 5.3, this function now returns the following string:

```
$
VAXstation 3100
```

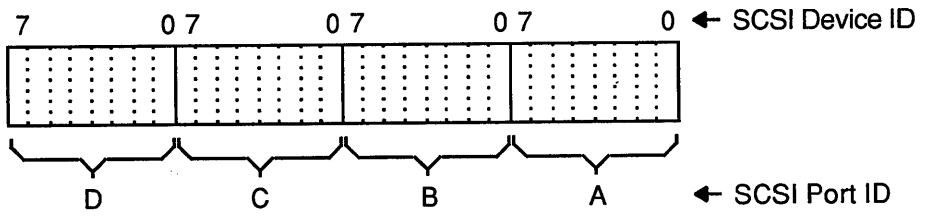
2.9 VMSD2 SYSGEN Parameter and SCSI Devices

VMS Version 5.3 temporarily defines the special SYSGEN parameter VMSD2 to be of use to MicroVAX/VAXstation configurations that include third-party Small Computer System Interface (SCSI) devices. (A complete discussion of this topic appears in the *VMS Version 5.3 Small Computer System Interface (SCSI) Device Support Manual.*)

SYSGEN's autoconfiguration facility automatically loads the VMS SCSI disk or tape class driver for a device on the SCSI bus that identifies itself as either a random-access or sequential-access device. If this SCSI device is to be supported instead by the VMS generic SCSI class driver or by a third-party SCSI class driver, the automatic loading of a VMS SCSI class driver for the device must be disabled.

The VMSD2 parameter, as shown in Figure 2–1, allows a configuration including a SCSI third-party device to prevent the loading of a VMS disk or tape SCSI class driver for any given device ID.

Figure 2–1 VMSD2 System Parameter



The VMSD2 system parameter stores a bit mask of 32 bits, where the low-order byte corresponds to the first SCSI bus (PKA0), the second byte corresponds to the second SCSI bus (PKB0), and so on. For each SCSI bus, setting the low-order bit inhibits automatic configuration of the device with SCSI device ID 0; setting the second low-order bit inhibits automatic configuration of the device with SCSI device ID 1, and so forth. For instance, the value 00002000_{16} would prevent the device with SCSI ID 5 on the bus identified by SCSI port ID *B* from being configured. By default, all of the bits in the mask are cleared, allowing all devices to be configured.

Note: A future release of VMS will provide a different mechanism for preventing the configuration of a VMS SCSI class driver for a given device ID. At that time, the VMSD2 SYSGEN parameter will revert to its status of a special parameter reserved to Digital.

The field test of VMS Version 5.3 used the temporary parameter VMSD1 for this purpose. If you used VMSD1 during field test, you will need to change to the VMSD2 parameter as described in this section.

2.10 VMS/RZ55 Data Integrity Problem Fixed

A possible data integrity problem with VMS when used with the RZ55 or third-party SCSI disk drives has been fixed. This problem existed on the following VMS versions:

- Version 5.1-B
- Version 5.1-1
- Version 5.2
- Desktop-VMS Version 1.0

2.10 VMS/RZ55 Data Integrity Problem Fixed

The problem has been corrected with VMS Version 5.3, as well as with VMS Version 5.2-1, which was a limited software release shipped only to customers who purchased the following new hardware:

VAXstation 3100 Model 38 and 48
MicroVAX 3100
VAXserver 3100

If your system has an RZ55, you should install VMS Version 5.3 as soon as possible. This problem does not affect other disks, such as the RZ22 or the RZ23.

The potential for data loss was discovered during laboratory testing, and only occurs under infrequent circumstances. To determine whether your RZ55 disk has been affected by the problem while running earlier versions of VMS, enter a DCL command in the following form, where the device-name parameter is the device name of the RZ55 disk on your system:

```
ANALYZE/ERROR/EXCLUDE=VOLUME_CHANGES/INCLUDE=device-name:
```

This command analyzes the error log for errors found on the RZ55 device. If the error log displays any recovered errors for the RZ55 device, then your device may be subject to data loss, and further investigation will be needed to verify the integrity of your data. See the sample error report at the end of this section for an example of a recovered error. Other types of errors logged for the RZ55 device will not cause the problem described in this report.

If your error report indicates recovered errors on the RZ55 disk, the Analyze Disk Structure Utility (ANALYZE/DISK_STRUCTURE) may help to determine which files are affected, and whether the errors can be repaired. Please see the *VMS System Manager's Manual* for information on using ANALYZE/DISK_STRUCTURE.

If your disk has lost data, ANALYZE/DISK_STRUCTURE can repair the disk's file structure. However, you may also need to restore missing data from backup copies of the disk.

Sample Error Report

The following error report contains an example of a recovered error.

❶. If your RZ55 disk has been affected by the problem described in this section, your error report will include a recovered error similar to the one displayed in this example.

```
                V A X / V M S                SYSTEM ERROR REPORT
                COMPILED 15-SEP-1989 18:10 PAGE   1.

******************************* ENTRY       4. *******************************
ERROR SEQUENCE 41623.                          LOGGED ON:        SID 0A000004
DATE/TIME 11-SEP-1989 11:05:45.93                                SYS_TYPE 04010002
SCS NODE:                                                         VAX/VMS V5.2-1

DEVICE ERROR  KA420  CPU REV# 5.

RZ55 SUB-SYSTEM, UNIT _DKA0:

      HW REVISION      30303830
                                          HW REVISION = 0800

      ERROR TYPE             05
                                          EXTENDED SENSE DATA RECEIVED

      SCSI ID                00
                                          SCSI ID = 0.

      SCSI LUN               00
                                          SCSI LUN = 0.

      SCSI SUBLUN            00
                                          SCSI SUBLUN = 0.

      PORT STATUS      00000001
                                          %SYSTEM-S-NORMAL, NORMAL SUCCESSFUL
                                           COMPLETION

      SCSI CMD         49020008
                           0003
                                          READ

      SCSI STATUS            02
                                          CHECK CONDITION

EXTENDED SENSE DATA

      EXTENDED SENSE   000100F0
                       0C490200
                       00000000
                       00000018
                           0000

                                         RECOVERED ERROR ❶

      UCB$B_ERTCNT           01
                                          1. RETRIES REMAINING

      UCB$B_ERTMAX           00
                                          0. RETRIES ALLOWABLE

      ORB$L_OWNER      00010001
                                          OWNER UIC [001,001]

      UCB$L_CHAR       1CCD4008
                                          DIRECTORY STRUCTURED
                                          FILE ORIENTED
                                          SHARABLE
                                          AVAILABLE
                                          MOUNTED
                                          ERROR LOGGING
                                          ALLOCATED
                                          CAPABLE OF INPUT
                                          CAPABLE OF OUTPUT
                                          RANDOM ACCESS

      UCB$W_STS              0000
      UCB$L_OPCNT      00000149
                                          329. QIO'S THIS UNIT
```

```
 V A X / V M S                SYSTEM ERROR REPORT     COMPILED 15-SEP-1989 18:10
                                                                      PAGE   2.


       UCB$W_ERRCNT        0001
                                        1. ERRORS THIS UNIT
       IRP$W_BCNT          0600
                                        TRANSFER SIZE 1536. BYTE(S)
       IRP$W_BOFF          0000
                                        TRANSFER PAGE ALIGNED
       IRP$L_PID       00010012
                                        REQUESTOR "PID"
       IRP$Q_IOSB      00000001
                       00510000         IOSB, 0. BYTE(S) TRANSFERRED
ANALYZE/ERROR/EXCLUDE=VOLUME_CHANGES/INCLUDE=DISKS/OUT=H:A.DAT
```

2.11 Using the RZ23 as a System Disk

The RZ23 disk, which contains 104 megabytes of storage, is the smallest system disk that is supported for systems that run VMS Version 5.3 and DECwindows. If any of the following conditions exist on your system, it may be necessary to take special measures to allocate space on your RZ23 system disk:

- The RZ23 system disk is used as a cluster common disk with multiple system roots.
- The RZ23 system disk is used on a system with 16 megabytes or more of main memory.
- The RZ23 system disk contains many layered products.

If you have insufficient space on your RZ23 system disk, you should consider moving the page and swap files to a separate disk. If that is not possible, then you should consider tailoring off portions of the operating system that you do not need, or you should consider removing the system dump file. If none of these solutions is acceptable, then you should move to a larger system disk such as the RZ55, which contains over 300 megabytes of storage capacity.

2.12 LAT Terminal Service Problem

A problem with LTDRIVER can cause invalid service announcements to be delivered. This may cause unexpected names to appear in the list of services provided on a DECserver terminal server.

The problem only occurs if you do not define any service names for your LAT node. The normal SYS$MANAGER:LTLOAD.COM file provided with this release will define a service name for you. If you modify or do not use this file, be sure that you include the LAT Control Program (LATCP) command CREATE SERVICE/ID in your LAT startup commands. You need this only if you have not explicitly created any other service names.

2.13 VMS DECwindows Release Notes

This section contains system manager release notes applicable to DECwindows.

2.13.1 New Template File

The file DECW$SYLOGIN.COM is no longer shipped with the DECwindows kit. Instead, DECW$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.

2.13.2 Adding DECwindows at a Later Date

If DECwindows is not installed during the Version 5.3 installation or upgrade, you can install DECwindows later by using DECW$TAILOR to add the DECwindows options that you want. Enter the following command to run DECW$TAILOR:

```
$
```

2.13.3 ULTRIX Connection (UCX) and TCP/IP Transport Notes

This section contains release notes for the ULTRIX Connection (UCX) layered product and the DECwindows TCP/IP transport.

2.13.3.1 Problem Starting UCX Before DECnet

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

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

If you wish 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:

```
$
$
$
```

Then you could add the following line to SYSTARTUP_V5.COM:

```
$
```

2.13.3.2 Sequence Lost Errors

Under heavy interactive load, applications using the DECwindows TCP/IP transport may exhibit "sequence lost" error messages. DECW$PAINT is a typical application that can exhibit this behavior.

This behavior is caused by a known problem in UCX Version 1.0. It is expected that this problem will be corrected in the next release of UCX. There is no known workaround for this problem.

2.14 VMS Version 5.2-1 Release Notes

This section contains VMS system manager release notes that originated with Version 5.2-1.

2.14.1 DECnet–VAX Notes

This section contains release notes applicable to the DECnet–VAX package.

2.14.1.1 Fix for Repeated Maintenance Operations

When maintenance operations, such as loopback tests, were repeated immediately by using some synchronous devices, intermittent fatal device errors would occur. These intermittent fatal device errors do not occur in VMS Version 5.2-1 and higher versions.

