

VMS

---

digital

VMS Version 5.1 Release Notes

# **VMS**

## **Version 5.1 Release Notes**

Order Number: AA-MG16A-TE

### **December 1988**

This document describes the software corrections in Version 5.1 of the VMS operating system. It also describes changes to documentation, software problems, restrictions, and notes that pertain to Version 5.1 of the VMS operating system.

**Revision/Update Information:** This document supplements the Version 5.0 Release Notes.

**Software Version:** VMS Version 5.1

**digital equipment corporation**  
**maynard, massachusetts**

---

**December 1988**

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

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

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

---


© Digital Equipment Corporation 1988.

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	MASSBUS	VAX RMS
DDIF	PrintServer 40	VAXstation
DEC	Q-bus	VMS
DECnet	ReGIS	VT
DECUS	ULTRIX	XUI
DECwindows	UNIBUS	
DIGITAL	VAX	
LN03	VAXcluster	

The following are third-party trademarks:

PostScript is a registered trademark of Adobe Systems, Inc.

X Window System, Version 10 and its derivations (X, X10, X Version 10, X Window System) are trademarks of the Massachusetts Institute of Technology.

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

---

## 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
---------	------

---

<b>CHAPTER 1 VMS VERSION 5.1 RELEASE NOTES</b>	<b>1-1</b>
--	------------

---

1.1	LAYERED PRODUCT CAUTION	1-1
-----	-------------------------	-----

---

<b>CHAPTER 2 GENERAL USER RELEASE NOTES</b>	<b>2-1</b>
---	------------

---

2.1	<b>VMS MAIL UTILITY NOTES</b>	<b>2-1</b>
2.1.1	<b>Mail Utility Folder Name Parameter Now Supports Mixed Cases</b>	<b>2-1</b>
2.1.2	<b>Mailing DDIF Files</b>	<b>2-1</b>
2.1.3	<b>Mail Utility in a Heterogeneous Environment</b>	<b>2-1</b>

---

2.2	<b>VAX RMS JOURNALING DETACHED RECOVERY IMPROVEMENT</b>	<b>2-2</b>
-----	---	------------

---

2.3	<b>OBSOLETE EVE COMMANDS</b>	<b>2-2</b>
-----	------------------------------	------------

---

2.4	<b>USING BACKUP WITH COMPOUND DOCUMENT FILES</b>	<b>2-2</b>
-----	--	------------

---

2.5	<b>DECWINDOWS RELEASE NOTES</b>	<b>2-2</b>
2.5.1	<b>Window Manager Notes</b>	<b>2-2</b>
2.5.2	<b>FileView Notes</b>	<b>2-3</b>
2.5.3	<b>SYSGEN PQL_MPRCLM Parameter and Captive Accounts</b>	<b>2-4</b>
2.5.4	<b>DECterm Release Notes</b>	<b>2-4</b>
2.5.5	<b>Desktop Application Notes</b>	<b>2-6</b>
2.5.5.1	Cardfiler Note • 2-6	
2.5.5.2	Clock Note • 2-6	
2.5.5.3	DDIF Viewer Restrictions • 2-6	
2.5.5.4	EVE Restrictions • 2-7	
2.5.5.5	DECwindows Mail Features • 2-8	
2.5.5.6	DECwindows Mail Problems and Restrictions • 2-9	
2.5.5.7	DECwindows Bookreader Restriction • 2-10	
2.5.6	<b>Running ULTRIX Applications From A Workstation</b>	<b>2-10</b>

## Contents

---

<b>CHAPTER 3</b>	<b>SYSTEM MANAGER RELEASE NOTES</b>	<b>3-1</b>
<hr/>		
<b>3.1</b>	<b>APPLYING THE 5.1 UPDATE TO A 5.0-2A SYSTEM</b>	<b>3-1</b>
<hr/>		
<b>3.2</b>	<b>MOUNT UTILITY—PROBLEM WITH THE AUTOMATIC TAPE-LABELING FEATURE</b>	<b>3-1</b>
<hr/>		
<b>3.3</b>	<b>AUTOGEN—PROBLEM</b>	<b>3-2</b>
<hr/>		
<b>3.4</b>	<b>CLUSTER DISK REAPPEARANCE VERIFIED AFTER MAJOR NODE REBOOT</b>	<b>3-2</b>
<hr/>		
<b>3.5</b>	<b>DUDRIVER AND DSDRIVER CONFIGURE CORRECTION</b>	<b>3-2</b>
<hr/>		
<b>3.6</b>	<b>LATSYM PRINT SYMBIONT CORRECTION</b>	<b>3-2</b>
<hr/>		
<b>3.7</b>	<b>HIGHWATER MARKING RESTRICTION REMOVED</b>	<b>3-3</b>
<hr/>		
<b>3.8</b>	<b>INCORRECT DEVICE REFERENCE COUNT PREVENTS VOLUME FROM BEING MOUNTED</b>	<b>3-3</b>
<hr/>		
<b>3.9</b>	<b>STARTNET.COM-PROBLEM CORRECTED</b>	<b>3-3</b>
<hr/>		
<b>3.10</b>	<b>MANUALLY CONFIGURING Q-BUS DEVICES ON MICROVAX 3400 SERIES SYSTEMS</b>	<b>3-4</b>
<hr/>		
<b>3.11</b>	<b>UETP (USER ENVIRONMENT TEST PACKAGE) MODIFICATIONS</b>	<b>3-4</b>
<hr/>		
<b>3.12</b>	<b>VAX 6200 SERIES COMPUTERS</b>	<b>3-5</b>
<b>3.12.1</b>	<b>Booting VDS from a TK50 _____</b>	<b>3-5</b>
<b>3.12.2</b>	<b>License Unit Requirement Table (LURT) Updated _____</b>	<b>3-5</b>
<b>3.12.3</b>	<b>VAX 6200 Front Panel Fault LED Problem Corrected _____</b>	<b>3-5</b>
<hr/>		
<b>3.13</b>	<b>UPDATING CONSOLE MEDIA FOR VAX 8530, 8550, 8700 OR 8800 COMPUTERS</b>	<b>3-6</b>

3.14	MA780 (MULTIPOINT SHARED MEMORY)	3-7
3.15	PROBLEM BUILDING STANDALONE BACKUP ON RX01 FLOPPY DISKETTES	3-7
3.16	STANDALONE BACKUP KIT NOW REQUIRES FIVE RX50 DISKETTES	3-8
3.17	PSEUDO TERMINAL DRIVER	3-8

---

**CHAPTER 4 PROGRAMMER RELEASE NOTES** 4-1

4.1	RUN <PROCESS> COMMAND CORRECTED	4-1
4.2	RMS STATISTICS RESTRICTIONS	4-1
4.3	RMS \$TRUNCATE SERVICE	4-1
4.4	RTL PPL\$ EXIT HANDLER FUNCTION CORRECTED	4-2
4.5	LIB\$VERIFY_VM_ZONE AND LIB\$SHOW_VM_ZONE ZONE ANALYSIS PROBLEM	4-2
4.6	POOR MAN'S LOCKDOWN	4-3
	LOCK_SYSTEM_PAGES	4-5
	UNLOCK_SYSTEM_PAGES	4-7
4.7	LOCATING NONPAGED SYSTEM PATCH SPACE	4-8
4.8	EXTENSIONS TO DECNET FILE ACCESS PROTOCOL	4-9
4.9	DCL SUBROUTINE ENTRY POINT SCOPING MODIFICATIONS	4-9



## Contents

---

4.10	DECWINDOWS PROGRAMMING NOTES	4-10
4.10.1	DECwindows VAXTPU Notes	4-10
4.10.2	XUI Toolkit Notes	4-11
4.10.3	Xlib Programming Notes	4-12
4.10.4	DECwindows Server and Driver Notes	4-13
4.10.5	Terminal Emulator Notes	4-14
	DECTERM PORT	4-15
4.10.5.1	Terminal Emulator Font Notes •	4-17
4.10.6	VAX C Definition File Requirements	4-17

---

## CHAPTER 5 DOCUMENTATION INFORMATION

---

5-1

---

5.1	VMS VERSION 5.1 DOCUMENTATION NOTE	5-1
5.2	NEW DOCUMENTATION FOR VMS VERSION 5.1	5-1
5.3	<i>VMS VERSION 5.1 NEW FEATURES MANUAL—CORRECTIONS</i>	5-3
5.4	NEW EVE COMMANDS AND FEATURES	5-3
5.4.1	New Commands	5-3
5.4.2	New Features	5-9
5.5	SYSMAN DISKQUOTA REMOVE COMMAND—CORRECTION	5-10
5.6	<i>VAX TEXT PROCESSING UTILITY MANUAL—OMISSIONS</i>	5-10
5.6.1	<i>VAX Text Processing Utility Manual</i> EVE Source File Location	5-11
5.7	<i>VMS RTL LIBRARY (LIB\$) MANUAL—DOCUMENTATION ERROR</i>	5-11
5.8	<i>CORRECTION TO VMS RTL PARALLEL PROCESSING (PPL\$) MANUAL</i>	5-11
5.9	DOCUMENTATION FOR LIB\$SYS_TRNLOG	5-12
	LIB\$SYS_TRNLOG	5-13

5.10	<b>VMS DECWINDOWS TOOLKIT ROUTINES REFERENCE MANUAL—ADDITION</b>	5-16
<hr/>		
5.11	<b>VMS DECWINDOWS XLIB ROUTINES REFERENCE MANUAL—CORRECTIONS</b>	5-16
5.11.1	Corrected X\$DELETE_MODIFIERMAP_ENTRY Routine _____ DELETE MODIFIERMAP ENTRY 5-17	5-16
<hr/>		
5.12	<b>VMS DECWINDOWS USER INTERFACE LANGUAGE REFERENCE MANUAL—CORRECTIONS</b>	5-19
<hr/>		
5.13	<b>VMS COMPOUND DOCUMENT ARCHITECTURE MANUAL—CORRECTIONS</b>	5-19
<hr/>		
5.14	<b>DOCUMENTATION CORRECTIONS FROM THE VMS VERSION 5.0-2 RELEASE NOTES</b>	5-20
5.14.1	<i>VMS Developer's Guide to VMSINSTAL</i> _____	5-20
5.14.2	<i>VMS System Dump Analyzer Utility Manual—Requirements for VIRTUALPAGECNT System Parameter</i> _____	5-20
<hr/>		
5.15	<b>DOCUMENTATION CORRECTIONS FROM THE VMS VERSION 5.0-1 RELEASE NOTES</b>	5-20
5.15.1	<i>Overview of VMS Documentation</i> _____	5-21
5.15.2	<i>VMS Device Support Manual</i> _____	5-21
5.15.3	<i>VMS VAXcluster Manual</i> _____	5-21
<hr/>		
5.16	<b>VMS VERSION 5.0 RELEASE NOTES—CORRECTIONS</b>	5-22
5.16.1	RQDX3 Microcode Revision Level Error _____	5-22
5.16.2	Layered Product Availability _____	5-22

---

**CHAPTER 6 VMS VERSION 5.0-2 AND VERSION 5.0-1 RELEASE NOTES** 6-1

6.1	<b>VMS VERSION 5.0-2 RELEASE NOTES</b>	6-1
6.1.1	<b>VMS Version 5.0-2 General User Release Notes</b> _____	6-1
6.1.1.1	SET ENTRY/REQUEUE Problem Corrected • 6-1	
6.1.1.2	SHOW QUEUE/SUMMARY Now Displays Correct Job Counts • 6-1	
6.1.2	<b>VMS Version 5.0-2 System Manager Release Notes</b> _____	6-1
6.1.2.1	SYSGEN Problem with Non-Workstation Q-bus and UNIBUS Systems Corrected • 6-2	