2.14.1.2 Downline Task Loading to RSX–11S Systems

During downline task loading from VMS hosts to RSX–11S satellites, the final segments would occasionally be lost and the following error message would appear on the satellite console:

```
Task "program name" terminated.
Load failure.  Read error.
```

This problem has been corrected in VMS Version 5.2-1.

2.14.1.3 XDDRIVER Retransmit Timer Correction

Prior to VMS Version 5.2-1, the retransmit (select) timer was being set incorrectly for the DMV11 device, which caused circuit problems on multipoint lines and inordinate delays following lost or corrupted messages on point-to-point lines. Some DMV11 device users have overcome this problem by setting the DECnet–VAX RETRANSMIT TIMER line parameter to one-tenth its default value. The problem has been corrected. Resetting the line parameter is no longer necessary.

2.14.1.4 Using NETACP$BUFFER_LIMIT Logical Name to Override Default BYTLM Quota

The NETACP$BUFFER_LIMIT logical name can be used to override the default BYTLM quota given to the network ancillary control process (NETACP). NETACP$BUFFER_LIMIT should be defined in SYSTARTUP_V5.COM prior to invoking SYS$MANAGER:STARTNET.COM.

2.14.1.5 NML Checks for Illegal Address Configurations

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 will help prevent illegal address configurations from being defined in the permanent network database.

2.14.1.6 NCP/NML Requires OPER Privilege to Obtain Service Passwords

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

2.14.1.7 NETACP/NETDRIVER—Synchronization Problem Fixed

Information in the volatile database no longer directly references data structures in the nonpaged pool. Previously, if the data structures were deleted, a timing window in SMP environments could cause system crashes.

As of VMS Version 5.2-1, NETACP keeps an index into a properly synchronized table in nonpaged pool. The Network Control Program (NCP) allows system managers to modify the size of this table with the executor node parameter, MAXIMUM DECLARED OBJECTS.

Note: New images for SYS$LOADABLE_IMAGES:NETDRIVER.EXE and SYS$SYSTEM:NETACP.EXE are provided with this kit. Special care should be used so that the new NETACP.EXE image is not running on a system with a previous version of NETDRIVER.EXE, nor should the new NETDRIVER.EXE be loaded on a system with an older NETACP.EXE running. Following proper installation procedures will ensure this does not happen.

2.14.2 Executing CLUSTER_CONFIG.COM when Modifying VAXCluster Configurations

When adding or removing a machine from a cluster, or when upgrading a CPU on a current node, it is essential to execute CLUSTER_CONFIG.COM. CLUSTER_CONFIG is responsible for providing information to AUTOGEN, which is then used to set various SYSGEN parameters. Failure to execute CLUSTER_CONFIG could result in poor performance, a hang, or a crash of the system and perhaps the entire cluster.

2.14.3 New Batch-Oriented Procedure for AUTOGEN Feedback

A technique has been developed with AUTOGEN to make system management easier. The technique is a batch-oriented procedure that runs AUTOGEN in two stages.

The first stage runs AUTOGEN at peak times to collect the most accurate information possible, using the following command:

```
$
```

When this command is executed, users will not see any noticeable effect on the system because it is a small executable image.

The second stage runs AUTOGEN during the off-peak hours to process the information gathered above. The following command is used:

```
$
```

The resulting AGEN$FEEDBACK.REPORT is then mailed to the system manager, using the following command:

```
$
_$
```

This procedure automates AUTOGEN feedback, allowing the system manager to receive feedback reports from all nodes, on a regular basis, at one central location.

The system manager can review the report to see whether the load on the node is changing. If AUTOGEN calculations diverge from the current values, the system manager can correct the problem by executing one of the following commands:

```
$

$
```

The first command should be used if reboot is possible. If reboot is not possible, use the second command. This command sets the parameters so that they will be in effect at the next reboot.

The following procedure is an example of a batch procedure that runs AUTOGEN as previously described. This procedure should be used only as a reference while the system manager writes a more suitable batch procedure.

```
$ BEGIN$:   ! ++++++++++ AGEN_BATCH.COM ++++++++++
$  on warning then goto error$
$  on error then goto error$
$  on severe_error then goto error$
$  on control_y then goto error$
$!
$! Setup process
$!
$! Set process information
$      set process/priv=all/name="AUTOGEN Batch"
$! Keep log files to a reasonable amount
$      purge/keep=5 AGEN_Batch.log
$      time = f$time()                          ! Fetch current time
$  hour = f$integer(f$cvtime(time,,"hour"))     ! Get hour
$  today = f$cvtime(time,,"WEEKDAY")            ! Get Day of the week
$  if f$integer(f$cvtime(time,,"minute")) .ge. 30 then hour = hour + 1
$!
$! Start of working day...
$!
$ 1AM$:
$  if hour .le. 2
$     then
$     next_time = "today+0-14"
$     gosub submit$                            ! Resubmit yourself
$     set noon
$!
$!      Run AUTOGEN to setparams using the parameter values collected
```

```
$!      earlier in the day (that is, yesterday at 2:00pm)
$     if today .eqs. "Tuesday" .OR. today .eqs. "Thursday" .OR. -
        today .eqs. "Saturday"
$        then
$          @sys$update:autogen getdata testfiles feedback
$          mail/sub="Autogen Feedback Report for <NODE>" -
                sys$system:agen$feedback.report system
$        ! Clean up
$          purge/keep=7 sys$system:agen$feedback.report
$          purge/keep=7 sys$system:agen$feedback.dat
$          purge/keep=7 sys$system:params.dat
$          purge/keep=7 sys$system:autogen.par
$          purge/keep=7 sys$system:setparams.dat
$          purge/keep=7 sys$system:agen$addhistory.tmp
$          purge/keep=7 sys$system:agen$addhistory.dat
$        endif
$     goto end$
$     endif
$!
$ 2PM$:
$  if hour .le. 15
$     then
$     next_time = "today+0-17"
$     gosub submit$
$     if today .eqs. "Monday" .OR. today .eqs. "Wednesday" .OR. -
        today .eqs. "Friday"
$        then
$          @sys$update:autogen savparams savparams feedback
$        endif
$     goto end$
$     endif
$!
$ 5PM$:
$  if hour .le. 18
$     then
$     next_time = "tomorrow+0-1"
$     gosub submit$
$     endif
$!
$! End of working day...
$!
$ END$:     ! ---------- BATCH.COM ----------
$  exit
$!++
$! Subroutines
$!--
$!
$ SUBMIT$:
$  submit/name="AGEN_Batch"/restart/noprint -
   /log=AGEN_batch.log -
   /queue=sys$batch/after="'next_time'" sys$system:AGEN_batch.com
$  return
$!++
$! Error handler
$!--
$ ERROR$:
$  mail/sub="AGEN_BATCH.COM - Procedure failed." _nl: system
$  goto end$
```

2.14.4 SYSMAN Utility—DECnet TASK Object

The System Management Utility (SYSMAN) requires that the DECnet TASK object be defined in order for SYSMAN to perform operations on nodes outside the local cluster. The TASK object does not need to be enabled, but it must be defined in the object database of the node from which SYSMAN is being run. If the TASK object is not defined, SYSMAN may display errors similar to the following example:

```
SYSMAN>
Remote Password:
%SYSMAN-I-ENV, current command environment:
        Individual nodes: JOKER
        At least one node is not in local cluster
        Username USER will be used on nonlocal nodes
SYSMAN>
%SYSMAN-I-NODERR, error returned from node JOKER
%SYSTEM-F-NOSUCHOBJ, network object is unknown at remote node
```

Use the following commands to redefine the TASK object:

```
$
NCP>
NCP>
```

2.14.5 Running UETP

The User Environment Test Package (UETP) does not support the RRD40 compact disc drive. Before you run UETP on a VAXstation 3100, 3150, or MicroVAX 3100, make sure there is no compact disc in the RRD40 compact disc drive. UETP will support the RRD40 in a future VMS release.

2.14.6 Error Messages During Installation

If you receive the following error message during installation of the mandatory update or during installation of VMS Version 5.2-1, be sure that you reboot the system before you let users log in:

```
%INSTALL-E-FAIL, failed to replace entry
-SYSTEM-F-GPTFULL, global page table is full
```

3 Programmer Release Notes

This chapter contains information about the VMS Version 5.3 operating system that is of interest to programmers.

3.1 MTH$SINCOS Routine Correction

The removal of the MACRO-32 POLYF code from the MTH$SINCOS routine has given rise to a possible 1 least significant bit (LSB) difference between separate MTHSIN/MTHCOS calls and one MTH$SINCOS CALL (see the following example). This difference arose because MTH$SINCOS often treats intermediate F_float results as D_float values. This treatment was valid when POLYF code was in place, since the F_float values were in R0, and POLYF automatically cleared R1. The replacement code did not clear R1. Thus, depending upon what was in R1, R0/R1 was a D_float value that could round to an F_float value 1 LSB off from R0 itself.

This problem has been fixed by clearing R1 at all appropriate places, as shown in the following FORTRAN code example:

```
REAL*4 A, A2, A3
REAL*4 Z, Z2, Z3
REAL*4 X, X2, X3

EQUIVALENCE (X, Z), (X2, Z2), (X3, Z3)
DATA X /'0A954086'x/

A = Z ! 1.047198

A2 = SIN(A)
A3 = COS(A)

Z2 = A2
Z3 = A3

WRITE(6,1) A2, A3, X2, X3
      1FORMAT(X, 2F15.9, 4X, 2Z10)
END
```

```
============================================================

OPTIMIZED RESULTS (FORTRAN/OPTIMIZE/LIS/MACHINE)

     0.866025627    0.499999672      B3DB405D  FFF53FFF
                                        ^

============================================================

     0.866025627    0.499999672      B3DA405D  FFF53FFF
                                        ^

============================================================
```

3.2 Math Run-Time Library Notes

The Math Run-Time Library consists of two math shareable images: UVMTHRTL and MTHRTL. These two libraries may differ by 1 LSB in some cases. To ensure that you get the same mathematical results when running an application on multiple machines, it is recommended that you define UVMTHRTL for your math library by entering the following commands:

```
$
$
```

Note that some VAX machines use MTHRTL as the default library.

3.3 VMS Debugger Restrictions

This section describes the new known problems with the VMS Debugger for Version 5.3.

3.3.1 GO Command Limitation

After the program terminates, if you enter a GO command and specify an address expression with that command, any previously set breakpoints, tracepoints, or watchpoints are then ignored and are not reported.

3.3.2 Pascal Subrange Expressions Restriction

In Pascal subrange expressions, the low/high bound expressions must be constants.

3.3.3 SET IMAGE Command Limitation

For some large programs with many program sections (usually caused by many FORTRAN routines with many COMMON blocks) the debugger may receive an internal error during the processing of a SET IMAGE command. In such cases, the image cannot be debugged.

3.3.4 Requesting Help About Diagnostic Messages

If you type the command HELP MESSAGES to obtain information about debugger diagnostic messages, ignore the "Parameters" subtopic. To obtain help on a particular message from the list of subtopics, use the following syntax:

```
HELP MESSAGES message-ident.
```

3.3.5 RUN/DETACHED Command Entered After LINK/DEBUG Command

If you link a program with the command LINK/DEBUG and then execute the program in a detached process (with the command RUN/DETACHED), the debugger goes into an infinite loop. Instead, specify the command RUN/DETACHED/OUTPUT to circumvent this problem.

3.3.6 Specifying Search List for SYS$LIBRARY

The debugger does not run if you have specified a search list for SYS$LIBRARY, as in the following example:

$

In such cases, enter the following definition to correct the problem:

$

3.3.7 $WAKE Call Followed by $HIBER Call

If a program running under the debugger issues a $WAKE call followed by a $HIBER call, the debugger hibernates.

3.3.8 Register Window in DECwindows Interface

When using the debugger's DECwindows interface, you may see a FIND_ LINE internal error when you shrink the REG (register) window vertically with the window's resize button. You must then close the REG window and create a new one. Proceed as follows (the procedure describes one of the possible ways to create a REG window):

1 Choose Close Window from the File menu of the REG window.

2 Choose Window Setups from the Customize menu.

3 Choose the bottom window layout, which positions the REG and INST windows side by side between the SRC and OUT windows.

3.3.9 Watchpoints in Installed Writable Shareable Images

The technique for setting watchpoints in installed writable shareable images is missing from the debugger documentation. The technique 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"
```

You must set a nonstatic watchpoint because variables declared in such shareable images are typically static variables. By default, the debugger watches a static variable by write-protecting the page containing that variable. However, the debugger cannot write-protect a page in an installed writable shareable image. Therefore, the debugger must use the slower method of detecting changes, as for nonstatic variables—that is, by checking the value at the watched location after each instruction has been executed.

Note that if any other process modifies the watched location's value, the debugger may report that your program modified the watched location.

3.4 Small Computer System Interface (SCSI) Release Notes

This section contains release notes pertaining to the Small Computer System Interface (SCSI).

3.4.1 Third-Party Device Support

The *VMS Version 5.3 Small Computer System Interface (SCSI) Device Support Manual* is not intended to be a complete description of SCSI. Rather, it is a description of the interfaces and tools that the VMS operating system provides for access to SCSI-based peripherals.

If you are planning to develop SCSI-based applications, write to the following address for the latest draft of X3.131-198X, (SCSI II SPECIFICATION):

Global Engineering Documents
2805 McGaw
Irvine, CA 92714

3.4.2 SCSI Class Drivers and the DPT$V_NO_IDB_DISPATCH Bit

The DPT$V_NO_IDB_DISPATCH field, new in VMS Version 5.3, indicates that the field IDB$L_UCBLIST is not used to store the addresses of unit control blocks (UCBs) for this device. The DPT$V_NO_IDB_DISPATCH flag is used for SCSI class drivers, and any other drivers that use very large unit numbers, where an alternative method is used to locate the device UCBs during an interrupt.

The following example shows how this bit is used in the DPTAB macro of a SCSI class driver:

```
DPTAB-                          ; DPT-creation macro
     END=SK_END,-                   ; End of driver label
     ADAPTER=NULL,-                 ; Adapter type
     UCBSIZE=<UCB_K_SK_UCBLEN>,-    ; Length of UCB
     NAME=SKDRIVER,-                ; Driver name
     FLAGS=<DPT$M_SMPMOD!-          ; Driver runs in SMP environment
          DPT$M_NO_IDB_DISPATCH>         ; Don't fill in IDB$L_UCBLST
```

3.5 PPL$TRIGGER_EVENT Routine Memory Problem

There is a known problem in the PPL$TRIGGER_EVENT routine, which causes PPL$ to lose a small amount of its internally managed shared memory during each call. After a large number of calls to PPL$TRIGGER_EVENT, PPL$ runs out of internal shared memory. The exact number of calls varies, depending upon the size of the PPL application specified in the call to PPL$INITIALIZE. At this point, any calls to routines that create PPL objects or that cause the process to block return a PPL$_INSVIRMEM error.

The problem may be alleviated slightly by specifying a larger size in the call to PPL$INITIALIZE. However, continued calls to PPL$TRIGGER_EVENT eventually cause PPL$ to run out of memory.

This problem will be fixed in the next release of the VMS operating system.

3.6 Use of TIMEDWAIT Macro on MicroVAX and VAXstation 3100 Series Systems

The TIMEDWAIT macro, when used on MicroVAX or VAXstation 3100 Series systems, may result in a delay that is up to 10 times the desired value. This occurs because of an error in the routine that calculates the base delay constants (CPU$L_UBDELAY and CPU$L_TENUSEC) that are referenced in the TIMEDWAIT macro. Note that the delay will never be less than the value specified.

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

3.7 Appending to Shared Sequential Files — Restriction Removed

The temporary restriction against using the deferred write (DFW) option when appending records to shared sequential files, which was documented in the *VMS Version 5.0 Release Notes*, has been removed. The problem of potential corruption to sequential files was corrected in VMS Version 5.2.

3.8 VMS DECwindows Release Notes

The following sections contain release notes applicable to VMS DECwindows.

3.8.1 Xlib Enhancements

The DECwindows implementation of Xlib is now equivalent to the Massachusetts Institute of Technology (MIT) Release 3 version. This version fixes several problems and includes the following new routines:

- MAX REQUEST SIZE—Returns the maximum request size the server will allow.
- RESOURCE MANAGER STRING—Returns the resource manager property from the server's root window of screen zero.

- DISPLAY MOTION BUFFER SIZE—Returns the size of the motion buffer.
- DISPLAY KEYCODES—Returns the minimum and maximum number of keycodes the server supports.
- VISUAL ID FROM VISUAL—Returns the Visual ID associated with a specified visual.

Programs that directly reference a field one of these routines specifies (for example, the maximum request size specified by MAX REQUEST SIZE) should use the appropriate new routine, to ensure Release 3 compatibility. The VAX and MIT C bindings for these routines are documented in the *VMS DECwindows Programming Documentation Supplement*.

3.8.2 DECwindows Server Notes

This section contains release notes relevant to the DECwindows server.

3.8.2.1 Server Modifications

The DECwindows server has been modified to perform error checking on the dashes element for CREATE GC and CHANGE GC. If a value of zero is specified for the dashes element, a Bad Value error will result. Note that for the CREATE GC request, specifying the dashes element to be zero will result in the GC not being created.

For the INTERN ATOM request, if the value of the **only_if_exists** argument is neither true nor false, a Bad Value error is returned.

The server execution of the MAP SUBWINDOWS and UNMAP SUBWINDOWS requests has improved considerably. To improve application performance, use these requests rather than mapping or unmapping individual windows.

3.8.2.2 Server Problems and Restrictions

The following problems and restrictions are known to 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.

  If the window has a bit gravity of FORGET, in two cases the child window gravity will be incorrect. STATIC window gravity results in the child window being placed in the wrong location. UNMAP gravity can cause the windowing system to be put into a deadlock state. To eliminate the deadlock condition, the window manager process must be stopped and restarted.

- The following problems occur when the server is replaying events collected during a synchronous grab:

  — The screen for the events being replayed is reported as the current screen. In multihead systems, the event may have occurred on a different screen during the synchronous grab collection.

— Motion events may be time-stamped with the current time during replay instead of the time the event was collected.

3.8.3 SET DISPLAY DCL Command Default Setting

The default setting for the SET DISPLAY command is SUPERVISOR_MODE. This is different from previous versions, where the default was equivalent to EXECUTIVE_MODE.

When displaying graphics to an ULTRIX server, the ULTRIX node name must be placed in quotation marks ("") if the name contains any lowercase characters.

For example, to display on node "uuu" type the following command:

$

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

3.8.4 XUI Toolkit Changes and Restrictions

This section discusses changes to the XUI Toolkit.

3.8.4.1 Internal Format of Compound Strings

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.

3.8.4.2 Performance of INIT GET SEGMENT

The change to compound strings has significantly decreased the performance of the routine DwtInitGetSegment (DWT$INIT_GET_SEGMENT) when used to fetch multiple segments from a compound string. The routines STRING INIT CONTEXT and STRING FREE CONTEXT have been added and should be used for better performance.

3.8.4.3 Font Unit Values

The XUI Toolkit formerly 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 for any other font, the value could be different.

3.8.4.4 List Box Dynamic Sizing

The preferred way to change the list box width is with the Set Values routine. The list box does not support dynamic dimension changes. Therefore, placing list boxes inside attached dialog boxes—with attachments to both the left and right side of the attached dialog box—may, under certain circumstances, lead to the Items Selectable area not spanning the full width of the list box. When an attached dialog box changes size, all dialog boxes for which that box is a root are dynamically resized.

Also, the preferred way to change the list box height is through the ItemsCount attribute. Modifying the height attribute will not reconfigure the number of visible items. For example, doubling the list box height but not modifying the ItemsCount attribute results in a list box only half full of items, with the remaining area left blank.

3.8.4.5 FileSelection Callback Correction

In the previous version, the fileselection widget did not correctly set the dirmask field in the callback DwtFileSelectionCallbackStruct. This meant that applications attempting to determine the directory mask upon widget callback were forced to use the GetValues mechanism. This problem has been fixed in the current version. Applications that used GetValues in their callback routines should use the callback field instead to save time.

3.8.4.6 Selection Push Buttons

Setting the Ok and Cancel push button labels to NULL or empty strings will not remove the push buttons, but instead results in blank labels.

3.8.4.7 Known Restrictions

The following problems and restrictions are known to exist in this version of the XUI Toolkit:

- **Unseen LeaveWindow events**—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 fixed in a future release.

- **Dialog Box Race Condition**—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 fixed in a future release.

- If you run an application linked against an earlier Toolkit version and an application linked against the current Toolkit, you will not be able to cut from one and paste to the other. No error message appears. Applications formerly shared a common clipboard, but applications linked against the current Toolkit share a separate and independent clipboard. This incompatibility was necessary to fix a more serious source of potential problems with the clipboard and steps have been taken to ensure that no similar clipboard incompatibility will exist among future versions.
- Unlike other Toolkit callbacks, the destroy callback only returns two arguments: widget id and tag. The reason argument is NULL. Applications therefore should avoid setting destroy callbacks to call 'general' callback routines (handling numerous actions such as activate, arm, and disarm) that depend on a reason argument. For Ada developers this may be particularly important, since Ada requires that all declared arguments be passed.
- In certain circumstances, the help widget's list box selectable area does not span the entire width of the widget. However, all items may still be selected by clicking the mouse button on the item text.
- There is a problem with the DwtToggleButtonSetState routine that affects toggle buttons with on/off pixmaps. If the widget has not been realized, SetState correctly updates the toggle button value but not the on/off pixmap. If the widget has been realized, it may display the wrong pixmap. To circumvent this problem, use the SetValues mechanism instead setting DwtNvalue to True. This correctly updates the pixmap as well as the toggle button value, regardless of whether the widget is realized.
- In the colormixing widget, the Red, Green, and Blue labels, the OK callback, and the Original Color resources cannot be modified after widget creation by using SetValues. You must specify label changes during widget creation.
- Pop-up dialog boxes with no icon button (DwtNnoIconify set true) are initially created as icons (DwtNiconic true). Not only does the icon pop-up not have an icon box (and therefore cannot be popped up), but operations such as SetInputFocus on the icon pop-up cause an access violation.
- An attempt to obtain context sensitive help on the About frame of the XUI Toolkit help widget can cause the application to abort.
- 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);
```