## Contents

6.1.2.2	VAX 6200 Series Computers—Problems Booting Standalone BACKUP from a TK50 • 6-2	
6.1.2.3	VAXINFO REMOTE—Installation Problem Corrected • 6-3	
6.1.2.4	VMS Batch/Print Facility • 6-3	
6.1.2.4.1	Generic Queue Restriction Now Enforced • 6-3	
6.1.2.4.2	Job Controller Now Deallocates Virtual Memory • 6-3	
<b>6.1.3</b>	<b>VMS Version 5.0-2 Programmer Release Notes</b> _____	<b>6-3</b>
6.1.3.1	ACMS V3.0 Queued Task Facility—Restriction Removed • 6-3	
6.1.3.2	VAX Ada CALENDAR Function—Restrictions Removed • 6-3	
6.1.3.3	VMS RTL—Problems with MTH\$DCOSD and MTH\$DSIND Functions Corrected • 6-4	
<hr/>		
<b>6.2</b>	<b>VMS VERSION 5.0-1 RELEASE NOTES</b>	<b>6-4</b>
<b>6.2.1</b>	<b>VMS Version 5.0-1 General User Release Notes</b> _____	<b>6-4</b>
6.2.1.1	SET ACL Command—Use of the /LIKE Qualifier • 6-4	
6.2.1.2	SET ACCOUNTING Facility • 6-4	
<b>6.2.2</b>	<b>VMS Version 5.0-1 System Manager Release Notes</b> _____	<b>6-4</b>
6.2.2.1	Backup Utility • 6-4	
6.2.2.1.1	Data Overwrite Problem with MOUNT/NOUNLOAD • 6-5	
6.2.2.1.2	Errors Using TU81-Plus Tape Drives • 6-5	
6.2.2.1.3	Errors Using Degaussed Tapes on HSC Drives • 6-5	
6.2.2.2	MSCP Server Problem • 6-5	
6.2.2.3	Print Symbiont Virtual Memory Deallocation • 6-6	
6.2.2.4	TU81-Plus Tape Drives—Data Loss • 6-6	
6.2.2.5	VAX 6200 Series Computers • 6-6	
6.2.2.5.1	LP11 Printer Not Supported when Standalone Backup is Booted from a TK50 on VAX 6210 and 6220 Systems • 6-6	
6.2.2.5.2	Power Failure Support Added • 6-6	
6.2.2.6	VAX Volume Shadowing—Shadow Set Copy Operation Problems • 6-6	
6.2.2.7	DECnet-VAX Session Incompatibility with Phase IV Implementations • 6-7	
6.2.2.8	LMF—New LMF\$CONFIG Command File • 6-8	
6.2.2.9	SYSGEN Problem with Non-Workstation Q-bus and UNIBUS Systems • 6-8	
6.2.2.10	Starting the Queue Manager in SYSTARTUP_V5.COM • 6-9	
6.2.2.11	VAX 6200 Series Computers—Problem Booting Standalone BACKUP Over a CI from a Disk • 6-9	
<b>6.2.3</b>	<b>VMS Version 5.0-1 Programmer Release Notes</b> _____	<b>6-10</b>
6.2.3.1	NCS Utility—Diacriticals Collation Problem • 6-10	
6.2.3.2	RMS \$FREE Restriction • 6-10	
6.2.3.3	VMS RTL • 6-10	
6.2.3.3.1	LIB\$GET_VM Routine Performance Degradation • 6-10	
6.2.3.3.2	MTH\$DCOSD, MTH\$DSIND, and MTH\$DTAND Functions • 6-11	

6.2.3.4	NCS Callable Routines Unavailable • 6-11
6.2.3.5	VMSINSTAL—CHECK_VMS_VERSION Callback Enhancement • 6-11

---

**APPENDIX A VMS DECWINDOWS PERFORMANCE CONSIDERATIONS** A-1

<b>A.1</b>	<b>RECOMMENDED MINIMUM MEMORY CONFIGURATIONS FOR DECWINDOWS</b>	<b>A-1</b>
<b>A.2</b>	<b>RUNNING VMS DECWINDOWS APPLICATIONS REMOTELY</b>	<b>A-1</b>
<b>A.2.1</b>	<b>Suggestion for Running Applications Remotely _____</b>	<b>A-2</b>

---

**APPENDIX B DECWINDOWS ADA PROGRAMMING INTERFACES** B-1

<b>B.1</b>	<b>USING THE PACKAGES</b>	<b>B-2</b>
<b>B.1.1</b>	<b>Package CDA _____</b>	<b>B-3</b>
<b>B.1.2</b>	<b>Package DDIF _____</b>	<b>B-3</b>
<b>B.1.3</b>	<b>Package DWT _____</b>	<b>B-3</b>
<b>B.1.4</b>	<b>Package X _____</b>	<b>B-4</b>
<b>B.2</b>	<b>CALLBACKS</b>	<b>B-5</b>
<b>B.3</b>	<b>TASKING CONSIDERATIONS</b>	<b>B-6</b>
<b>B.4</b>	<b>EXAMPLES</b>	<b>B-6</b>

---

**INDEX**

---

**EXAMPLES**

<b>4-1</b>	<b>Sample Output of Routine LIB\$SHOW_VM_ZONE _____</b>	<b>4-2</b>
------------	---	------------

# Contents

---

## TABLES

1-1	Layered Products Currently Available for VMS Version 5.1	1-2
1-2	Layered Products to be Supported for VMS Version 5.1	1-7
5-1	General VMS Version 5.1 Documentation	5-1
5-2	DECwindows User Documentation Kit Contents	5-2
5-3	Currently Available Products	5-22
5-4	Future Supported Products	5-23
6-1	Devices Affected by SYSGEN Problem	6-2
6-2	Patches for DECnet Compatibility Problems	6-7
6-3	Devices Affected by SYSGEN Problem	6-8
B-1	Subtype Definitions—Package DWT	B-3
B-2	Subtype Definitions—Package X	B-4

---

# Preface

---

## Intended Audience

*VMS Version 5.1 Release Notes* is intended for all system users. Read the release notes before you use your Version 5.1 system.

---

## Document Structure

*VMS Version 5.1 Release Notes* is organized into the following chapters:

- Chapter 1 describes the contents and purpose of the VMS Version 5.1 release.
- Chapter 2 contains release notes intended for general users of the VMS operating system and DECwindows.
- Chapter 3 contains release notes intended for system managers.
- Chapter 4 contains release notes intended for programmers.
- Chapter 5 contains additions and corrections to the VMS Documentation Set.
- Chapter 6 contains the release notes for VMS Version 5.0-2 and Version 5.0-1.
- Appendix A describes the performance implications of running DECwindows applications remotely.
- Appendix B describes the DECwindows Ada programming interface.

---

## Associated Documents

For more information see the VMS Version 5.0 release notes.

---

## Conventions

The following conventions are used in this manual:

mouse	The term <i>mouse</i> is used to refer to any pointing device, such as a mouse, a puck, or a stylus.
MB1, MB2, MB3	MB1 indicates the left mouse button, MB2 indicates the middle mouse button, and MB3 indicates the right mouse button. (The buttons can be redefined by the user.)
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.
<span style="border: 1px solid black; padding: 2px;">Return</span>	A key name is shown enclosed to indicate that you press a key on the keyboard.

## Preface

...	In examples, a horizontal ellipsis indicates one of the following possibilities:
	<ul style="list-style-type: none"><li>• Additional optional arguments in a statement have been omitted.</li><li>• The preceding item or items can be repeated one or more times.</li><li>• Additional parameters, values, or other information can be entered.</li></ul>
.	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 is optional; you can select none, one, or all of the choices.
{}	In format descriptions, braces surround a required choice of options; you must choose one of the options listed.
red ink	Red ink indicates information that you must enter from the keyboard or a screen object that you must choose or click on. For online versions, user input is shown in <b>bold</b> .
<b>boldface text</b>	Boldface text represents the introduction of a new term or the name of an argument, an attribute, or a reason.
<i>italic text</i>	Italic text represents information that can vary in system messages (for example, Internal error <i>number</i> ).
UPPERCASE TEXT	Uppercase letters indicate that you must enter a command (for example, enter OPEN/READ).
UPPERCASE TEXT	Uppercase letters 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

---

## VMS Version 5.1 Release Notes

This documentation describes software enhancements and corrections to Version 5.1 of the VMS operating system. It also describes documentation changes, software problems, restrictions, and notes that pertain to Version 5.1 of the VMS operating system.

Version 5.1 of the VMS operating system is the first major update to Version 5.0 of the VMS operating system. To apply the Version 5.1 update, you must be running Version 5.0-2 or Version 5.0-2A on your system. To get to Version 5.0-2, you must have:

- 1 Installed Version 5.0
- 2 Updated to Version 5.0-1
- 3 Updated to Version 5.0-2

See the *VMS Version 5.1 Installation Guide* for more information about installing Version 5.1. The *VMS Version 5.1 Installation Guide* also contains information about how to upgrade from VMS Version A5.1 to Version 5.1

VMS Version 5.1 contains functional enhancements to VMS to support the DECwindows graphics and windowing system. It also includes support for a new VMS file format for encoding compound text, graphics, and image data. All VMS systems that will be running the new DECwindows software, or that will be exchanging data with systems running the new DECwindows software, must be upgraded with these enhancements. For a more detailed description of these new features, please refer to the *VMS Version 5.1 New Features Manual* and the *Overview of VMS DECwindows*.

VMS commands and utilities, as well as existing application programs that accept input, can now use the text content of DECwindows compound documents. The term **compound documents** refers to files that contain a number of integrated components including text, graphics, and scanned images. The *VMS Version 5.1 New Features Manual* describes VMS support for using the text from DECwindows compound documents that are structured in accordance with the DIGITAL Data Interchange Format (DDIF) specification. Refer to the *VMS Compound Document Architecture Manual* for more information about compound documents.

---

### 1.1 Layered Product Caution

Because of the way the VMS Version 5.1 upgrade procedure is designed, you should not have to reinstall most layered products after the upgrade. However, you must reinstall certain layered products because of product-specific installation procedures. For example, you must reinstall products that create directories synonymous with system directories and products that use data structures defined by VMS. If a product is available (refer to Table 1-1) yet exhibits unexpected behavior once Version 5.1



# VMS Version 5.1 Release Notes

## 1.1 Layered Product Caution

is running, check the appropriate documentation (see footnotes) for layered products restrictions. If problems persist, contact your DIGITAL support representative.

Table 1-1 lists the versions of layered products that are currently supported for VMS Version 5.1. Table 1-2 lists the layered products that will be supported after the release of VMS Version 5.1. Contact your DIGITAL representative for information regarding the availability of the products listed in Table 1-2.

**Table 1-1 Layered Products Currently Available for VMS Version 5.1**

Product Name	Version Number
A-to-Z Base System for MicroVAX	2.6
A-to-Z Business Graphics for MicroVAX	2.6
A-to-Z Database Manager for MicroVAX	2.6
A-to-Z Developer's Kit for MicroVAX	2.6
A-to-Z Electronic Mail for MicroVAX	2.6
A-to-Z Word Processing for MicroVAX	2.6
ALL-IN-1 <sup>1</sup>	2.2
ALL-IN-1/BEV <sup>1</sup>	2.2
ALL-IN-1/CAN-FRANCAIS	2.1
BASEVIEW	1.2
CMR21 Host Utility	1.1
Courseware Design System	1.0
DECDX/VMS	1.1
DECINTACT	1.0
DECnet Router Server	1.2
DECnet/SNA Data Transfer Facility Server	2.0
DECnet/SNA Data Transfer Facility Utilities	2.0
DECnet/SNA Gateway	1.5
DECnet/SNA Gateway for Channel Transport	1.0
DECnet/SNA Gateway for Synchronous Transport	1.0
DECnet/SNA VMS 3270 Data Stream Programming Interface	1.3
DECnet/SNA VMS 3270 Terminal Emulator	1.4
DECnet/SNA VMS APPC/LU6.2 Programming Interface	2.1
DECnet/SNA VMS Application Programming Interface	2.2
DECnet/SNA VMS DISOSS Document Exchange Facility	1.4
DECnet/SNA VMS Gateway Management	2.0
DECnet/SNA VMS Gateway Management for DECSA	1.5

<sup>1</sup>Some restrictions apply. Refer to Section 9.2 of the VMS Version 5.0 Release Notes.

(continued on next page)

# VMS Version 5.1 Release Notes

## 1.1 Layered Product Caution

**Table 1-1 (Cont.) Layered Products Currently Available for VMS Version 5.1**

Product Name	Version Number
DECnet/SNA VMS Printer Emulator	1.1
DECnet/SNA VMS Remote Job Entry	1.3
DECnet-VAX	5.1
DECpage	3.0
DECrouter 200	1.0
DECserver 100	2.0
DECserver 200	2.0
DECserver 500/VMS <sup>2</sup>	1.0
DECvoice	1.0
DRB32 VMS Drivers	2.0
Fortran IV/VAX to RSX	2.8
Ethernet Terminal Server	3.0
High Performance Workstation Software	1.1
IXV11/VMS Driver	2.0
IXV/VAXELN Driver	2.0
KMV1A MicroVAX Driver	2.0
KMV1A MicroVAX Driver and Development Tools	2.0
KMV1A MicroVAX X.25 Link Level Software	2.0
LCG01 Software	1.4
MicroPower/Pascal-VMS	2.4
MicroVAX VSV21 Support Software	3.0
MicroVAX/DRQ3B Device Driver	1.2
MIRA Switch Control/MicroVMS	2.0
MUXserver 100 Remote Terminal Server	2.2
PDP-11 FORTRAN-77/VAX to RSX	5.2
PDP-11 Symbolic Debugger/VAX to RSX	2.0
PLXY-11/VAX	1.4
Remote System Manager Client VMS	2.1
Remote System Manager Server VMS	2.1
Session Support Utility	1.1
Spatial/II	1.1
Terminal Server Manager	1.2
VAXcluster Software	5.1
VAXELN Ada	1.2
VAXELN Toolkit	3.1

<sup>2</sup>Refer to the cover letters for supporting VAX 8800 computers and the VAX 6200. Contact your DIGITAL representative for copies.