3.8.5 User Interface Language (UIL) Compiler

This section describes additional revisions to the User Interface Language (UIL) compiler and associated documentation that are not covered in the *VMS DECwindows Programming Documentation Supplement.*

3.8.5.1 Specifying Multiline Compound Strings

In VMS Version 5.3, the UIL compiler does not consistently process newline characters (\n) that are embedded in compound strings. The effect of a newline character that is embedded in a compound string is now solely dependent on the character set specified, and the result may not always be the creation of a multiline compound string.

To guarantee the creation of a multiline compound string, you must use the SEPARATE clause in the COMPOUND_STRING function and the concatenation operator (&) to join the segments into a multiline compound string. The SEPARATE clause takes the form SEPARATE = *boolean-expression*, and implements the newline character for VMS Version 5.3. For example, in VMS Version 5.1, the UIL compiler would generate a multiline compound string from the following input:

```
value
    sample_string : compound_string ("Hello\nWorld!");
```

To guarantee the same result in VMS Version 5.3, you must input the following:

```
value
    sample_line1 : compound_string ("Hello", separate = true);
    sample_line2 : compound_string ("World!");
    sample_string : sample_line1 & sample_line2;
```

To retain VMS Version 5.1 behavior of the newline character (\n) in a compound string, compile your UIL specification file using the /VERSION qualifier as follows:

```
$
```

See the *VMS DECwindows User Interface Language Reference Manual* for more information on the COMPOUND_STRING function. See the *VMS DECwindows Programming Documentation Supplement* or the online edition of the *VMS DECwindows User Interface Language Reference Manual* for more information on the /VERSION qualifier.

3.8.5.2 Arguments Not Directly Supported in UIL

The following arguments are not directly available in the UIL compiler for VMS Version 5.3:

UIL Argument Name	Data Type	Default Value	Valid for These Objects
auto_unmanage	Boolean	true	selection file_selection
auto_unrealize	Boolean	false	selection file_selection
direction_r_to_l	integer	DwtDirectionRightDown	color_mix
grab_key_syms	translation_table	Default translation table syntax	color_mix
menu_extend_last_row	Boolean	true	menu_bar popup_menu pulldown menu radio_box work_area_menu
no_resize	Boolean	true	color_mix
take_focus	Boolean	true (modal); false (modeless)	color_mix

To set one of these arguments in a UIL module, use the ARGUMENT function to define the argument. For information on the ARGUMENT function, refer to Section 2.5.7 in the VMS Version 5.1 edition of the *VMS DECwindows User Interface Language Reference Manual*. (In the online edition for VMS Version 5.3, refer to Section 2.5.9.)

3.8.5.3 Width Argument for Work Area Menu

In the *VMS DECwindows User Interface Language Reference Manual* for VMS Version 5.1, the default value description of the **width** argument in Table B.33, Work Area Menu, is incorrect. The revised description is as follows:

> If orientation is vertical, width is the maximum entry width or 16 pixels. If orientation is horizontal, width is the sum of width and spacing or 16 pixels.

The online edition of the *VMS DECwindows User Interface Language Reference Manual* for VMS Version 5.3 contains the correct description.

3.8.5.4 Color Name Strings

The command to display the color name strings understood by the VMS DECwindows servers was documented incorrectly for VMS Version 5.1 in both the online and hardcopy versions of the following manuals:

VMS DECwindows Guide to Application Programming	Section 3.2.7.6, p. 3-21
VMS DECwindows User Interface Language Reference Manual	Section 2.5.1, p. 2-17

The correct command for displaying color name strings is as follows:

```
$
```

The online editions of these manuals for VMS Version 5.3 show the correct command.

3.8.5.5 References to XmATTACH_NONE in the *VMS DECwindows User Interface Language Reference Manual*

In Appendix B of the *VMS DECwindows User Interface Language Reference Manual*, all occurences of XmATTACH_NONE should be changed to DwtAttachNone.

3.8.6 Font Quality Cleanup

The 29 Core Fonts (Times, New Century, Helvetica, Courier, Souvenir, AvantGarde, Lubalin, and Symbol families found in the LPS40 and LN03R) have had glyph quality corrections in both 75- and 100-dpi formats since the last release.

The contents of the COPYRIGHT property field have been corrected in both resolution sets.

The height of accented capital letters has been increased so the letter height under the accent matches the height of the corresponding nonaccented capital.

There have been no other property content or character encoding changes since the last release.

3.8.7 Terminal Font Changes

The terminal fonts have been re-encoded to a true ISO Latin 1 character set from a hybrid of the DEC Multinational Character Set (MCS). This change resulted in the following changes:

- Terminal DECtech fonts
  - — Moving the line drawing set out of the Terminal set and into the Terminal-DECtech set, and changing the character codes from C0 space to GL space
  - — Changing the DECTech character codes from GL space to GR space

- Terminal fonts
  - — Adjusting the GL and GR space to true ISO Latin 1 (no impact on 7-bit ASCII; some 8-bit characters impacted)
  - — Moving three MCS characters, not in ISO Latin 1, to the C1 space

No coding changes have been made to the 7-bit ASCII characters. Therefore, using the new terminal fonts with the previous version of DECterm is suitable for the ASCII characters. The line drawing, technical set and ISO Latin C1 set do not work properly with the previous version of DECterm.

Note that the terminal fonts are intended solely for use by DECterm and should not be used by other clients.

4 Documentation Release Notes

This chapter contains information about the documentation provided with VMS Version 5.3. It also lists the changes you should make to specific manuals in your VMS documentation set.

4.1 VMS Version 5.3 Documentation

VMS Version 5.3 documentation includes new and revised manuals describing Version 5.3 upgrade and installation procedures, various new VMS features, enhancements to the DECwindows user and programming interfaces, expanded Compound Document Architecture (CDA) facilities, and device support for the Small Computer System Interface (SCSI).

The following manual ships with the software media kit:

- *VMS Version 5.3 Upgrade and Installation Procedures*—Order number AA-NG61B-TE

The manuals in the Version 5.3 Release Notes Kit (Order number QA-001AY-GZ) include:

- *VMS Version 5.3 Release Notes*—Order number AA-MG16B-TE
- *VMS Version 5.3 New Features Manual*—Order number AA-MG29B-TE
- *VMS Version 5.3 Small Computer System Interface (SCSI) Device Support Manual*—Order number AA-PAJ2A-TE

The following revised manuals update the DECwindows user documentation within the VMS Base Documentation Set (order number QA-09SAA-GZ):

- *VMS DECwindows User's Guide*—Order number AA-MG18B-TE
- *VMS DECwindows Desktop Applications Guide*—Order number AA-MG19B-TE

You can also order DECwindows user documentation separately from the Base Set; the DECwindows User Documentation Kit order number is AA-09SAB-GZ.

The DECwindows Programming Documentation Kit (Order number QA-001AM-GZ) is a multivolume documentation set that can be ordered separately. The new and revised manuals within this kit include:

- *VMS DECwindows Transport Manual*—Order number AA-PABWA-TE
- *CDA Reference Manual*—Order number for the volume is QA-09SAC-GZ. Order number for Part I is AA-PABUA-TE; order number for Part II is AA-PABVA-TE.

- *VMS DECwindows Programming Documentation Supplement*—Order number QA-09SAD-GZ

For further information about VMS documentation and how to order it, please refer to the *Overview of VMS Documentation.*

4.2 Correction to the *VMS Version 5.3 New Features Manual*

On page 7-45, add the following entry to the list of Condition Values Returned for SYS$DNS:

SS$N_BADPARAM Bad parameter value.

4.3 Correction to the *VMS Version 5.1 Release Notes*

The General User chapter of the *VMS Version 5.1 Release Notes* incorrectly states the version of VMS Services for MS-DOS that is compatible with VMS Version 5.1.

The note should read that Version 2.1 of the VMS Services for MS-DOS is compatible with VMS Version 5.1.

4.4 Correction to the *VMS Version 5.2 Release Notes*

On page 3-38, replace the first sentence of Section 3.17.3 with the following text:

> Under certain rare circumstances, the DEQNA Ethernet adapter, running at a revision prior to FCO K3, in large and complex Ethernet configurations, may receive corrupted data.

4.5 Correction to the *VMS Version 5.2 New Features Manual*

This section clarifies the recommendations made in the *VMS Version 5.2 New Features Manual* on using default DECnet object accounts.

4.5.1 Changes Made in NETCONFIG.COM for VMS Version 5.2

In VMS Version 5.2, the DECnet network configuration command procedure SYS$MANAGER:NETCONFIG.COM was modified to provide several options for restricting default DECnet access. Prior to Version 5.2, NETCONFIG created a default DECnet account named [DECNET]. That account provided default access to all network objects and applications, such as MAIL, PHONE, file access listener (FAL), and so on, that were not restricted by other forms of access control. If you chose to limit default access, it was necessary to manually enter all the appropriate commands in the user authorization file (UAF), using the Authorize Utility (AUTHORIZE), and in the network configuration database using Network Control Program (NCP) commands.

4.5.2 Default DECnet Access Options by Means of NETCONFIG.COM

NETCONFIG.COM now provides two different ways to limit default network access. The most restrictive form is to not create the default DECnet account, but to grant default access for certain system objects by creating a different default account for each object that you want to use. Using NETCONFIG.COM, you can create individual default accounts for one or more of the following network objects:

- MAIL
- File access listener (FAL)
- PHONE
- Network management listener (NML)
- Loopback mirror (MIRROR)
- VMS Performance Monitor (VPM)

The rationale for creating individual default accounts for each object is that it provides a more secure environment for DECnet objects to run in. For example, you may want to enable security auditing features for the individual default MAIL or FAL accounts.