(continued on next page)

# VMS Version 5.1 Release Notes

## 1.1 Layered Product Caution

**Table 1–1 (Cont.) Layered Products Currently Available for VMS Version 5.1**

<b>Product Name</b>	<b>Version Number</b>
VAXinfo I	1.4
VAXinfo II	1.4
VAXinfo III	1.4
VAXLINK for IMS and VSAM Client	1.0
VAXset	Release 6
VAXsim	2.1
VAX 3271 Protocol Emulator	2.4
VAX ACMS Product Set <sup>3</sup>	3.0
VAX Ada	1.5
VAX ADE	2.4A
VAX APL	3.1
VAX BASIC <sup>4</sup>	3.2
VAX BCP	1.1
VAX BLISS-32 Implementation Language	4.4
VAX C	2.4
VAX COBOL	4.0
VAX COBOL Generator	1.2
VAX Common Data Dictionary-Plus	4.0
VAX CORAL 66	1.3
VAX DATATRIEVE	4.1
VAX Data Distributor	2.0-1
VAX DBMS	4.0
VAX DECalc	3.0A
VAX DECalc/DECgraph	3.0
VAX DECalc-Plus	3.0A
VAX DECgraph	1.5
VAX DECmail	2.2
VAX DECreporter	2.1
VAX DECSCAN VMS and ELN Bitbus Drivers	2.0
VAX DECslide	1.3
VAX DECspell Verifier/Corrector	1.1
VAX DECTYPE	4.0
VAX DEC/CMS	3.0
VAX DEC/Map	2.0
VAX DEC/MMS	2.3

<sup>3</sup>Requires reinstallation after VMS Version 5.1 has been installed.

<sup>4</sup>Some restrictions apply. Refer to Section 9.3 of the VMS Version 5.0 Release Notes.

(continued on next page)

# VMS Version 5.1 Release Notes

## 1.1 Layered Product Caution

**Table 1–1 (Cont.) Layered Products Currently Available for VMS Version 5.1**

Product Name	Version Number
VAX DEC/Shell	2.1
VAX DEC/Test Manager	2.3
VAX DIBOL	4.0
VAX Distributed Name Service	1.1
VAX Distributed Queuing Service	1.1
VAX DOCUMENT	1.0
VAX DT07	2.1
VAX EDCS	1.1A
VAX FMS	2.3
VAX FORTRAN	5.0
VAX FTAM	1.0
VAX GKS	3.1
VAX Grammar Checker	1.0
VAX KCT32	1.2
VAX KMS11-BD/BE HDLC/BSC Framing Software	1.2
VAX KMS11-BD/BE X.25 Link Level Software	1.3
VAX Language Sensitive Editor	2.2
VAX LIMS/SM	1.3
VAX LISP for VMS Systems	2.2
VAX Message Router	3.1
VAX Message Router Telex Gateway	1.0
VAX Message Router X.400 Gateway	2.1
VAX Message Router/P Gateway	1.1
VAX Message Router/S Gateway	1.1
VAX Notes	1.3
VAX OPS5	2.2
VAX OSI Application Kernel	1.1
VAX OSI Transport Service	2.0
VAX Pascal <sup>5</sup>	3.7
VAX PBX/Facilities Management	3.0
VAX Performance Advisor	1.2
VAX Performance and Coverage Analyzer	2.0
VAX PHIGS	1.0
VAX PL/I	3.1
VAX Printserver 40 Client Software	2.1
VAX Printserver 40 Supporting Host Software	2.1

<sup>5</sup>Some restrictions apply. Refer to Section 4.3.2 of the VMS Version 5.0-2 Release Notes.

(continued on next page)

# VMS Version 5.1 Release Notes

## 1.1 Layered Product Caution

**Table 1-1 (Cont.) Layered Products Currently Available for VMS Version 5.1**

<b>Product Name</b>	<b>Version Number</b>
VAX PSI	4.2
VAX PSI Access	4.2
VAX Rally	2.0
VAX Rdb/VMS	3.0A
VAX REGIS Graphics Library	1.1
VAX REGIS to Sixels Converter	1.0
VAX RMS Journaling	5.1
VAX RPG II	2.1
VAX Scan	1.1
VAX Scriptprinter Software	1.1
VAX Software Project Manager	1.1
VAX Source Code Analyzer	1.2
VAX SPM	3.2
VAX SQL	2.0
VAX TDMS	1.7
VAX TEAMDATA	1.3
VAX TU70/72 Device Driver	1.2
VAX VALU	2.1
VAX Volume Shadowing	5.1
VAX VTX	3.1
VAX Wide Area Network Device Driver	1.0
VAX XWAY	1.1A
VAX-11 RSX	2.4
VAX-11 RTEM <sup>6</sup>	2.3
VAX/PC LAN Server	1.0
VAX/VMS Computer Integrated Telephone Applications Interface	1.1
VAX/VMS Computer Integrated Telephony PBX Server	1.1
VAX/VMS Services for MS-DOS	2.0
VIDA with IDMS/R	2.0
VMS Workstation Software	4.0
VMS/SNA	1.3
VMS/ULTRIX Connection	1.0
VNXSET	Release 5

<sup>6</sup>Refer to the cover letter for RTEM, Version 2.3.

(continued on next page)

# VMS Version 5.1 Release Notes

## 1.1 Layered Product Caution

**Table 1-1 (Cont.) Layered Products Currently Available for VMS Version 5.1**

Product Name	Version Number
WPS-PLUS/VMS	3.0
WPS-PLUS/VMS/BEV	2.1
X25Router 2000	1.0

**Table 1-2 Layered Products to be Supported for VMS Version 5.1**

Product Name
AAF01/VMS Subroutine Library
ADF01/VMS Subroutine Library
American Business Lexicon
American Medical Lexicon
British Lexicon
C.A.S Delivery System
Courseware Authoring System
DECnet/SNA VMS Distributed Host Command Facility
DECrouter 2000
DRX11-C/VMS Driver
EDE with IBM DISOSS
EDE-W Document Exchange
French Lexicon
German Lexicon
IEX-VMS-Driver
LAN Traffic Monitor
NMCC/DECnet ETHERNIM
NMCC/DECnet Monitor
PDP-11 DATATRIVE/VAX
PBX Server
VAX 2780/3780 Protocol Emulator
VAX DAL
VAX DECrad
VAX DECScan VMS Software Tool Kit
VAX Distributed File Service
VAX DSM
VAX DST32 VMS Device Driver
VAX DY32
VAX Key Distribution Center

(continued on next page)

# VMS Version 5.1 Release Notes

## 1.1 Layered Product Caution

**Table 1-2 (Cont.) Layered Products to be Supported for VMS Version 5.1**

<b>Product Name</b>
VAX Public Access Communications
VAX Real-Time Accelerator Software
VAX Remote Bridge Monitor Software
VAX Storage Library System
VAXcluster Console System
VAXinfo Remote
VAXlab Software Library
VSV21 VMS Support Software

---

## 2 General User Release Notes

This chapter contains information about the VMS Version 5.1 operating system that is of interest to general users.

---

### 2.1 VMS Mail Utility Notes

This section contains notes about the VMS Mail Utility.

---

#### 2.1.1 Mail Utility Folder Name Parameter Now Supports Mixed Cases

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

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

---

#### 2.1.2 Mailing DDIF Files

The VMS Mail Utility preserves the Digital Document Interchange Format (DDIF) file tag when DDIF files are mailed between systems running VMS DECwindows. The VMS Mail Utility also preserves the DDIF file tag when you create an output file on a VMS DECwindows system using the EXTRACT Command.

When you read a mail message that is a DDIF file, the VMS Mail Utility outputs only the text portion of the file. Similarly, if you edit a DDIF mail file, you can access only the file text; the output file is a text file that can no longer be used as a DDIF file. However, if you forward a message that consists of a DDIF file, the VMS Mail Utility sends the entire DDIF file, including the DDIF semantics and the DDIF tag, to the addressee.

---

#### 2.1.3 Mail Utility in a Heterogeneous Environment

If you try to send mail messages containing DDIF files to non-VMS systems that do not support tagged files, the VMS Mail Utility returns the NOACCEPTMSG error message, indicating that the remote node cannot accept the message format.



## General User Release Notes

### 2.1 VMS Mail Utility Notes

Similarly, the VMS Mail Utility does not support the mailing of DDIF files to VMS systems that are not running VMS Version 5.1 or later. As with non-VMS systems, the VMS Mail Utility returns the NOACCEPTMSG error message, indicating that the remote node cannot accept the message format.

---

### 2.2 VAX RMS Journaling Detached Recovery Improvement

Prior to VMS Version 5.1, opening a file marked for recovery unit journaling initiated detached recovery. This was a problem for many sites that opened and closed many files per day.

VMS Version 5.1 improves this situation by initiating detached recovery only when RMS determines that recovery is truly required. This has greatly decreased the time needed to open a file marked for recovery unit journaling.

---

### 2.3 Obsolete EVE Commands

Beginning with this version of EVE, the SET SHIFT KEY and SET NOSHIFT KEY commands are obsolete to avoid confusion with the SHIFT key on the main keyboard. The commands do work, although a warning message is displayed. Instead, use the SET GOLD KEY and SET NOGOLD KEY commands respectively. You might want to edit any EVE initialization files to replace "SHIFT" with "GOLD."

---

### 2.4 Using BACKUP with Compound Document Files

Normal usage of BACKUP correctly preserves all file attribute information for compound document (for example, DDIF) files. However, BACKUP/INTERCHANGE fails to preserve the semantics attribute. As a workaround, DDIF files restored from BACKUP/INTERCHANGE savesets can be re-marked as DDIF files using the command:

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

See the *VMS Version 5.1 New Features Manual* for further information on compound document file support.

---

### 2.5 DECwindows Release Notes

The following sections describe release notes applicable to DECwindows.

---

#### 2.5.1 Window Manager Notes

The Window Manager has the following restrictions:

- The Window Manager does not set the WM\_ICON\_SIZE property on the root window.
- If you try to run the Window Manager while you are already running a Window Manager, you receive an error message.

## 2.5.2 FileView Notes

---

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

FileView has the following restrictions:

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

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

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

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

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

## General User Release Notes

### 2.5 DECwindows Release Notes

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

- If you paste a large amount of text into a FileView Task Output box while a text editor is running, the text might appear incorrectly. Press Ctrl/w to correctly display the text.
- If the total length of all the filenames selected in the FileView window (including the device and directory name on each file) exceeds 65535 characters, only a subset of the files are be operated on when a verb is selected from a menu.

### 2.5.3 **SYSGEN PQL\_MPRCLM Parameter and Captive Accounts**

When you AUTOGEN a workstation running DECwindows, PQL\_MPRCLM (the process quota Minimum Process Limit) is set to 8 to allow FileView to function with its sub-processes. Note that this parameter affects only workstations running DECwindows, even in a mixed cluster of workstations and non-workstations.

The *Guide to VMS System Security* recommends that you set the process limit to 0 for a captive account. This prevents a user from accessing DCL when running an application that allows a SPAWN in a captive account. However, Mail no longer allows a SPAWN if the CAPTIVE flag is set in the account record; following this recommendation is unnecessary for captive accounts running Mail only.

If you are setting up a captive account with access to other applications, you should check them to see if SPAWN is allowed.

To override the DECwindows setting, add a "PQL\_MPRCLM = 0" to your SYS\$SYSTEM:MODPARAMS.DAT file and edit the SYS\$MANAGER:DECW\$CHECK\_PARAMS.COM file so that it does not check for the setting of this SYSGEN parameter.

### 2.5.4 **DECterm Release Notes**

The DECterm release notes are as follows:

- DECwindows includes a VT300 series DECterm Terminal Emulator. This terminal emulator can be invoked only through the Session Manager's Create menu or the DECwTermPort() routine. The DECwTermPort routine is described in Section 4.10.5.
- DECterm cannot use the logical name DECW\$USER\_DEFAULTS if it is defined in your LOGIN.COM file. The initial configuration files must reside in your SYS\$LOGIN directory.

# General User Release Notes

## 2.5 DECwindows Release Notes

- To avoid having your DECterms shrink to 80x24 unexpectedly, DIGITAL suggests that systemwide SYS\$SYLOGIN files and user LOGIN.COM files *not* execute the SET TERM/INQUIRE command on DECterms. The DECterm controller tells VMS the proper characteristics and size of DECterms; SET TERM/INQUIRE is unnecessary.

To make login procedures work correctly on both DECterms and non-DECterms, use the following command:

```
$ IF f$getdvi( "SYS$OUTPUT:", "TRM") THEN
  IF .NOT. f$getdvi( "SYS$OUTPUT:", "TT_DECRT3") THEN --
  SET TERM SYS$OUTPUT:/INQ
```

- Caps lock/shift lock is not functioning in the Customize Keyboard menu item. To change between caps lock and shift lock, use the Customize Keyboard menu item in the Session Manager.
- The ReGIS “one-shot interactive position report” ( R(P(I) ) command continues to process input from the host even before a locator button is pressed.

A workaround is to not send a ReGIS String Terminator (ST or ESC) until after your program has read the report from the terminal.

- Attempting to resize a DECterm window before you get a prompt in the window can cause the window to disappear.
- On a 4-plane color system, such as the VS2000/GPX, DECterm might create a private color map for each window in order to emulate a 4-plane VT340 on a 4-plane workstation. DECterm creates the color map when any ReGIS or sixel graphics are displayed in the window; the window manager loads the private color map when that window has the input focus, and it loads the default color map when the window loses the input focus. When the private color map is loaded, other windows are not displayed with their correct colors.

To restore a DECterm window to using the default color map, first clear the window by selecting Clear Display from the Commands menu and then reset the terminal by selecting Reset Terminal from the Commands menu.

- There are various problems when displaying ReGIS or sixel graphics while recording lines off the top. 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.
- Sixel files can over-write the window borders. To restore the borders, iconify and then de-iconify the 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.
- ReGIS character sizes are defined differently than for terminals such as the VT340. The VT340 uses a character size of 8 by 20 pixels, while the ReGIS character size in DECterm depends on the font that

# General User Release Notes

## 2.5 DECwindows Release Notes

is being used. Because of this, text labels are not aligned correctly in applications such as DECgraph and DECslide.

- ReGIS pictures can be scaled incorrectly when the window is resized or when the font size is changed.

---

### 2.5.5 Desktop Application Notes

The following sections provide notes about the DECwindows Desktop Applications.

---

#### 2.5.5.1 Cardfiler Note

The cardfiler generates an X error (X\_ClearArea) the very first time you use the file selection dialog box. This does not affect the functionality of the cardfiler.

---

#### 2.5.5.2 Clock Note

The clock does not correctly notice that you have changed the system time backward while the clock was running. You must exit and rerun the clock if you change the system time.

---

#### 2.5.5.3 DDIF Viewer Restrictions

The DDIF Viewer has the following known restrictions:

- Complex document layout, including page and galley layout, is not supported. Some text display attributes are also not processed when you display a document.
- If, on the command line, you specify a file that contains a graphic that displays in the first window, the Viewer menu bar options and scroll bar arrows do not appear. This problem can be avoided by selecting such files from the Viewer open file selection box. A workaround is to use the resize button to change the window size.
- The Viewer might exit if you open several DDIF files containing graphics in a single session. A workaround is to limit the number of graphics files viewed and reinvoke the Viewer to view more files.
- Graphics displayed on a monochrome screen use a black foreground and white background. If the default user settings include a default dark background, the graphics will probably not be seen until a re-expose; that is, the screen will appear blank because the graphics have been drawn black on black. A workaround is to use the resize button to change the window size.
- If scrolling an image is not possible, it might be represented by an improperly specified DDIF bounding box. A workaround is to use the resize button to change the window size.
- The Compound Document Architecture (CDA) Toolkit requires that documents referred to by DDIF document external references exist in the current default directory if a full, file path name is not specified in the DDIF file external reference description.

# General User Release Notes

## 2.5 DECwindows Release Notes

---

### 2.5.5.4 EVE Restrictions

The following is a list of the known problems or restrictions in DECwindows EVE:

- Journaling and recovery are *not* supported if you invoke EVE with the /DISPLAY=DECWINDOWS qualifier. Therefore, you should save (write out) your edits frequently during the editing session so that, in case of a system failure, you will not lose your work.

If you invoke EVE with /DISPLAY=CHARACTER\_CELL (the default setting), journaling and recovery are provided the same as on a character-cell terminal.

- The ATTACH, SPAWN, and SPELL commands are not supported if you invoke EVE with the /DISPLAY=DECWINDOWS qualifier.

If you invoke EVE with /DISPLAY=CHARACTER\_CELL (the default setting), the ATTACH, SPAWN, and SPELL commands work (although in the case of SPELL, remember that DECspell must be installed on your system).

- Do not use the SET SCROLL MARGINS command on DECwindows, because clicking the mouse outside the scroll margins can produce undesirable effects. You might want to comment out the command in any of your EVE initialization files before using them on DECwindows.
- Scroll bars do not work in HELP. To scroll through lengthy HELP topics, press the NEXT SCREEN and PREV SCREEN keys.
- Mouse operations, such as choosing menu items or using the scroll bars, are *not* recorded in a LEARN sequence.
- The DECwindows clipboard is *not* enabled by default. Copy, cut, and paste operations in EVE use the Insert Here buffer, the same as on character-cell terminals. To enable the clipboard, use the SET CLIPBOARD command. Remember that WPS keypad functions do not use the clipboard, regardless of the setting.
- In certain cases, clicking a mouse button in a scroll margin implemented by the SET (SCROLLING) built-in creates an unintended select range.

If you click the MB1 mouse button while the pointer cursor is in a scroll margin, the EVE editor executes the POSITION (MOUSE) built-in. This causes the VAXTPU screen manager to scroll the text until the editing point is in the main screen region between the scroll margins. However, after performing this scrolling, VAXTPU also performs a select operation on some of the text in the buffer.

At present, VAXTPU and the EVE editor operate under the assumption that you will not click mouse buttons when the pointer cursor is in a scroll margin. If you are designing a layered application, you might want to consider adding a feature to the application to avoid this situation.

For more information about SET (SCROLLING) and an example of its use, see the description of this built-in in the *VAX Text Processing Utility Manual*.

## General User Release Notes

### 2.5 DECwindows Release Notes

- If you specify an ambiguous string while entering a Help topic, the EVE choices window might not display all the topics matching the string you have typed. If the string you entered matches both commands and informational topics, EVE displays all possible commands but none of the informational topics. If the string you entered matches only informational topics, EVE displays all possible informational topics.
- The REPEAT command does not work well in initialization files. Execution of an initialization file stops when the REPEAT command is encountered and EVE prompts for which command to repeat and how many times to repeat it.
- You can get help on a user-written command (assuming you have supplied a user help library) by pressing the DO key, entering the command HELP, entering the name of the command, and pressing the RETURN key. However, if you bind a command to a key using the EVE command DEFINE KEY, you cannot get help on a user-written command (even if you have supplied a user help library) by pressing the HELP key and then pressing the key bound to the command.
- The third parameter to the EVE procedure EVE\$DECLARE\_HELP\_LIBRARY is optional. However, if you do not specify a value for the parameter, you must supply a null string ("") as a placeholder. This restriction will probably be removed in a future version of EVE.

---

#### 2.5.5.5 DECwindows Mail Features

DECwindows Mail shares its mail storage “file cabinet” with the VMS Mail Utility. Messages can be accessed from either utility. However, you perform operations much differently with DECwindows Mail than with the VMS Mail Utility. Experienced VMS Mail users should not expect operations to be performed similarly. The ability to customize the interface for greater compatibility with the VMS Mail Utility is provided through the Modify User Profile item from the Customize pull-down menu.

DECwindows Mail has the following features:

- Most operations from the main window operate on selected messages. Operations that require a selection are dimmed if there is no selected object for them to operate on.
- The first time DECwindows Mail is run, it scans your mail directory searching for mail drawers. Then, the list of drawers are saved in the file DECW\$USER\_DEFAULTS:DECW\$MAIL.DAT. This list is read each time DECwindows Mail is run, bypassing the scan. If you create new drawers using the VMS Mail Utility, you can make them visible to DECwindows Mail using the “Scan mail directory for mail drawers...” or the “Add drawer to index...” menu items from the Maintenance pull-down menu.
- DECwindows Mail fully supports sending, viewing, and extracting DDIF messages. Only the text of a DDIF message can be printed directly. DDIF messages can be converted to another format (such as PostScript) using the Extract function, and the converted file can then be printed (outside of DECwindows Mail).

# General User Release Notes

## 2.5 DECwindows Release Notes

- `SYSSCRATCH` defines the directory where any temporary files are created. If DECwindows Mail is run in a detached process, `SYSSCRATCH` must be explicitly defined in that process.
- DECwindows Mail supports mixed case folder names. When moving or copying using a dialog box, if a folder exists and differs only in case from the specified destination, that folder is used as the destination. New folders are created with the exact case typed in by the user. The only way to create two folders differing only in case is to explicitly create the second folder using Create Folder.

---

### 2.5.5.6 DECwindows Mail Problems and Restrictions

DECwindows Mail has the following known problems and restrictions:

- When you read a mail message while the Auto Refile option is set, the message is immediately moved to the MAIL folder. If you delete or move the message, the action does not take place until the INBOX folder is closed. Therefore, if you read a message, delete it, and open the MAIL folder without first closing the INBOX or NEWMAIL folder, you see the deleted message listed in the MAIL folder. When you eventually close the INBOX or NEWMAIL folder, the deleted message is deleted from the MAIL folder.
- Printing is performed through the standard DECwindows print function. Any print defaults specified in the VMS Mail Utility user profile are not set initially when printing from DECwindows Mail. Any modifications made to the print options using either the Print... menu item from the File pull-down menu or the Modify Print Defaults... menu item from the Customize pull-down menu are remembered as long as DECwindows Mail is active but are not remembered from DECwindows Mail session to DECwindows Mail session.
- When using small icons (the session manager default), the DECwindows Mail icon does not darken when new mail arrives. The icon picture does change to show two overlapping envelopes.
- Delete Drawers... deletes all messages and folders within a drawer but does not delete the underlying mail file itself. To reclaim the disk space used by the drawer and prevent the drawer from being added back to the index by a subsequent scan, delete the file using FileView or DCL.
- When you use a dialog box to move or copy messages selected in the main window, selecting text within the dialog box (which occurs automatically when using TAB to move to a non-empty text field) removes the main window selection. Clicking on OK at this point gives a warning that messages must be selected. If the messages become unselected, click on the main window title bar to restore the selection before clicking on OK in the dialog box.
- System managers should be sure to set `MAIL$SYSTEM_FLAGS` at system startup time as described in Section 9 of the *VMS Mail Utility Manual*.



## General User Release Notes

### 2.5 DECwindows Release Notes

- DECwindows Mail enables deleted-mail purging by default. This is true even if you have set NOAUTO\_PURGE through the VMS Mail Utility.

---

#### 2.5.5.7 DECwindows Bookreader Restriction

The DECwindows Bookreader has the following restrictions:

- For books with large indices, you cannot drag-scroll from the the top of the index to the bottom, or from the bottom to the top, in a single operation. You can get to the bottom or top of the index by using two drag operations or by using other scrollbar functions.
- When the Bookreader starts, it looks for the file DECW\$BOOK:LIBRARY.DECW\$BOOKSHELF. If LIBRARY is defined as a logical name, the Bookreader is not be able to open the library file. To work around this, define a logical, DECW\$BOOKSHELF to be LIBRARY.DECW\$BOOKSHELF before invoking the Bookreader.

---

### 2.5.6 Running ULTRIX Applications From A Workstation

When a DECnet connection initiated by an ULTRIX client is received by a VMS server, the user name associated with the connection is not the ASCII name of the user (for example, jqpublic) but the user identification (UID) number in ASCII form (for example, 517). Using this example, any of the following authorization strings can be entered to the security customization menu to allow jqpublic access to the VMS server (assume the ULTRIX node is ultrix):

```
ultrix::517
ultrix::*
*::517
*::*
```

Note that the strings with wildcard characters could allow other users to connect to the server and are a generally dangerous security practice.