The second, less restrictive form of default access is to create a default DECnet account but to disable default access to user-written programs and procedures (also known as TASK objects).

For sites with high security requirements, or sites where it is difficult to recognize all the intended users, Digital advises *against* creating a FAL account. To control which users gain access, these sites may establish one or more proxy accounts for specific purposes.

4.5.3 Recommendation on the Use of FAL

The chapter on "NETCONFIG.COM Security Enhancements" in the *VMS Version 5.2 New Features Manual* states:

> FAL, if enabled by the default DECnet account or a separate default account, makes a system vulnerable to unauthorized access. Digital advises against creating a default account for FAL. Other, more secure access methods are available. For more information about other secure access methods and the security implications of a FAL account, refer to the *Guide to VMS System Security*.

The ramification of *not* using FAL with a default account means that remote file requests must include explicit file access control information, or the local system manager must set up proxy access for remote users. Consider an example with a local node ETHQKE, and a remote node MISHA with no default account. Entering the command $DIR MISHA:: from node ETHQKE produces the following messages:

```
%DIRECT-E-OPENIN, error opening MISHA::*.*;* as input
-RMS-E-FND, ACP file or directory lookup failed
-SYSTEM-F-INVLOGIN, login information invalid at remote node
```

However, you can access node MISHA by entering the command $ DIR MISHA"Username Password":: from node ETHQKE.

The system manager could also, by using AUTHORIZE, enable proxy access for node ETHQKE by adding REMOTE_USER_FOO, as shown in the following example:

```
$
$
UAF>
UAF>
```

Entering the command $ DIR MISHA:: from node ETHQKE would then give user ETHQKE::REMOTE_USER_FOO access to remote node MISHA by proxy; MISHA then associates this account with the account LOCAL_USER on node MISHA.

4.5.4 Related Documentation

For more information about secure access methods and the security implications of a FAL account, refer to the *Guide to VMS System Security*.

For more detailed information about NETCONFIG.COM, and NETCONFIG_UPDATE.COM, refer to the *VMS Version 5.2 New Features Manual*.

4.6 $MOUNT System Service Addition

The following text for the /NOREBUILD flag for the $MOUNT system service was omitted from the Version 5.2 *VMS System Services Reference Manual*:

> The volume to be mounted should be returned to active use immediately, without performing a rebuild operation. A rebuild operation can consume a considerable amount of time, depending on the number of files on the volume and, if quotas are in use, on the number of different file owners. The volume can be rebuilt later with the DCL command SET VOLUME/REBUILD to recover the free space; for more information, see the *VMS DCL Dictionary*.
>
> If a disk volume is dismounted improperly, for example, during a system failure, it must be rebuilt to recover any caching limits that were enabled on the volume at the time of the dismount. By default, $MOUNT attempts to rebuild the disk volume. A successful rebuild operation includes reclaiming all the available free space. Therefore, you must mount all of the volume set members.
>
> MNT$M_NOREBUILD applies only to disks.

Also, on page SYS-355 of the *VMS System Services Reference Manual*, the description of the $MOUNT flag, MNT$M_NOMNTVER, incorrectly states that the flag applies only to disks. As of Version 5.0, mount verification applies to tapes as well as disks.

4.7 Linker Image-ID Field Correction

In Section 3.3 (Link Options), IDENTIFICATION=id-name subsection, of the *VMS Linker Utility Manual*, the maximum length of the image-id field is incorrectly stated as 39 characters. The correct maximum length of the image-id field is 15 characters.

4.8 VAXstation 3100 Model Information

VMS Installation and Operations: VAXstation 3100, 3150, and MicroVAX 3100, Version 5.2-1 referred to the VAXstation 3150 computer. This has been corrected in the most recent version of the manual. The correct system names are the VAXstation 3100 Model 30 and Model 40, and the VAXstation 3100 Model 38 and Model 48 systems. The manual is now called *VMS Installation and Operations: VAXstation 3100, MicroVAX 3100 Series*.

4.9 Corrections to the *VMS DCL Dictionary*

This section contains corrected descriptions of the CONVERT /DOCUMENT and VIEW commands.

4.9.1 CONVERT/DOCUMENT Command

The DCL command CONVERT/DOCUMENT converts a revisable format file to another revisable or final form file. You can use this command only if you have DECwindows installed on your system.

The CONVERT/DOCUMENT command has the following format:

CONVERT/DOCUMENT *input-file[/FORMAT=format-name]* - *output-file[/FORMAT=format-name]*

Parameters

input-file
Specifies the file to be converted. The default file type is DDIF.

output-file
Specifies the name of the converted file. The default file type is DDIF.

Qualifiers

/FORMAT=format-name
Specifies the encoding format of the input or output file. The default format is DDIF for both input and output files.

Input formats bundled with the VMS operating system and their default file types are as follows:

Input Format	File Type
DDIF	DDIF
DTIF	DTIF
TEXT	TXT

Output formats bundled with the VMS operating system and their default file types are as follows:

Output Format	File Type
DDIF	DDIF
DTIF	DTIF
TEXT	TXT
PS	PS
ANALYSIS	CDA$ANALYSIS

Note: The DTIF format and file type are new in VMS Version 5.3.

Digital's CDA Converter Library is a layered product that provides additional input and output formats. Independent software vendors who write DDIF- and DTIF-conforming applications and front and back end converters also provide input and output formats that are layered on the VMS operating system. Contact your system manager for a complete list of input and output formats available on your system.

/OPTIONS=*options-filename*

Specifies a file that contains processing options for both input and output. The default file type for a VMS options file is CDA$OPTIONS.

Creating the Options File

You can create an options file that contains all the input and output processing options to be applied during the conversion of the input file to the output file. These processing options affect how your input file is processed and how your output file is created or displayed.

Each line of the options file specifies a format keyword (for example, PS for PostScript) that can be followed optionally by _INPUT or _OUTPUT to restrict the option to the front or back end converter. The second keyword is a valid processing option preceded by one or more spaces, tabs, or a slash (/) and can contain upper- and lowercase alphabetic characters (alphabetic case is not significant), digits (0–9), dollar signs ($), and underscores (_). If an option requires you to specify a value, the option keyword can be separated from the value by one or more spaces or tabs, or by an equal (=) sign. Each line can optionally be preceded by spaces and tabs.

The following example is a typical entry in an options file:

```
PS PAPER_HEIGHT 10
```

In this example, the extension _OUTPUT is not required for the format keyword, since PostScript is only available as an output format. By default, the value specified for PAPER_HEIGHT is in inches.

To specify several options for the same input or output format, you must specify each option on a separate line. The CDA Converter checks the input format and the output format you specified on the command line and, if the processing options in your options file are valid for the input or output format, the options are applied during the conversion of your file. If you specify an invalid option for an input or output format or an invalid value for an option, the CDA Converter returns an error message. Each input and output format that supports processing options specifies any restrictions or special formats required when specifying processing options.

CDA Converter processing options available for several of the file formats that are bundled with VMS are listed in the following sections.

Text Back End Processing Options

The text back end converter supports the following options:

- ASCII_FALLBACK [ON,OFF]

  Causes the text back end converter to output text in 7-bit ASCII. The fallback representation of the characters is described in the ANSI ASCII standard. If this option is not specified, the default is OFF; if this option is specified without a value, the default is ON.

- CONTENT_MESSAGES [ON,OFF]

  Causes the text back end converter to put a message in the output file each time a nontext element is encountered in the in-memory CDA structures. If this option is not specified, the default is OFF; if this option is specified without a value, the default is ON.

- HEIGHT *value*

  Specifies the maximum number of lines per page in your text output file. If you specify 0, the number of lines per page will correspond to the height specified in your document. If you additionally specify OVERRIDE_FORMAT, or if the document has no inherent page size, the document is formatted to the height value specified by this option. The default height is 66 lines.

- OVERRIDE_FORMAT [ON,OFF]

  Causes the text back end converter to ignore the document formatting information included in your document, so that the text is formatted in a single large galley per page that corresponds to the size of the page, as specified by the HEIGHT and WIDTH processing options. If this option is not specified, the default is OFF; if this option is specified without a value, the default is ON.

  SOFT_DIRECTIVES [ON,OFF]

  Causes the text back end converter to obey the soft directives contained in the document when it creates your text output file. If this option is not specified, the default is OFF; if this option is specified without a value, the default is ON.

- WIDTH *value*

  Specifies the maximum number of columns of characters per page in your text output file. If you specify zero, the number of columns per page will correspond to the width specified in your document. If you additionally specify OVERRIDE_FORMAT, or if the document has no inherent page size, the document is formatted to the value specified by this processing option. If any lines of text exceed this width value, the additional columns are truncated. The default width is 80 characters.

PostScript Back End Processing Options

The PostScript back end converter supports the following processing options:

- PAPER_SIZE *size*

  Specifies the size of the paper to be used when formatting the resulting PostScript output file. Valid values for the *size* argument are as follows:

Keyword	Size
A0	841 x 1189 millimeters (33.13 x 46.85 inches)
A1	594 x 841 millimeters (23.40 x 33.13 inches)
A2	420 x 594 millimeters (16.55 x 23.40 inches)
A3	297 x 420 millimeters (11.70 x 16.55 inches)
A4	210 x 297 millimeters (8.27 x 11.70 inches)
A	8.5 x 11 inches
B	11 x 17 inches
C	17 x 22 inches
D	22 x 34 inches
E	34 x 44 inches
LEDGER	11 x 17 inches
LEGAL	8.5 x 14 inches
LETTER	8.5 x 11 inches
LP	13.7 x 11 inches
VT	8 x 5 inches

  The A paper size (8.5 x 11 inches) is the default.

- PAPER_HEIGHT *height*

  Specifies a paper size other than one of the predefined values provided. The default paper height is 11 inches.

- PAPER_WIDTH *width*

  Specifies a paper size other than one of the predefined sizes provided. The default paper width is 8.5 inches.

- PAPER_TOP_MARGIN *top-margin*

  Specifies the width of the margin provided at the top of the page. The default value is 0.25 inch.

- PAPER_BOTTOM_MARGIN *bottom-margin*

  Specifies the width of the margin provided at the bottom of the page. The default value is 0.25 inch.

- PAPER_LEFT_MARGIN *left-margin*

  Specifies the width of the margin provided on the left-hand side of the page. The default value is 0.25 inch.

- PAPER_RIGHT_MARGIN *right-margin*

  Specifies the width of the margin provided on the right-hand side of the page. The default value is 0.25 inch.