# 3

---

## System Manager Release Notes

This chapter contains information about the VMS Version 5.1 operating system that is of interest to system managers.

---

### 3.1 Applying the 5.1 Update to a 5.0-2A System

The Version 5.1 kit has been constructed so it can be installed on both Version 5.0-2 and Version 5.0-2A systems. When it is installed on a Version 5.0-2A system, some of the patches will have already been applied, so you can ignore any %PATCH-I-ECOSSET and resulting %VMSINSTALL-W-NOFILE messages that you see. If you get warning or error messages on other items in the kit, they should be investigated.

You can also expect “is already updated” messages on some of the files that the update procedure attempts to edit.

---

### 3.2 Mount Utility—Problem with the Automatic Tape-Labeling Feature

The /AUTOMATIC qualifier to the MOUNT command does not work as intended. This qualifier controls the automatic tape-labeling feature (which is enabled by default) and causes the system to generate labels for multiple magnetic tape reels. For example, to mount a magnetic tape, you could enter the following command:

```
$ MOUNT MFA0: FIRST
```

If you use more than one magnetic tape reel, the system expects the next tape to be labeled FIRS02 (the label generated by the automatic tape labeling feature).

Although this feature is intended to allow users to mount a magnetic tape and specify their own volume labels for multiple magnetic tape reels, it does not work correctly. For example, suppose you mount a tape using the following command:

```
$ MOUNT MFA0: FIRST, NEXT
```

The first magnetic tape must be labeled FIRST, but the system ignores the user-specified label for the second reel (NEXT) and expects the label of the second tape to be FIRS02.

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

## System Manager Release Notes

### 3.3 AUTOGEN—Problem

---

#### 3.3 AUTOGEN—Problem

When executing AUTOGEN from the GENPARAMS phase using an old (pre-release) SYS\$SYSTEM:PARAMS.DAT file, AUTOGEN reports two undefined symbols: DEBNA\_CNT and TK\_CNT. If this problem occurs, rerun AUTOGEN from GETDATA or SAVPERAMS, which will create a new version of SYS\$SYSTEM:PARAMS.DAT.

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

---

#### 3.4 Cluster Disk Reappearance Verified After Major Node Reboot

In some Local Area VAXcluster configurations, remote disks do not reappear after a major node reboot. Disks on satellite nodes that are being mount verified return an erroneous “offline” status in response to a “get unit” status command from the rebooting disk. As a result, the rebooting boot node does not create the proper data structure for the disk.

Version 5.1 of the VMS operating system corrects this problem.

---

#### 3.5 DUDRIVER and DSDRIVER CONFIGURE Correction

Changes have been made to the disk class drivers (DUDRIVER and DSDRIVER) and CONFIGURE that improves failover on dual path DSA disks (for example, RA60, RA81) connected to local controllers. A node with a local controller that is accessing a disk through the MSCP server on the other node now discovers its local, secondary path soon after boot and switches to that path if it becomes unreachable through the remote server.

It is still a requirement that local controller disks that are to failover automatically must be mounted on both nodes with local controllers. These changes do not affect the operation of the quorum disk, quorum watch (see *VMS VAXcluster Manual*) or satellite nodes.

---

#### 3.6 LATSYP Print Symbiont Correction

Prior to VMS Version 5.1, the LAT symbiont would allocate an output device, print the requested file, and then deallocate the device. This would occur even if another print job was in the print queue.

Beginning with VMS Version 5.1, this behavior has been modified. Now, every time a print queue that has a LAT printer as a target device is started, LATSYP allocates the LTAX: output device when it begins to print the first file in queue and keeps it allocated across multiple print jobs, until the output queue is stopped. The result is that other applications that assign an output channel to that LTAX: write their output to a temporary disk file, which gets spooled to the LATSYP print queue when the application closes its output channel.

# System Manager Release Notes

## 3.7 Highwater Marking Restriction Removed

---

### 3.7 Highwater Marking Restriction Removed

*VMS Version 5.0 Release Notes* contains an entry (8.17 on page 8-21) stating that your VMS system will fail as a result of memory data page corruption when you use SYSGEN command CREATE to extend the size of the primary paging file SYS\$SYSTEM:PAGEFILE.SYS or the primary swapping file SYS\$SYSTEM:SWAPFILE.SYS when the following conditions exist:

- Highwater marking is enabled on the system volume containing the primary or swap file.
- The system volume has been restored with BACKUP and the highwater mark of the paging file or swap file is incorrect.

This VMS Version 5.0 note points out that you can avoid this problem by temporarily disabling highwater marking on the system volume while using the SYSGEN command CREATE.

This restriction has been removed for VMS Version 5.1. The system initialization process has been enhanced to reset the highwater mark of the paging and the swap file appropriately. As a result, you can enter the SYSGEN command CREATE to extend the paging and the swap file without temporarily disabling highwater marking on the system volume.

---

### 3.8 Incorrect Device Reference Count Prevents Volume From Being Mounted

In previous versions of VMS, beginning with Version 4.2 and prior to 5.1, a problem existed that caused the device reference count to be incorrect.

In a cluster environment, a node shutting down after an incorrect reference count would not be able to mount the device when the node rejoined the cluster. MOUNT would return a "VOLALRMNT, another volume of same label already mounted" error.

This problem mainly occurred on systems that made extensive use of global sections—in particular, systems running ALL-IN-1.

This problem is corrected in VMS Version 5.1.

---

### 3.9 STARTNET.COM-Problem Corrected

VMS Versions 5.0, 5.0-1, and 5.0-2 did not assign logical names necessary for downline loading if the VAX-11 PSI product was installed.

This problem has been corrected in VMS Version 5.1. A new STARTNET.COM file is provided. If you have made site-specific changes to the STARTNET.COM file, you need to apply them to the new version. Your old STARTNET.COM file is copied to STARTNET.PRE\_V51\_COPY to avoid losing your existing edits. You should delete this file when you have edited the new STARTNET.COM file.

## System Manager Release Notes

### 3.10 Manually Configuring Q-bus Devices on MicroVAX 3400 Series Systems

---

#### 3.10 Manually Configuring Q-bus Devices on MicroVAX 3400 Series Systems

When using the `SYSGEN CONNECT` command to manually configure Q-bus devices, note that on MicroVAX/VAXserver 3400 series systems, the Q-bus adapter is not adapter nexus 0. The `SYSGEN SHOW/ADAPTER` command shows the following:

```
SYSGEN> SHOW/ADAPTER

CPU Type:  MicroVAX 3400 Series

Nexus      Generic Name or Description
  0         KA640
  1         UB0
```

The following `CONNECT` command connects a Q-bus device to the KA640 adapter and may result in a system crash:

```
SYSGEN> CONNECT xxA0 /ADAPTER=0 /CSR=%Oyyyyyy -
SYSGEN> /VECTOR=%Ozzz /DRIVERNAME=xxDRIVER
```

Use `/ADAPTER=UB0` to specify the first Q-bus adapter.

---

#### 3.11 UETP (User Environment Test Package) Modifications

The following UETP modifications were made for VMS Version 5.1.

- VMS Version 5.1 updates the `UETSUPDEV.DAT` file in UETP to support the DELQA Ethernet controller.
- VMS Version 5.1 updates the `UETTAPE00` test in UETP to support the RV20 Optical Disc Drive.
- In versions prior to Version 5.1, UETP disabled, by default, the DECnet and the LAT software during its `DEVICE` phase in order to test communication devices. However, due to increasing VMS dependence on DECnet and the LAT, UETP no longer disables DECnet or the LAT.

If DECnet or the LAT software is running during the `DEVICE` phase, the `UETUNAS00` test displays the message:

```
-UETP-W-TEXT, Device is in use by DECnet or another application.
```

Other UETP communication device tests display the message:

```
"-SYSTEM-W-DEVALLOC, device already allocated to another user"
```

This message will be updated to match the message printed by the `UETUNAS00` test in a future release of VMS.

- If you are running UETP from a DECterm window on a VAXstation that is running DECwindows, the `UETTTYS00` test might give the following messages:

# System Manager Release Notes

## 3.11 UETP (User Environment Test Package) Modifications

%UETP-W-TEXT, The process -UETTY500\_000n- returned a final status of:  
%SYSTEM-F-BADPARAM, bad parameter value

and

-UETP-E\_DEUNUS, UETTY500 device TWAn: is unusable, error code=00018272  
-RMS-E-DNR, device not ready, not mounted, or unavailable

This is a known problem and will be fixed in a future release of VMS.

---

### 3.12 VAX 6200 Series Computers

This section contains information relevant to VAX 6200 series computers.

---

#### 3.12.1 Booting VDS from a TK50

When you boot the VAX Diagnostic Supervisor (VDS) on a VAX 6200 series system, VMB.EXE prompts you to mount another tape cartridge. Because VMB.EXE and VDS are contained on a single tape cartridge, you must remount the tape.

The VMS Version 5.1 VMB.EXE does not require you to remount the tape cartridge. This new VMB.EXE will be included in a future version of the VDS software tape cartridge.

---

#### 3.12.2 License Unit Requirement Table (LURT) Updated

VMS Version 5.0-2 included support for VAXservers 6210 and 6220. The following values should be added to Table 6-4 on page 6-128 of the License Management manual:

---

System Model	License Types by Codes					
	VMS				SIP	LP
	A	B	C	D	E	F
VAXserver 6210	NA	1520	NA	NA	50	10
VAXserver 6220	NA	1823	NA	NA	50	10
MicroVAX 3400	60	-1	100	-1	230	100
VAXserver 3400	-1	100	-1	-1	50	10

---

---

#### 3.12.3 VAX 6200 Front Panel Fault LED Problem Corrected

Previous versions of VMS contain a software problem that intermittently caused the front panel fault LED to remain lit on VAX 6200 series processors. VMS Version 5.1 corrects this software problem. The front panel fault LED now lights only when there is a true hardware failure.

## System Manager Release Notes

### 3.13 Updating Console Media for VAX 8530, 8550, 8700 or 8800 Computers

---

## 3.13 Updating Console Media for VAX 8530, 8550, 8700 or 8800 Computers

Complete this step only if you have a VAX 8530, 8550, 8700, or 8800 computer.

**NOTE:** Use the console terminal, DEC Pro CPU, for steps 1 through 13.

You must copy the Version 5.0 VMB.EXE onto your system's console media.

- 1 Place a blank RX50 diskette in one of the diskette drive slots on the console (DEC Pro CPU). The console refers to the upper (or left-hand) diskette drive as DZ1, while the VMS operating system refers to it as CSA1. The console refers to the lower (or right-hand) drive as DZ2, while the VMS operating system refers to it as CSA2.
- 2 Ensure that the diskette drives have been connected to the VMS operating system, as follows:

```
$ SHOW DEVICE CSA
Device          Device      Error      Volume
  Name          Status     Count      Label
CSA0:           (SYSTEM)  Online     0
CSA1:           (SYSTEM)  Online     0
CSA2:           (SYSTEM)  Online     0
```

If the devices are not on line, the following message is displayed:

```
%SYSTEM-W-NOSUCHDEV, no such device available
```

If this message is displayed, enter the following commands (CMKRNL privilege is required):

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> CONNECT CONSOLE
SYSGEN> [CTRL/Z]
```

- 3 Using the VMS operating system, initialize the diskette. Substitute CSA1 or CSA2 for CSAu and substitute a 1- to 12-character name for the volume identification (VOLID) in the following command:

```
$ INITIALIZE/STRUCTURE=1 CSAu: volid
```

- 4 To mount the diskette that you placed in the console's diskette drive, enter the following command, substituting CSA1 or CSA2 for CSAu:

```
$ MOUNT/OVERRIDE=ID CSAu:
```

- 5 To create the required directories on the diskette, enter the following command, substituting CSA1 or CSA2 for CSAu:

```
$ CREATE/DIRECTORY CSAu: [TRANSFER]
```

- 6 To copy the new VMB.EXE to the diskette, enter the following command, substituting CSA1 or CSA2 for CSAu:

```
$ COPY SYS$SYSTEM:VMB.EXE CSAu: [TRANSFER]
```

- 7 Once you have copied all the files, dismount the diskette. Substitute CSA1 or CSA2 for CSAu in the following command:

```
$ DISMOUNT CSAu:
```

## System Manager Release Notes

### 3.13 Updating Console Media for VAX 8530, 8550, 8700 or 8800 Computers

- 8 To enter console mode and then exit from the console program, enter the following commands:

```
$ CTRL/P  
>>> EXIT
```

The dollar sign (\$) prompt should be displayed.

- 9 Open the diskette access door, pause for a moment, and then close the diskette access door. The red indicator light should flash.
- 10 To copy VMB.EXE from the diskette to the fixed drive, enter the following commands:

```
$ COPY DZu:[TRANSFER]VMB.EXE LB0:[CONSOLE]
```

Substitute DZ1 or DZ2 for DZu.

The fixed drive in the console is known by two names, DW2 and LB0. Most console files are stored using the name LB0. If the system displays a message that describes a protection violation on the output device, copy the files using the device name DW2.

- 11 When you have finished copying the files, remove the diskette from the drive.
- 12 Restart the console program by entering the following command:

```
$ RUN CONTROL
```

- 13 Return the console to the VMS operating system by entering the following command:

```
>>> SET TERM PROGRAM
```

---

### 3.14 MA780 (Multiport Shared Memory)

All processors connected to the MA780 must be running the same version of VMS, either Version 4.x or Version 5.x. Running one processor at Version 4.x and another at Version 5.x does *not* work, due to changes in the global section data structure for VMS Version 5.0.

---

### 3.15 Problem Building Standalone BACKUP on RX01 Floppy Diskettes

The standalone BACKUP kit for Version 5.1 does not successfully build on RX01 media. Standalone BACKUP does continue to build on all other supported media (e.g. RA series disks, TK50, RL02, etc.). This problem will be corrected in a future release.

For Version 5.1, if you need to boot standalone BACKUP from RX01 media, you must be certain that you have a bootable copy from Version 5.0, Version 5.0-1 or Version 5.0-2 before proceeding with the Version 5.1 update.

**Note:** This problem applies only to the RX01 media that is the console media for VAX-11/78x series processors. There are no problems building standalone BACKUP on any other media.



## System Manager Release Notes

### 3.15 Problem Building Standalone BACKUP on RX01 Floppy Diskettes

Please refer to the *VMS BACKUP Utility Manual*, which is a part of the extended documentation set, for details on how to build a standalone BACKUP kit.

---

### 3.16 Standalone BACKUP Kit Now Requires Five RX50 Diskettes

The number of blocks in the VMS Version 5.1 standalone BACKUP kit has increased. Five RX50 diskettes are now required to build a standalone BACKUP kit.

---

### 3.17 Pseudo Terminal Driver

VMS Version 5.1 includes a Pseudo Terminal driver. The components of this driver are PYDRIVER and TWDRIVER. This Pseudo Terminal driver is intended for the exclusive use of DECwindows; any other use of these devices is unsupported.

A future release of DECwindows will discontinue the use of this driver, at which point these images will no longer be shipped as part of VMS.

---

## 4 Programmer Release Notes

This chapter contains information about the VMS Version 5.1 operating system that is of interest to programmers.

---

### 4.1 RUN <process> Command Corrected

In previous versions of the VMS operating system, any qualifier to the RUN <process> command that accepted a delta time value as an argument would fail if the time value contained a nonzero number of days *and* a nonzero hours-minutes-seconds entry.

Version 5.1 of the VMS operating system corrects this problem.

---

### 4.2 RMS Statistics Restrictions

The following restrictions apply to the use of RMS Statistics:

- RMS Statistics cannot be gathered on files residing on ODS-1 (On Disk Structure Level 1) disks.
- RMS Statistics are not maintained for process-permanent file accesses. Process-permanent file accesses are those that are not released on image rundown. These are typically accesses resulting from the DCL OPEN command. If a file is accessed both as a process-permanent file and by a user image, then only operations done by the user image are counted in the RMS statistics. Enable or Disable the gathering of RMS statistics with the SET FILE/[NO]STATISTICS command.

---

### 4.3 RMS \$TRUNCATE Service

The RMS \$TRUNCATE service is now sensitive to record-access mode.

In sequential record-access mode, you can use this service only immediately after setting the context of the current record by successfully executing a Get or a Find service.

In random-access-by-key mode, VMS RMS establishes the current record position as defined by the key of reference or by the relative record number, as applicable.

In random-access-by-RFA (Record File Address) mode, VMS RMS establishes the current record position as defined by the RFA.

## Programmer Release Notes

### 4.4 RTL PPL\$ Exit Handler Function Corrected

---

#### 4.4 RTL PPL\$ Exit Handler Function Corrected

In previous versions of the VMS operating system, when the RTL Parallel Processing Library (PPL\$) facility was used in conjunction with FORTRAN/PARALLEL, it did not allow a FORTRAN/PARALLEL subprocess to properly terminate. This was due to the PPL\$ Exit handler's inability to stop checking for subprocesses that had once joined but since left the PPL\$ application.

Version 5.1 of the VMS operating system corrects this problem.

---

#### 4.5 LIB\$VERIFY\_VM\_ZONE and LIB\$SHOW\_VM\_ZONE Zone Analysis Problem

The routines LIB\$VERIFY\_VM\_ZONE and LIB\$SHOW\_VM\_ZONE can, under specific conditions, incorrectly determine that a virtual memory zone is corrupted.

If a program causes a zone to have one or more 8-byte blocks of free memory, the routine LIB\$VERIFY\_VM\_ZONE incorrectly returns the status LIB\$\_BADZONE. In the same situation, LIB\$SHOW\_VM\_ZONE reports that the area free list is corrupted with an invalid block size.

A sample of the incorrect output from LIB\$SHOW\_VM\_ZONE is shown in Example 4-1. Note that it is the zone analysis that is incorrect; the memory zone itself is not corrupted.

---

##### Example 4-1 Sample Output of Routine LIB\$SHOW\_VM\_ZONE

---

```
Zone ID = 00073600, Zone name = ""
Algorithm = LIB$K_VM_FIRST_FIT
Flags = 00000000
Initial size = 16 pages      Current size = 16 pages in 1 area
Extend size = 16 pages      Page limit = None

Requests are rounded up to a multiple of 8 bytes,
naturally aligned on 8 byte boundaries

8 bytes have been freed and not yet reallocated

144 bytes are used for zone and area control blocks, or 1.7% overhead

Area Summary:
      First      Last      Pages      Bytes not yet
      address   address  assigned  allocated
      -----  -
00073E00 00075DFF      16      8184

Scanning Free List for Area at 00073E00
**** ERROR -- invalid block size ****
```

---

(continued on next page)

4.5 LIB\$VERIFY\_VM\_ZONE and LIB\$SHOW\_VM\_ZONE Zone Analysis Problem

Example 4-1 (Cont.) Sample Output of Routine LIB\$SHOW\_VM\_ZONE

Link Analysis for Current Block:

	Previous	Current	Next
	-----	-----	-----
Block adr :	00062EF0	00073E00	00062EF0
Forw link (abs):	00073E00	00062EF0	00073E00

Block size = 8192

Block contents:

```

❶ 00000000 00000000 00062EF0 00000008 ....?..... 00000 00073E00
   00000000 00000000 00000000 00000000 ..... 00010 00073E10
(510 matching lines skipped)

```

- ❶ The key to recognizing that this is the known problem is the value 00000008 in the first longword of the block dump.

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

4.6 Poor Man's Lockdown

Certain privileged code, written prior to VMS Version 5.0, utilizes a technique, commonly known as "poor man's lockdown," whereby one or two pages of code are locked into a process or system working set as a side-effect of elevating IPL. Such code has one of the following forms:

```

ASSUME 10$-. LE 511 ; Check for contiguity of pages
SETIPL 10$
.
.
.
;Code to be locked into memory
.
.
.
RSB
10$: .LONG IPL$xxxx

```

The effect of this coding technique is that, because the system must determine the value of the argument to the SETIPL macro from location 10\$, it must fault into memory the page in which 10\$ resides. As a result, before the code actually elevates IPL, the pages in which the SETIPL macro and 10\$ reside will become memory-resident. In this way, the code can avoid a page fault while executing the code between the SETIPL and 10\$ at elevated IPL. The ASSUME macro guarantees that the pages to be faulted are contiguous.

This technique has several limitations:

- It cannot lock more than two virtually contiguous pages.
- Beginning with VMS Version 5.0, it is only useful in locking per-process pages, not system pages. In a VMS multiprocessing system, a page in the system working set could be faulted in by one processor, only to be removed from the system working set by another processor.

## Programmer Release Notes

### 4.6 Poor Man's Lockdown

To lock system pages, you must use the `LOCK_SYSTEM_PAGES` and `UNLOCK_SYSTEM_PAGES` macros as described in the next section. (Note that you cannot use these macros to lock per-process pages in memory.)

- Prior to VMS Version 5.0, IPLs were the means by which system tasks were prioritized and access to system data was synchronized. Code executing at an elevated IPL would effectively block other code in the system from executing at or below that IPL. In VMS Version 5.0, which introduced symmetric multiprocessing, and later versions of the operating system, merely raising IPL does not synchronize systemwide activity or enforce orderly access to data.

Sometimes it might be necessary only to block tasks or synchronize activity on the local processor. In these instances, raising IPL provides sufficient synchronization and “poor man's lockdown” behaves as it did before VMS Version 5.0. For instance, use of “poor man's lockdown” to lock a code segment executing at `IPL$_RESCHED` effectively prevents process deletion and rescheduling while the code executes at nonpageable IPL.

However, if a locked code segment must access system data structures at an elevated IPL—for instance, at `IPL$_SYNCH`—it must obtain the spin lock associated with the database by using one of the spin lock synchronization macros (`LOCK`, `FORKLOCK`, or `DEVICELOCK`). After accessing the data, it must release the acquired spinlock by invoking `UNLOCK`, `FORKUNLOCK`, or `DEVICEUNLOCK`. Appendix B of the *VMS Device Support Manual* discusses the spin lock synchronization macros.

---

## LOCK\_SYSTEM\_PAGES

Locks a paged code segment in system memory.

---

**FORMAT**            **LOCK\_SYSTEM\_PAGES** [*startva*],*endva* [*ipl*]

---

**PARAMETERS**    **[*startva*]**  
System virtual address in the first page to be locked. If the **startva** argument is omitted, the starting virtual address defaults to the current PC.

**[*endva*]**  
System virtual address in the last page to be locked.

**[*ipl*]**  
IPL at which the locked code segment is to execute. If the **ipl** argument is omitted, the locked code segment executes at the current IPL.

---

**DESCRIPTION**    The LOCK\_SYSTEM\_PAGES macro calls a memory management routine to lock as many pages as necessary into the system working set. The macro accepts a virtual address that indicates the first page to be locked and a virtual address that indicates the last page to be locked. You can also supply the IPL at which the code in the locked pages is to execute.

The LOCK\_SYSTEM\_PAGES macro executes under the following conditions:

- The LOCK\_SYSTEM\_PAGES macro should be used only on system virtual addresses.
- All pages requested in a single LOCK\_SYSTEM\_PAGES macro call must be virtually contiguous. If you must lock discontinuous memory, you must invoke the LOCK\_SYSTEM\_PAGES macro once for each page or set of contiguous pages.
- You must invoke LOCK\_SYSTEM\_PAGES at IPL 2 or lower to allow pagefaulting to occur.
- When the locked code segment is finished, it must invoke the UNLOCK\_SYSTEM\_PAGES macro to release all previously locked pages. In other words, there must be exactly one UNLOCK\_SYSTEM\_PAGES macro call per LOCK\_SYSTEM\_PAGES macro call.
- When it invokes the UNLOCK\_SYSTEM\_PAGES macro, the code must ensure that the stack is exactly as it was when the LOCK\_SYSTEM\_PAGES macro was invoked. That is, if the code has pushed anything on the stack, it must remove it before invoking UNLOCK\_SYSTEM\_PAGES.

## VMS Macros Invoked by Drivers

### LOCK\_SYSTEM\_PAGES

- If the **ipl** argument is supplied to the LOCK\_SYSTEM\_PAGES macro, the locked code segment must invoke the appropriate system synchronization macros (LOCK, FORKLOCK, or DEVICELOCK and UNLOCK, FORKUNLOCK or DEVICEUNLOCK) to obtain and release any spin locks required to protect the resources accessed at the elevated IPL.
- If it specified the **ipl** argument to the LOCK\_SYSTEM\_PAGES macro, the code segment must restore the previous IPL, either explicitly, through the use of the **ipl** argument to the UNLOCK\_SYSTEM\_PAGES macro, or through the use of one of the system synchronization macros.

---

### EXAMPLE

```
30$:          TSTB      (R0)                ; Fault in page
              LOCK_SYSTEM_PAGES, -
              END=100$                    ; Lock down pages
              LOCK     LOCKNAME=MMG, -    ; Synch with MMG
              SAVIPL=- (SP)              ; Save current IPL
              MOVL     W^MMG$GL_SYSPHD, R3 ; Get system PHD
              .
              .
              .