- PAPER_ORIENTATION *orientation*

  Specifies the paper orientation to be used in the output PostScript file. The valid values for the *orientation* argument are as follows:

Keyword	Meaning
PORTRAIT	The page is oriented so that the larger dimension is parallel to the vertical axis.
LANDSCAPE	The page is oriented so that the larger dimension is parallel to the horizontal axis.

  The default is PORTRAIT.

- EIGHT_BIT_OUTPUT *{ON,OFF}*

  Specifies whether the PostScript back end converter should use 8-bit output. The default value is ON.

- OUTPUT_BUFFER_SIZE *output-buffer-size*

  Specifies the size of the output buffer. The value you specify must be within the following range:

$$64 \leq output - buffer - size \leq 256$$

  The default value is 132.

- SOFT_DIRECTIVES *{ON,OFF}*

  Specifies whether the PostScript back end converter processes soft directives in the DDIF file in order to format output. (Soft directives specify such formatting commands as new line, new page, and tab.) If the PostScript back end processes soft directives, the output file will look more like you intended. The default value is ON.

- WORD_WRAP *(ON,OFF)*

  Specifies whether the PostScript back end converter performs word wrapping of any text that would exceed the right margin. The default value is ON. If you specify OFF, the PostScript back end allows text to exceed the right margin.

- PAGE_WRAP *(ON,OFF)*

  Specifies whether the PostScript back end converter performs page wrapping of any text that would exceed the bottom margin. The default value is ON.

- LAYOUT *(ON,OFF)*

  Specifies whether the PostScript back end converter processes the layout specified in the DDIF document. The default value is ON.

Analysis Back End Processing Option

The Analysis back end converter produces an analysis of the CDA in-memory structure in the form of text output showing the named objects and values stored in the document. This is useful for debugging DDIF application programs.

The Analysis back end converter supports an /INHERITANCE processing option that specifies that the analysis is shown with attribute inheritance enabled. Inherited attributes are marked by "[default]" in the output.

Domain Conversion Processing Options

When you are converting any table format to any document format, you can specify the following processing options by using a format name of DTIF_TO_DDIF.

- COLUMN_TITLE

  Enables display of the column titles as contained in the column attributes centered at the top of the column.

- CURRENT_DATE

  Enables display of the current date and time in the bottom left corner of the page. The value is formatted according to the document's specification for a default date and time.

- DOCUMENT_DATE

  Enables display of the document date and time as contained in the document header in the top left corner of the page. The value is formatted according to the document's specification for a default date and time.

- DOCUMENT_TITLE

  Enables display of the document title, or titles, as contained in the document header centered at the top of the page, one string per line.

- PAGE_NUMBER

  Enables display of the current page number in the top right corner of the page.

- PAPER_SIZE *size*

  Specifies the size of the paper to be used when formatting the resulting PostScript output file. The values are the same as those for the PostScript back end.

- PAPER_HEIGHT *height*

  Specifies a paper size other than one of the predefined values provided. The default paper height is 11 inches.

- PAPER_WIDTH *width*

  Specifies a paper size other than one of the predefined sizes provided. The default paper width is 8.5 inches.

- PAPER_TOP_MARGIN *top-margin*

  Specifies the width of the margin provided at the top of the page. The default value is 0.25 inch.

- PAPER_BOTTOM_MARGIN *bottom-margin*

  Specifies the width of the margin provided at the bottom of the page. The default value is 0.25 inch.

- PAPER_LEFT_MARGIN *left-margin*

  Specifies the width of the margin provided on the left-hand side of the page. The default value is 0.25 inch.

- PAPER_RIGHT_MARGIN *right-margin*

  Specifies the width of the margin provided on the right-hand side of the page. The default value is 0.25 inch.

- PAPER_ORIENTATION *orientation*

  Specifies the paper orientation to be used in the output file. The values are the same as those for the PostScript back end.

Example

```
$
_$
```

This command converts an input file named FOOBAR.DTIF, which has the DTIF format, to an output file named MOOMAR.DDIF, which has the DDIF format. The specified options file is named OPTIONS.CDA$OPTIONS.

4.9.2 VIEW Command

The DCL command VIEW invokes the Compound Document Architecture (CDA) Viewer, which lets you view a compound document file on a character cell terminal or on a DECwindows display. Note that because of the limitations on some smaller device screens, some qualifiers concerning terminal display cannot be processed.

The VIEW command has the following format:

VIEW *input-file*

Parameter

input-file
Specifies the name of the file to be viewed. If you do not specify an input file name, you are prompted for one. You cannot use wildcard characters in the file name. The default input file-encoding format is DDIF, and the default file type is DDIF. Valid input file formats are any of those for which there is a CDA converter front end installed on the system.

Qualifiers

/FORMAT[=*fmt-name*]
/FORMAT=DDIF (default)
Specifies the format of your input file. The input formats that you can use with the CDA Viewer depend on the CDA converters installed on your system. The default input format is DDIF. Input formats bundled with the VMS operating system and their default file types are as follows:

Input Format	File Type
DDIF	DDIF
DTIF	DTIF
TEXT	TXT

Additional input formats are provided in Digital's CDA Converter Library, a layered product. Independent software vendors who write DDIF- and DTIF-conforming applications and front and back end CDA converters also provide input formats that are layered on the VMS operating system. Contact your system manager for a complete list of input formats available on your system.

/HEIGHT=*nn*
Specifies the height of the page in number of characters. If you specify the /OVERRIDE_FORMAT qualifier, or if the document being viewed has no inherent format, this page height is used. On the DECwindows display, the default height is 66 lines, which is equivalent to the default page height of 11 inches. On character cell displays, the page height defaults to your terminal's screen height. However, if you use the /OUTPUT qualifier, the page height depends on the page height of your document.

Note: The /HEIGHT qualifier is new in VMS Version 5.3.

/INTERFACE=DECWINDOWS
/INTERFACE=CHARACTER_CELL (default)
Specifies the type of display you are using.

Note: The /INTERFACE qualifier is new in VMS Version 5.3.

/OPTIONS=*options-filename*

Specifies the name of a file that contains processing options. The default file type for a VMS options file is CDA$OPTIONS. This qualifier is used only with input formats for which you can specify processing options.

/OUTPUT[=*output-filename*]
/NOOUTPUT (default)

Specifies a file to which you want output directed instead of having it displayed on the screen. You cannot use this qualifier when you have specified the /INTERFACE=DECWINDOWS qualifier.

If you specify the /OUTPUT qualifier but you do not specify a file name, the CDA Viewer creates a file with the same name as your input file but with a file type of LIS. If you specify the /OUTPUT qualifier, you cannot also specify the /PAGE qualifier.

/OVERRIDE_FORMAT
/NOOVERRIDE_FORMAT (default)

Controls whether the CDA Viewer overrides the format of your document or uses the formatting information stored in your document.

Note: The /OVERRIDE_FORMAT and /NOOVERRIDE_FORMAT qualifiers are new in VMS Version 5.3.

/PAGE
/NOPAGE (default)

On a character cell terminal, this qualifier determines whether the output display of the CDA Viewer pauses after displaying each page of your file. If you specify the /PAGE qualifier, you can page backward and forward, or jump to the top or bottom of the document. Note that, if you specify the /PAGE qualifier, you cannot specify either the /OUTPUT qualifier or the /INTERFACE=DECWINDOWS qualifier.

/WIDTH=*nn*

Specifies the number of characters per line. If you specify the /OVERRIDE_FORMAT qualifier or if the document being viewed has no inherent format, this page width is used. On the DECwindows display, the default width is 85 characters, which is equivalent to the default page width of 8.5 inches. On character cell displays, the page width defaults to your terminal's screen width. However, if you use the /OUTPUT qualifier, the default is 132 columns.

Note: The /WIDTH qualifier is new in VMS Version 5.3.

Example

```
$
_$
_$
_$
_$
_$
_$
```

This command invokes the CDA Viewer to view a file named FOOBAR.DTIF, which has the DTIF format. The display interface is DECwindows, and the CDA Viewer will override the document's default format. The display width will be 80 characters, and the display height will be 66 lines.

4.10 Additions and Corrections to the *VMS Developer's Guide to VMSINSTAL*

This section lists additions and corrections to the *VMS Developer's Guide to VMSINSTAL*. A note in the margin indicates the VMS operating system version for which the addition or correction applies.

4.10.1 System Disk Directory Structure

V5.0

The *VMS Developer's Guide to VMSINSTAL* does not reflect changes to the system disk directory structure that were made at VMS Version 5.0. Prior to Version 5.0, there were two types of VMS system disks:

- **Common system disks**, which held both private and common system directories
- **Private system disks**, which included only one system root, and did not include a common directory.

With VMS Version 5.0, private system disks are obsolete. All system disks running VMS Version 5.0 or higher are common system disks.

If you must choose an option based on whether your system disk is common or private, use the option listed for a common system disk.

4.10.2 Layered Product Installation

V5.0

Page 1-4 lists the tasks performed by VMSINSTAL when it installs a layered product. Item 4 states the following:

> VMI$KWD is a subdirectory of the system update directory, VMI$ROOT:[SYSUPD].

With VMS Version 5.0, this sentence should read as follows:

> VMI$KWD is a subdirectory of the system-specific update directory, VMI$SPECIFIC:[SYSUPD].

With VMS Version 5.2, this sentence should read as follows:

> VMI$KWD is either a subdirectory of the system-specific update directory, VMI$SPECIFIC:[SYSUPD] or a subdirectory of the device and directory specified by the Alternate Working Device (AWD) option.

4.10.3 Command Line Qualifiers

V5.2

On page 1-7, the description of qualifiers to the VMSINSTAL command line states the following:

> If options are not specified on the command line, VMSINSTAL prompts the installer for this information.

In fact, VMSINSTAL prompts the installer for this information only if VMSINSTAL must also prompt for one of the required parameters, productlist or source.

4.10.4 Get Save Set Option

V5.0

On page 1-8, the description of the get save set (G) option should state the following:

> When save sets are copied using this option, the directory structure originally assigned to files within the save set is not maintained. Because the directory structure is lost, this option cannot be used to copy VMS operating system kits.

4.10.5 Release Notes Option

V5.0

On page 1-8, the description of the release notes (N) option should state the following:

> Release notes to be processed by this option and automatically moved to SYS$HELP must be included in the kit's primary save set (the save set with the file extension A, for example, TEST042.A).

4.10.6 Alternate Root Option

V5.0

On page 1-9, the description of the alternate root (R) option states:

> **R**—The *alternate root* option is used to install the product in a system root other than that of the running system.

This sentence should read as follows:

> **R**—The *alternate root* option is used to install the product on a system disk other than that of the running system.

4.10.7 Booting Option

V5.0

On page 1-9, the description for the booting option should be deleted. The booting option is reserved for Digital use only.