              UNLOCK  LOCKNAME=MMG, -    ; Unlock MMG
              NEWIPL= (SP) +             ; Restore IPL
              UNLOCK_SYSTEM_PAGES        ; Unlock pages
100$:
```

In this example, the LOCK\_SYSTEM\_PAGES macro locks all pages between labels 30\$ and 100\$ into the system working set. The UNLOCK\_SYSTEM\_PAGES macro does the co-routine return to unlock those pages locked by the LOCK\_SYSTEM\_PAGES macro call.

---

## UNLOCK\_SYSTEM\_PAGES

Terminates a request to lock down a series of system pages.

---

**FORMAT**            **UNLOCK\_SYSTEM\_PAGES** *[ipl]*

---

**PARAMETERS**    *[ipl]*  
IPL at which to continue execution.

---

**DESCRIPTION**    The UNLOCK\_SYSTEM\_PAGES macro terminates a request to lock down a series of contiguous system pages. In a code segment that uses this locking technique, there must be exactly one UNLOCK\_SYSTEM\_PAGES macro call per LOCK\_SYSTEM\_PAGES macro call. When the locked code segment completes, it must invoke the UNLOCK\_SYSTEM\_PAGES macro to release all previously locked pages.

The UNLOCK\_SYSTEM\_PAGES macro executes under the following conditions:

- When it invokes the UNLOCK\_SYSTEM\_PAGES macro, the code must ensure that the stack is exactly as it was when the LOCK\_SYSTEM\_PAGES macro was invoked. That is, if the code has pushed anything on the stack, it must remove it before invoking UNLOCK\_SYSTEM\_PAGES.
- If it specified the **ipl** argument to the LOCK\_SYSTEM\_PAGES macro, the code segment must restore the previous IPL, either explicitly, through the use of the **ipl** argument to the UNLOCK\_SYSTEM\_PAGES macro, or through the use of one of the system synchronization macros (UNLOCK, FORKUNLOCK or DEVICEUNLOCK). If it lowers IPL, the locked code segment must invoke the appropriate system synchronization macro to release any spin locks that were required to protect the resources accessed at the elevated IPL.



# Programmer Release Notes

## 4.7 Locating Nonpaged System Patch Space

---

### 4.7 Locating Nonpaged System Patch Space

When using the XDELTA debugging tool to debug a device driver not supplied by DIGITAL, a system programmer might need to construct a complex breakpoint or store an XDELTA command string in nonpaged system patch space. Sections 16.10.6.5 and 16.10.8 of the VMS Version 5.0 edition of the *VMS Device Support Manual* discuss these needs, but contain outdated information about how the programmer can locate this space.

VMS Version 5.0 introduced a change in the composition of the VMS executive image. No longer does the VMS executive comprise a single, large, static executive image, but rather a set of vectors and a set of independently loadable images. The system map file (SYS\$SYSTEM:SYS.MAP) does not contain information useful in determining the location of local symbols within these loadable images.

Each of the loadable images of the executive contains an area reserved as nonpaged system patch space. In each loadable image, the symbol PAT\$A\_NONPAGED contains a descriptor that identifies the location and size of the unused nonpaged system patch space in that image. This descriptor has the following form:

```
PAT$A_NONPAGED::  
    .LONG      size-in-bytes  
    .LONG      offset to patch-space-start-address
```

A process having suitable privileges can access unused system patch space in any of the loadable images of the VMS executive. For instance, a system programmer debugging a device driver can deposit an XDELTA command string in the nonpaged system patch space of any of the loadable images.

To determine the size of patch space and its starting address in any given loadable image of the VMS executive, perform the following steps:

- 1 Enter the following commands to display a list of all images of the VMS executive that have been loaded into memory:

```
$ ANALYZE/SYSTEM  
SDA> SHOW EXECUTIVE  
SDA> EXIT
```

The XDELTA command ;L also displays a list of the loaded images.

- 2 Note the base address of the image whose patch space you want to use.

For example, you might have selected PROCESS\_MANAGEMENT and determined its base address to be 80127C00<sub>16</sub>.

- 3 Determine the image value of the nonpaged system patch space descriptor (PAT\$A\_NONPAGED) in the selected image.

For example, to determine the image value of PAT\$A\_NONPAGED in PROCESS\_MANAGEMENT, enter the following commands from the DCL prompt:

```
$ ANALYZE/IMAGE/OUT=TEMP.DAT SYS$LOADABLE_IMAGES:PROCESS_MANAGEMENT.EXE  
$ SEARCH/WIND TEMP.DAT PAT$A_NONPAGED
```

## Programmer Release Notes

### 4.7 Locating Nonpaged System Patch Space

For example, you might have determined the image value of `PAT$A_NONPAGED` in `SYS$LOADABLE_IMAGES:PROCESS_MANAGEMENT.EXE` to be  $6544_{16}$ .

- 4 Using the Patch Utility, locate `PAT$A_NONPAGED` in the image and examine its contents.

For example, the following session locates and examines `PAT$A_NONPAGED` in `SYS$LOADABLE_IMAGES:PROCESS_MANAGEMENT.EXE`:

```
$ PATCH/NOJOURNAL SYS$LOADABLE_IMAGES:PROCESS_MANAGEMENT.EXE
PATCH> EX 6544
00006544: 00000077
PATCH> EX
00006548: 00000F85
PATCH> EXIT
```

In this example, the Patch Utility output shows that there are  $77_{16}$  bytes remaining in the nonpaged system patch space of `PROCESS_MANAGEMENT` and that the available patch space starts at offset  $F85_{16}$  into the image.

- 5 Calculate the starting address of nonpaged system patch space in the selected loadable executive image by adding the offset from the descriptor to the base address of the image you determined in Step 2.

For instance, the base address of nonpaged system patch space in `SYS$LOADABLE_IMAGES:PROCESS_MANAGEMENT.EXE` is  $80127C00_{16} + F85_{16}$ , or  $80128B85_{16}$ .

---

## 4.8 Extensions to DECnet File Access Protocol

The DECnet file access protocol (DAP) support in VMS has been enhanced to properly handle compound document (for example, DDIF) files. This support is fully upward compatible with earlier versions of the DAP protocol. One enhancement has extended the actual length of the `SYSCAP` field in the `CONFIG` message. The field is still below the defined maximum field length as defined in earlier versions of the DAP protocol; therefore, the change is compatible with implementations conforming to DAP Versions 5.6 and later.

---

## 4.9 DCL Subroutine Entry Point Scoping Modifications

To make the scoping of subroutine entry points more intuitive, the following restrictions will be added to a future release of VMS:

- Subroutine entry points that are defined within another subroutine are local to that subroutine. You cannot call a subroutine if the subroutine entry point is within a separate subroutine block. For example, the following call will **not** work in a future release of VMS:

## Programmer Release Notes

### 4.9 DCL Subroutine Entry Point Scoping Modifications

```
$ CALL BAR
$
$ MAIN: SUBROUTINE
$
$     BAR: SUBROUTINE
$     ENDSUBROUTINE
$
$ ENDSUBROUTINE
```

The following call **will** work because BAR is defined within MAIN:

```
$ MAIN: SUBROUTINE
$
$     BAR: SUBROUTINE
$     ENDSUBROUTINE
$
$ CALL BAR
$ ENDSUBROUTINE
```

- If a subroutine entry point is located within an IF-THEN-ELSE block, you cannot call this subroutine from outside the IF-THEN-ELSE block. For example, the following call will not be allowed in a future release of VMS:

```
$ IF 1
$ THEN
$     FOO:SUBROUTINE
$     ENDSUBROUTINE
$ ENDIF
$ CALL FOO
```

- Every SUBROUTINE command must have a matching ENDSUBROUTINE command to delimit the subroutine. This is not a new restriction, but it will be more strictly enforced.

In the following example, the entry point for subroutine B is defined within subroutine A because there is not an ENDSUBROUTINE to delimit A (i.e. the EXIT is not a delimiter of A). Therefore, subroutine B is inaccessible from outside subroutine A.

```
$ A: SUBROUTINE
$ EXIT
$
$ B: SUBROUTINE
$ ENDSUBROUTINE
```

---

## 4.10 DECwindows Programming Notes

This section contains release notes applicable to VMS DECwindows programmers.

---

### 4.10.1 DECwindows VAXTPU Notes

The following list describes DECwindows VAXTPU architectural limitations and problems.

- Do not set the value of any widget's AUTO\_UNMANAGE resource using the XUI User Interface Language, VAXTPU, any of the XUI Toolkit procedures, or any other means. VAXTPU handles the input

## Programmer Release Notes

### 4.10 DECwindows Programming Notes

focus in unpredictable ways when this value is set. Use the VAXTPU built-in UNMANAGE\_WIDGET instead.

- If your layered application includes popup menus, the application should create them at startup time using the CREATE\_WIDGET built-in. Do not wait until the user has performed the action invoking the menu (such as a click of a mouse button) to create the popup menu widget. If you do so, the first invocation (for example, the first click of the mouse button) does not allow the user to choose any of the menu items. The menu becomes useable only after subsequent invocations.
- In the DECwindows version of VAXTPU, the READ\_CHAR built-in cannot read a keypad key or a function key. If a VAXTPU procedure uses READ\_CHAR and the user presses a keypad or function key, READ\_CHAR returns a null string and signals the warning TPU\$\_NOCHARREAD.

For more information about the READ\_CHAR built-in, see the *VAX Text Processing Utility Manual*.

- In the DECwindows version of VAXTPU, the SET (AUTO\_REPEAT) built-in procedure has no effect on whether VAXTPU repeats keystrokes when you hold down a key. On DECwindows, SET (AUTO\_REPEAT) is disabled. On character-cell terminals, the behavior of SET (AUTO\_REPEAT) is unchanged.

For more information on SET (AUTO\_REPEAT), see the *VAX Text Processing Utility Manual*.

- If you invoke the DECwindows version of VAXTPU Version 2.2 through its callable interface (for example, from the MAIL utility) more than once during a session, the results are unpredictable. Possible results include error messages from the XUI Toolkit or an access violation.

This problem will be fixed in a future release.

- If a DECwindows application sends a request for the LINE\_NUMBER of a global selection to the DECwindows version of VAXTPU (Version 2.2), VAXTPU does not respond to the request with information of data type SPAN. In Version 2.2, VAXTPU responds with information of data type INTEGER.

This problem will be fixed in a future release.

---

#### 4.10.2 XUI Toolkit Notes

The following information describes specific features of the XUI Toolkit:

- **Help widget**—The Help widget does not support XtSetValues of many text resources. This will be corrected in a future release.
- **Unseen LeaveWindow events**—There is a problem with widgets that pop up spring loaded 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 Intrinsic event dispatching mechanism.

## Programmer Release Notes

### 4.10 DECwindows Programming Notes

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 highlights. When MB2 is pressed, the popup menu is displayed. A LeaveWindow event should be dispatched to un-highlight the widget as the pointer is moved into the popup menu. This LeaveWindow event is not delivered and the widget is left in the highlighted state when the menu pops down.

This 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 will be fixed in a future release.

- **Attached Dialog Box**—A call to XtSetValues on the attachments of a child of an attached dialog box can result in an infinite loop. Attachments should be set up when the child is created. In most cases this problem can also be avoided by performing the XtSetValues while the widget is unmanaged.
- **Listbox Widget**—The Shift/MB1 double-click for the extend-confirm callback is not supported. This will be fixed in a future release.
- **Cut and Paste**—The cut and paste routines do not work for transferring data between applications running on host machines having different byte orders. This will be fixed in a future release.

---

#### 4.10.3 Xlib Programming Notes

The following restrictions apply to the Xlib programming library routines.

- The PUT IMAGE routine does not implement image format conversions for ZPixmap images with a depth greater than one if the depth does not match the depth of the server.
- If you call the CIRCULATE SUBWINDOWS and CIRCULATE SUBWINDOWS DOWN routines with a **direction** argument of Lower Highest, the routines do not circulate the subwindows if the highest mapped child that occludes other children is not completely visible.

One instance of this case is when the highest mapped child that occludes other windows is clipped to the parent window. Although the child is not occluded, it is not totally visible.

Additionally, if a totally visible window is found, lowering it to the bottom of the window stack can result in screen corruption.

## Programmer Release Notes

### 4.10 DECwindows Programming Notes

A work around for this problem is to use the CONFIGURE WINDOW routine to alter the stacking order of the child windows. In addition to providing an alternative method for changing windows stacking order, CONFIGURE WINDOW does not expend time finding the highest occluding visible window.

#### 4.10.4 DECwindows Server and Driver Notes

The DECwindows server is DIGITAL's implementation of the X Window System's server. The server is the component of the architecture that allows application interfaces to look the same on all supported systems.

Device drivers process requests from the server to the display and from the input devices to the server.

VAXstation configurations are subject to the following notes:

- On a VAXstation 2000, The keyboard and mouse serial lines are TTA0 and TTA1, respectively. Terminal operations such as SET/SHOW TERM, or SET HOST DTE do not work for these devices. The terminal lines are TTA2 and TTA3.
- The server looks for a number at the end of its process name. If it finds a number, it considers that number to be its server number and listens for connections on that number rather than on 0. The default value is 0. This is normally resolved by the DECwindows startup command files.
- You cannot depend on the values of BlackPixel and WhitePixel being 0 and 1. Their values will differ depending on the hardware.
- Put Image is restricted to a maximum width of 1024 pixels for GPX servers.
- The X11 protocol allows the server to "arbitrarily transform" the components of a cursor in order to meet the requirements of the display. Since neither the VAXstation 2000 nor the VAXstation II monochrome workstations support recoloring cursors, you should not expect the colors you specify for the cursor to actually be reproduced on the hardware.
- The DECwindows server contains a facility called a **condition handler** that detects problems that might otherwise cause the server to stop and tries to let the server continue. When the condition handler intercepts a problem like this, it either sends an "Implementation" error to the client, disconnects the client, or both.

When the condition handler recovers from an error, the server might lose resources such as memory. Therefore, after a number of these interceptions, the condition handler broadcasts a warning message to all users on the workstation indicating that the server may be running in a degraded mode and suggesting that it be restarted. If you get messages like this, you should restart the server at the next convenient opportunity. Enter the following command from a privileged account (it may be one logged into a DECwindows terminal emulator window) to restart the server:

## Programmer Release Notes

### 4.10 DECwindows Programming Notes