4.10.8 Developer's Options

V5.0

On page 1-10, add the following to the list of developer's options:

I—The *inhibit initial prompts* option is used to suppress the following initial VMSINSTAL prompts:

* Do you want to continue anyway [NO]?
* Are you satisfied with the backup of your system disk [YES]?

4.10.9 VMIMARKER.DAT

V5.0

On pages 1-11 and 4-9, replace "VMIMARKER.DAT" with "VMIMARKERpid.DAT," where pid is the process identification of the installing process.

4.10.10 Referencing Other Products

V5.2

On page 2-4, add the following item to the section titled **Referencing Other Products**:

- If a product requires a minimum version of another product to be installed on the system, use the CHECK_PRODUCT_VERSION callback. See Section 5.5 for more information on this product.

4.10.11 Request Codes

V5.2

Page 3-5 states the following:

> Currently, VMSINSTAL may pass one of two request codes: VMI$_INSTALL to initiate the installation, or VMI$_IVP to initiate the IVP.

Add VMI$_POSTINSTALL to this list of request codes. A description of the VMI$_POSTINSTALL request code follows:

> The VMI$_POSTINSTALL request code initiates work required after the installation is complete. This phase is executed before the IVP phase.

4.10.12 SYSMGR Directory

V5.0

On page 3-6, change [SYSMGR] to [SYS$STARTUP].

4.10.13 KITINSTAL.COM

V5.2

On page 3-7, a sample KITINSTAL.COM includes the following command lines:

```
$
$
```

These command lines should read as follows:

```
$
$
```

4.10.14 CHECKTRAN.COM

V5.0

On page 3-8, replace "CHECKTRAN.COM" with "CHECKTRAN_STARTUP.COM" in the following command line:

```
$
_$
```

On page 3-8, delete the following command lines:

```
$ !
$ !    Add an entry for the product to the startup database
$ !
$ VMI$CALLBACK MODIFY_STARTUP_DB ADD CHECKTRAN.COM LPMAIN "" ""
```

The four deleted lines should be replaced by the following:

```
$ !    Create IVP directory
$ !
$ VMI$CALLBACK CREATE_DIRECTORY COMMON SYSTEST.CHECKTRAN
$ !
$ !    Move IVP files
$ !
$ VMI$CALLBACK PROVIDE_FILE CHECKTRAN_ CHECKTRAN_IVP.COM -
_$ VMI$ROOT:[SYSTEST.CHECKTRAN]
```

The following line is incorrect:

```
$
```

The line should read:

```
$
```

The following command line is incorrect:

```
$
```

The line should read:

```
$
```

4.10.15 VMSINSTAL Steps

V5.2

Page 4-4 incorrectly states that step 4 of VMSINSTAL has no function. A description of step 4 of VMSINSTAL follows:

> In this step, VMSINSTAL sets up the environment for restoring the kit's save sets. In this step, VMSINSTAL checks if the restore save set and pause (RSP) option was specified, and if the specified pause is for one save set or for all save sets. VMSINSTAL also checks if the alternate working device (AWD) option was specified, and, if so, defines the logical name VMI$KWD to point to the specified device.

V5.0

Page 4-10 includes a description of VMSINSTAL step 13. In this description, "restore" should read "install."

4.10.16 ADD_IDENTIFIER Callback

V5.1

On page 5-2, the description of the ADD_IDENTIFIER callback incorrectly names the SET callbacks. Make the following corrections:

- "SET DEVICE_ACL" should read "SET ACL DEVICE."
- "SET DIRECTORY_ACL" should read "SET ACL DIRECTORY."
- "SET FILE_ACL" should read "SET ACL FILE."

4.10.17 ASK Callback

V5.2

Page 5-3 describes the ASK callback. In the paragraph describing the symbol parameter, "uppercase" should read "the required case." Uppercase is the default choice.

In the sentence describing the L option for the the ASK callback, "uppercase" should read "lowercase."

V5.0

Page 5-4 shows the syntax of a typical command line that uses the ASK callback. This command line includes a global symbol named CHECK, which does not adhere to the recommended format for global symbol names. Digital recommends that global symbol names adhere to the following format:

```
fac_symbol
```

The global symbol DIGIT incorrectly omits the prefix indicating the product facility. In this example, the symbol "DIGIT" should correctly be named "CHECKTRAN_DIGIT."

4.10.18 CHECK_NETWORK Callback

V5.0

Page 5-5 shows an example of a command line that uses the CHECK_NETWORK callback. In this example, the symbol "NETSTAT" should be named "CHECKTRAN_NETSTAT."

4.10.19 CHECK_NET_UTILIZATION Callback

V5.0

Page 5-6 shows an example command line that uses the CHECK_NET_UTILIZATION callback. In this example, the symbol "CHECK$" should be named "CHECKTRAN_CHECK."

4.10.20 CHECK_PRODUCT_VERSION Callback

V5.2 Page 5-7 describes the parameters of CHECK_PRODUCT_VERSION callback. This section should state that the symbol parameter is a *global* symbol.

V5.2 Page 5-8 shows an example of a command line that uses the CHECK_PRODUCT_VERSION callback. The command line should read as follows:

```
$
```

4.10.21 CHECK VMS VERSION Callback

V5.0 Page 5-8 describes the CHECK_VMS_VERSION callback. This section incorrectly states the following:

> If the version number of the running system does not meet the specified criteria, the installation is interrupted, and the installer receives an error message that specifies the versions necessary to install the product.

This sentence should read as follows:

> If the version number of the running system does not meet the specified criteria, the product's installation procedure should notify the installer. To do this, use the MESSAGE callback to display a message that specifies the current VMS version of the installing system, and the VMS version necessary to install the product.

V5.0 Page 5-9 describes the return status of the CHECK_VMS_VERSION callback. The section incorrectly states the following:

> This callback returns VMI$_SUCCESS when the version number of the system meets the criteria specified in P3 and P5. Otherwise, it returns VMI$_FAILURE.

The CHECK_VMS_VERSION callback always returns VMI$_SUCCESS.

4.10.22 COMPARE_IMAGE Callback

V5.2 Page 5-10 describes the COMPARE_IMAGE callback. This section should state that the parameter symbol is a *global* parameter.

V5.2 On page 5-10, the description of the file_spec1 and file_spec2 parameters to the COMPARE_IMAGE callback is incomplete. This description should read as follows:

- **file_spec1**—Use this parameter to indicate the full file specification of the image file that is currently on the system.
- **file_spec2**—Use this parameter to indicate the full file specification of the image file that is part of the kit.

V5.0 On page 5-10, the symbol "MY$CHECK" should correctly be named "MY_CHECK." The $ character is reserved for Digital use only.

4.10.23 CONTROL_Y Callback

V5.0

Page 5-10 incorrectly states that the CONTROL_Y callback returns a fatal status. In fact, the CONTROL_Y callback returns a warning status.

4.10.24 NETUAF.DAT

V5.0

On page 5-11 and page 5-40, replace "NETUAF.DAT" with "NETPROXY.DAT." With VMS Version 5.0, the file NETUAF.DAT is obsolete, and has been replaced with the file NETPROXY.DAT.

4.10.25 CREATE_DIRECTORY Callback

V5.0

Page 5-13 describes the SPECIFIC parameter of the CREATE_DIRECTORY callback. Remove the following sentence from this description:

> When you specify SPECIFIC, you can omit the disk designation in the name parameter (P3).

4.10.26 DELETE_FILE Callback

V5.0

Page 5-15 describes the file name parameter to the DELETE_FILE callback. This section should state that wildcard characters may be used in file names. In addition, "VMI$_FAILURE message" should read "VMI$_FAILURE status."

4.10.27 FIND_FILE Callback

V5.0

Page 5-16 describes the locate parameter to the FIND_FILE callback. The following sentence should be deleted from the description of the S value for the locate parameter:

> If this is a small-disk system and the file is not found, the callback checks the corresponding directory on the library disk.

4.10.28 GET_IMAGE Callback

V5.2

Page 5-17 shows an example of a command line that uses the GET_IMAGE_ID callback. In this example, the symbol "MY$IMAGE_ID" should be named "MY_IMAGE_ID."

4.10.29 GET_SYSTEM_PARAMETER Callback

V5.0

Page 5-18 shows an example of a command line that uses the GET_SYSTEM_PARAMETER callback to find the value of the SYSGEN parameter WSMAX. In this example, the symbol "MAX" should correctly be named "CHECKTRAN_MAX."

V5.0 Page 5-18 incorrectly states that the GET_SYSTEM_PARAMETER callback always returns a VMI$_SUCCESS status. In fact, this callback returns a VMI$_FAILURE status if the input file is not found.

4.10.30 MODIFY_STARTUP_DB Callback

V5.0 Pages 5-18 and 5-19 describe the MODIFY_STARTUP_DB callback. This section should be deleted. The MODIFY_STARTUP_DB callback is no longer supported.

4.10.31 MESSAGE Callback

V5.0 Page 5-20 describes the MESSAGE callback. The section describing the text parameter to the MESSAGE callback is incorrect. This section should state that the text parameter is P4, and can specify up to four message lines. In addition, you can use P5, P6, and P7 as quoted parameters for additional message lines.

4.10.32 PROVIDE_FILE Callback

V5.0 Page 5-25 describes the PROVIDE_FILE callback. The description of the destination parameter (P4) should state that this parameter is null if you specify T in the options parameter (P5).

V5.0 Page 5-26 contains an example of the contents of an input file used when you specify the T option in the options parameter (P5). This file should appear as follows:

```
! PROVIDE_FILE data file
TEST_FREDE FREDE.TXT VMI$ROOT:[SYSEXE] C
TEST_FREDC FREDC.COM VMI$ROOT:[SYSUPD] K
TEST_FREDH FREDH.DAT VMI$ROOT:[SYSHLP]
TEST_FREDM FREDM.NTS VMI$ROOT:[SYSLIB]
```

4.10.33 PROVIDE_IMAGE Callback

V5.0 Page 5-27 describes the PROVIDE_IMAGE callback. The description of the destination parameter (P4) should state that this parameter is null if you specify T in the options parameter (P5). In addition, the R option in the description of the options parameter (P5) should be deleted. The R option is no longer supported for the PROVIDE_IMAGE callback.

V5.0 Page 5-28 shows an example of the contents of an input file used when you specify the T option in the options parameter (P5). This file should appear as follows:

```
! PROVIDE_IMAGE data file
TEST_FRED FRED.EXE VMI$ROOT:[SYSEXE] CKE 1,2,3
TEST_RALPH RALPH.EXE VMI$ROOT:[SYSEXE]
TEST_JULIE JULIE.EXE VMI$ROOT:[SYSEXE] KO
```

V5.0

On page 5-28 the first example in the description of the ECO_list parameter should read as follows:

```
$
_$
```

4.10.34 SET IVP Option

V5.0

Page 5-33 shows the command line syntax for the SET IVP option. As printed, the syntax indicates that the keyword parameter (P3) is optional. In fact, the keyword parameter is required. To indicate this, the command line syntax should appear as follows:

```
VMI$CALLBACK SET IVP keyword [help]
```

4.10.35 SET POSTINSTALL Option

V5.2

On page 5-34, the command line syntax for the SET POSTINSTALL option incorrectly omits the keyword parameter. The syntax should appear as follows:

```
VMI$CALLBACK SET POSTINSTALL keyword
```

Use the keyword parameter (P3) to make the appropriate selection from the following list:

- YES—Use this keyword to indicate that you want to call the postinstall phase.
- NO—Use this keyword to indicate that you do not want to call the postinstall phase.

4.10.36 SET PURGE Option

V5.0

On page 5-34, the command line syntax for the SET PURGE options suggests that the keyword parameter is optional. This parameter is required. The correct command line syntax is as follows:

```
VMI$CALLBACK SET PURGE keyword [help]
```

4.10.37 SET REBOOT and SET SAFETY Options

V5.0

On page 5-35, the keyword parameter is required for both the SET REBOOT and SET SAFETY options. The correct command line syntax for these options is as follows:

```
VMI$CALLBACK SET REBOOT keyword
VMI$CALLBACK SET SAFETY keyword [peak]
```

V5.0

Page 5-36 describes the peak parameter for the SET SAFETY option. In this section, replace "insufficient disk space" with "sufficient disk space."

4.10.38 SET STARTUP Option

V5.0

On page 5-36, the syntax of the SET STARTUP option incorrectly omits the parameters option. The correct syntax for specifying this option is as follows:

```
VMI$CALLBACK SET STARTUP filename [parameters]
```

Use the parameters option (P4) to pass parameters to the startup procedure. Separate parameters by a space, and enclose the list in quotation marks (" "). For example, to pass the parameters VMSINSTAL as P1 and DOALL as P2, enter the following command:

```
$
```

4.10.39 Appendix B Corrections

V5.2

On page B-2, correct the following line:

```
$
```

This line should read as follows:

```
$
```

V5.0

On page B-6, replace "VMI$ROOT:[SYSMGR]" with "VMI$ROOT:[SYS$STARTUP]."

4.10.40 Product Registration Information

V5.2

Page E-1 lists an address and telephone number to contact for information about product registration. The following telephone number is incorrect:

(603) 881-1362

The correct telephone number is as follows:

(603) 881-1373

A VMS DECwindows Performance Considerations

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 may ultimately be limited by the amount of workstation memory available for this purpose.

When an application runs on a remote node, many of its performance characteristics reflect those of the remote node. Your application performance depends to a degree on how much memory the remote node has and on how busy the remote CPU and the network are. For example, if the remote node is a relatively fast processor, phases of the

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 into the remote node, run the following command procedure:

```
$
$
```

Run the previous command procedure as a noninteractive detached process, using the following command:

```
$
_$
```

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 may not be optimal for your environment.

AUTOGEN should be run after you have operated your system for a period of time to establish your workload resource profile. This time period is usually a few days. You should invoke AUTOGEN from the system manager's account by entering the following command:

```
$
```

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, GBLSECTIONS, or both, with these additional images installed as shareable images. See the *VMS Install Utility Manual* for further information.

Index

A

Address
  NML address check • 2–15
Analysis back end converter
  processing options • 4–10
AUTOGEN feedback
  new procedure for • 2–15

B

Back end converter
  analysis • 4–10
  PostScript • 4–8
  text • 4–7
  use of • 4–6, 4–12
BATCH/PRINT facility • 1–5

C

Calendar • 1–14
CDA converter library • 4–12
CLUSTER_CONFIG.COM file • 2–15
Console carrier error handling • 1–16
CONVERT/DOCUMENT command • 4–5
  /FORMAT qualifier • 4–5
  /OPTIONS qualifier • 4–6
Corrections to VMS manuals • 4–1
Creating the options file • 4–6

D

DECnet–VAX • 2–14
Data structures
  nonpaged pool • 2–15
DCL commands
  CONVERT/DOCUMENT • 4–5
  VIEW • 4–11
DDIF (Digital Document Interchange Format)
  analyzing files encoded in • 4–10
DDIF input file
  format • 4–12
Debugger • 3–2
DEC$SYLOGIN.COM file • 2–13
DEC$SYLOGIN.TEMPLATE file • 2–13
DECnet TASK object • 2–18
DECterm • 1–8
  controller • 1–9
  fonts • 1–11
  graphics • 1–11
  initialization • 1–9
  keyboards • 1–10
  languages • 1–10
  text • 1–11
  user interface • 1–11
DECwindows
  desktop applications • 1–14
  installing after the VMS Version 5.3 installation • 2–13
  mail • 1–14
  minimum memory • A–1
DECwindows applications
  performance • A–1
  running remotely • A–1, A–2
DECwindows VAXTPU
  incompatible new features of • 1–2
Digital Document Interchange Format
  See DDIF
Documentation • 4–1
Domain converter • 4–10
  processing options • 4–10
Downline task loading • 2–14
DPT$V_NO_IDB_DISPATCH bit • 3–4

E

Error messages
  during mandatory update installation • 2–18
Ethernet Circuit Loopback Tests • 2–1
EVE (Extensible VAX Editor)
  programming
    obsolete keywords for messages constants • 1–3
Extensible VAX Editor
  See EVE

F

FileView • 1–6
/FORMAT qualifier
  for the CONVERT/DOCUMENT command • 4–5
  for the VIEW command • 4–12
Front end converter
  options file • 4–6
  use of • 4–6, 4–12

H

Help widget • 1–14

I

Icon box • 1–6
Initialization
  incompatibility of in DECwindows VAXTPU • 1–2
Initialization procedures
  VAXTPU • 1–2
Initializing unformatted diskettes • 1–2

L

License Management Facility
  See LMF
License Management Utility (LICENSE)
  codes for license types • 2–5
  modifying license units • 2–4
  MOD_UNITS option • 2–4
License Unit Requirement Table (LURT) • 2–5
Linker image-ID field • 4–5
LMF (License Management Facility) • 2–4
LOCKDIRWT parameter • 2–15
LTDRIVER • 2–12

M

$MOUNT system service • 4–4
Magnetic tape files • 1–17
Maintenance operations • 2–14
Math Run-Time Library • 3–2
Memory
  minimum recommended for DECwindows • A–1
MTH$SINCOS routine • 3–1
MTHRTL math library • 3–2

N

NCP (Network Control Program) • 2–15
NETACP$BUFFER_LIMIT logical name • 2–14
NETACP (network ancillary control program) • 2–14
NETACP/NETDRIVER • 2–15
NETCONFIG.COM • 4–2
Network ancillary control program
  See NETACP
Network Control Program
  See NCP
Network Management Listener
  See NML
NML (network management listener)
  check for illegal address • 2–15
  object • 2–15

O

Options file • 4–6
  CDA conversion • 4–7
/OPTIONS qualifier
  for the CONVERT/DOCUMENT command • 4–6

P

Paint • 1–15
PGFIPLHI crash • 2–2
PostScript back end converter
  processing options • 4–8
PPL$TRIGGER_EVENT routine • 3–5
PRINT command
  error conditions • 1–1
PrintScreen • 1–15
Processing options
  analysis back end converter • 4–10
  domain converter • 4–10
  PostScript back end converter • 4–8
  text back end converter • 4–7

Q

Quorum disk processing • 2–1

R

RMS recovery • 1–16
RZ23 disk • 2–12
RZ55 disk • 2–9

S

SCSI
  third-party device support • 3–4
SCSI disk class driver
  disabling the loading of • 2–8
SCSI tape class driver
  disabling the loading of • 2–8
Server • 3–6
Service passwords • 2–15
SET DISPLAY command • 3–7
Small Computer System Interface
  See SCSI
Startup
  incompatibility of DECwindows VAXTPU • 1–2
SUBMIT command
  error conditions • 1–1
SUBMIT/DELETE command • 1–1
SYSGEN
  INVVEC error • 1–16
  LOCKDIRWT parameter • 2–15
SYSLOA • 2–2
SYSMAN • 2–18
System Management Utility
  See SYSMAN

T

Text back end converter
  processing options • 4–7
TIMEDWAIT macro • 3–5

U

UETP (User Environment Test Package) • 2–18
UIL compiler • 3–10
User environment test package
  See UETP
User Interface Language Compiler
  See UIL compiler
UVMTHRTL math library • 3–2

V

VAX Text Processing Utility
  See VAXTPU
VAXTPU (VAX Text Processing Utility)
  incompatibility of DECwindows initialization/startup • 1–2
VIEW command • 4–11
  /FORMAT qualifier • 4–12
VMSD2 system parameter • 2–8
VOLPRO privilege • 1–2

W

Window manager (DECwindows) • 1–6

X

XDDRIVER retransmit timer • 2–14
Xlib Enhancements • 3–5
XMDRIVER (DMR11 powerfail recovery) • 2–2
XUI Toolkit • 3–7

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 *or* U.S. Area Software Supply Business Digital Equipment Corporation Westminster, Massachusetts 01473

[1]For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

VMS
Version 5.3 Release Notes
AA–MG16B–TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

I would like to see more/less _____

_____

_____

What I like best about this manual is _____

_____

_____

What I like least about this manual is _____

_____

_____

I found the following errors in this manual:

Page Description

_____ _____

_____ _____

_____ _____

_____ _____

_____ _____

Additional comments or suggestions to improve this manual:

_____

_____

_____

_____

I am using **Version** _____ of the software this manual describes.

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Phone _____

--- Do Not Tear - Fold Here and Tape ---

digital™

No Postage
Necessary
if Mailed
in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
Corporate User Publications—Spit Brook
ZK01-3/J35 110 SPIT BROOK ROAD
NASHUA, NH 03062-9987

--- Do Not Tear - Fold Here ---

Reader's Comments

VMS
Version 5.3 Release Notes
AA–MG16B–TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

I would like to see more/less _____

_____

_____

What I like best about this manual is _____

_____

_____

What I like least about this manual is _____

_____

_____

I found the following errors in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

_____

_____

_____

_____

I am using **Version** _____ of the software this manual describes.

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Phone _____

-- Do Not Tear - Fold Here and Tape ----------

digital™

No Postage
Necessary
if Mailed
in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
Corporate User Publications—Spit Brook
ZK01-3/J35 110 SPIT BROOK ROAD
NASHUA, NH 03062-9987

-- Do Not Tear - Fold Here ----------

Cut Along Dotted Line