```
$_SYS$MANAGER:DECW$STARTUP RESTART
```

- There are a number of X11 protocol requests and corresponding XLIB requests that take an unsigned word (“short” in C) as a width or height argument. A common application mistake is to calculate a width or height incorrectly and end up with a small negative number. The protocol interprets this as a large unsigned number. The DECwindows server does not deal with large widths and heights correctly in many cases. You may get an “Implementation” error returned by the condition handler, or by the server directly detecting that the number is too large.

Note that the numbers that the server has trouble with are generally greater than 32767, or combinations of coordinates and width/height that add up to greater than 32767. Coordinates in the range of any supported display devices are much smaller than this number.

The server acts in an unfriendly way when a client does not read its events often enough. After the server runs out of buffer space when trying to write events, errors, or replies to a client, it hangs in HIB state and retries the write to the client periodically. The server does not service other clients while it is in this state.

After a timeout period, the server disconnects the offending client and services the remaining clients. This problem can happen when a client is working slowly and generating a lot of requests. However, the most common occurrence is when the client is being debugged. Future releases of the server will deal more gracefully with this problem. In the meantime, there is little possibility of a workaround, except to process events as often as possible.

- Under some circumstances the XCopyGC routine causes the server’s memory use to grow slightly. If you do a large number of XCopyGC requests, the server gets larger and slower until it starts returning “Implementation” errors to the client. There is no known workaround other than to restart the server.

---

#### 4.10.5 Terminal Emulator Notes

There is a DECTERM\_PORT routine that creates a DECterm window. This routine is found in the decw\$terminalshr shareable image. An example of linking with this routine follows:

```
$_LINK FOO, $_SYS$INPUT/OPTIONS  
$_SYS$SHARE:DECW$TERMINALSHR/SHARE  
$_
```

The DECTERM\_PORT routine is as follows:

## DECTERM PORT

Creates a DECterm window that emulates a VT320 terminal.

**VAX FORMAT**    **VMS Status Code =***DECW\$TERM\_PORT*  
(*display, setup\_file, customization, result\_dev, result\_len*)

**argument information**

Argument	Usage	Data Type	Access	Mechanism
display	char_string	char string	read	By descriptor
setup_file	char_string	char string	read	By descriptor
customization	char_string	char string	read	By descriptor
result_dev	char_string	char string	write	By descriptor
result_len	word	word	write	Reference

**MIT C FORMAT**    **VMS Status Code =***DECwTermPort*  
(*display, setup\_file, customization, result\_dev, result\_len*)

```
DECwTermPort (display, setup_file, customization, result_dev,
              result_len)

char* display;
char* setup_file;
char* customization;
char* result_dev;
short* result_len
```

**ARGUMENTS**

***display***

A character string that identifies which server and screen the created DECterm appears on. If set to 0, the default is used. This argument is specified as the address of a descriptor for the VAX binding and the address of a null-terminated string for the C binding. The maximum display name length is 50 characters.

***setup\_file***

A character string that specifies the name of the setup file. The setup file changes DECterm's initial settings. If set to 0, the default setup file is used. This argument is specified as the address of a descriptor for the VAX binding and the address of a null-terminated string for the C binding. The maximum length of this string is 200 characters.



## DECterm Port Routine

### DECTERM PORT

#### ***customization***

A character string that specifies setup options. The syntax is:

```
"param=value, param=value, param=value..."
```

This argument is passed by a descriptor or null-terminated string. The maximum allowed length is 512 characters.

#### ***result\_dev***

A character string that specifies the device name for the created DECterm. It is intended for applications that may wish to assign the created DECterm or pass the name to create a new process. This argument is specified as a descriptor for the VAX binding and as an address of an area for the C binding. The area pointed to by the **result\_dev** descriptor must be 50 characters long.

#### ***result\_len***

This is the address of a 16-bit word into which the actual length of the returned device name is written. For VAX, you can point this argument directly at the **result\_dev** descriptor to trim the descriptor for subsequent use.

---

### **DESCRIPTION**

DECTERM\_PORT creates a DECterm window, displays it on the screen, and returns a device name for the caller to use in other system calls, such as \$ASSIGN, \$QIO, \$SPAWN, or \$CREPRC. If you use \$SPAWN, you must first assign the DECterm.

# Programmer Release Notes

## 4.10 DECwindows Programming Notes

---

### 4.10.5.1 Terminal Emulator Font Notes

All DECwindows Terminal fonts are for private use by DECterm, and should not be used by other applications. There are several problems with the terminal emulator fonts:

- Character set encoding problems with ISO8859-1 (ISOLatin1) fonts. The character set is a hybrid of DEC Multinational and Special Graphics that does not correspond to a standard character set.
- Character set encoding problems with DEC-DECtech fonts. The character set is a 7-bit DECTechnical set that is private to Digital.
- There are missing or incorrect characters in many fonts.
- There are various spacing problems in many fonts.
- 28-point fonts are not actually two times the height of the 14-point font (DECterm requirement).
- Line-drawing characters in many fonts (particularly Double\_Wide) do not join properly.
- The Narrow, Wide, and DoubleWide 14-point fonts at 75 dots per inch (dpi) do not have the same cell size as the Normal and Bold 14-point fonts.

---

### 4.10.6 VAX C Definition File Requirements

During the VAX C installation procedure, you have the option to extract the VAX C definition files (.h files), or leave the .h files in the text library. If you extract the definition files, you are able to use #include control lines of the following form:

```
#include <filename.H>
```

All of the DECwindows sample C programs assume that the .h files were extracted; the samples contain #include <module\_name.h> notation for the included files. The DECwindows programming documentation also makes this assumption.

VAX C should be installed using the option to extract the library modules.

If you have already installed VAX C and you did not extract the .h files, the DECwindows sample C programs do not work. To correct this problem, re-install VAX C and extract the .h files.



# 5

---

## Documentation Information

This chapter contains additions and corrections to the VMS Documentation Set.

---

### 5.1 VMS Version 5.1 Documentation Note

The VMS Version 5.1 documentation includes references to a DECwindows desktop application called the PostScript Previewer. Due to last-minute technical difficulties, DIGITAL has not included this application in the VMS Version 5.1 release. Please disregard any documentation references to the PostScript Previewer.

---

### 5.2 New Documentation for VMS Version 5.1

VMS Version 5.1 includes new documentation that describes the DECwindows user and programming interfaces. DECwindows documentation is available in two separate kits: a User Documentation Kit and a Programming Documentation Kit. You receive the User Documentation Kit as part of the documentation Base Set shipped with the software. You can also order it separately. The Programming Kit is an optional kit that must be ordered separately. See the *Overview of VMS DECwindows* for a brief description of each DECwindows manual.

Other new documentation describes Version 5.1 update and DECwindows installation procedures and new VMS features provided to support DECwindows and the Compound Document Architecture.

Table 5-1 lists the order numbers for general VMS Version 5.1 documentation. See the *Overview of VMS Documentation* for information on ordering other VMS manuals and kits.

**Table 5-1 General VMS Version 5.1 Documentation**

Manual	Order Number
VMS Version 5.1 Installation Guide	AA-MG15A-TE
VMS Version 5.1 Release Notes	AA-MG16A-TE
VMS Version 5.1 New Features Manual	AA-MG29A-TE

The order number for the DECwindows User Documentation Kit is QA-09SAB-GZ. This kit contains the manuals listed in Table 5-2.

## Documentation Information

### 5.2 New Documentation for VMS Version 5.1

**Table 5-2 DECwindows User Documentation Kit Contents**

Manual	Order Number
Overview of VMS DECwindows	AA-MG17A-TE
VMS DECwindows User's Guide	AA-MG18A-TE
VMS DECwindows Desktop Applications Guide	AA-MG19A-TE

The DECwindows Programming Documentation Kit is a multi-volume set divided into separately orderable Volumes 1, 2 and 3. The order number for the entire DECwindows Programming Documentation Kit is QA-001AM-GZ. This kit contains the following three volumes:

- Volume 1—Application Programming, order number QA-001A5-GZ. This volume contains the following manuals:

Manual	Order Number
XUI Style Guide	AA-MG20A-TE
VMS DECwindows Guide to Application Programming	AA-MG21A-TE
VMS DECwindows User Interface Language Reference Manual	AA-MG22A-TE
VMS DECwindows Toolkit Routines Reference Volume Part 1	AA-MG23A-TE
VMS DECwindows Toolkit Routines Reference Volume Part 2	AA-MK88A-TE

- Volume 2—Xlib Programming, order number QA-001A6-GZ. This volume contains the following manuals:

Manual	Order Number
VMS DECwindows Guide to Xlib Programming: MIT C Binding	AA-MG24A-TE
VMS DECwindows Guide to Xlib Programming: VAX Binding	AA-MG25A-TE
VMS DECwindows Xlib Routines Reference Manual Part 1	AA-MG26A-TE
VMS DECwindows Xlib Routines Reference Manual Part 2	AA-MG27A-TE

- Volume 3—Device Driver/CDA, order number QA-001A7-GZ. This volume contains the following manuals:

Manual	Order Number
VMS DECwindows Device Driver Manual	AA-MG28A-TE
VMS Compound Document Architecture Manual	AA-MG30A-TE

---

## 5.3 VMS Version 5.1 New Features Manual—Corrections

The corrections to the *VMS Version 5.1 New Features Manual* are as follows:

- The *VMS Version 5.1 New Features Manual*, which documents the version of VAXTPU released with VMS Version 5.1, neglects to mention that the built-in procedure STR takes a second parameter if the first parameter is of type INTEGER. The correct syntax for the STR built-in procedure when the first parameter is of type INTEGER is as follows:

```
string := STR (integer, [base_integer])
```

The optional second parameter indicates what base you want VAXTPU to use when it converts the specified integer to a string. The valid values for this optional parameter are 8, 10, and 16. The default is 10.

For example, the following statement assigns the string "32" to the variable *the\_string*:

```
the_string := STR (32, 10);
```

However, the following statement assigns the string "00000020" to the variable *the\_string*:

```
the_string := STR (32, 16);
```

- In Section 9.4, DFS is referred to as the Distributed File System both in the section title and the text. The correct product name for DFS is VAX Distributed File Service.

---

## 5.4 New EVE Commands and Features

EVE has several new commands and features. For more information about EVE, see the following books:

- *VAX Text Processing Utility Manual*  
Contains an EVE command dictionary and other information about EVE.
- *Guide to VMS Text Processing*  
Contains an introduction to EVE, covering basic text-editing operations, key definitions, and other information.
- *VMS DECwindows Desktop Applications Guide*  
Explains how to use EVE on DECwindows.

In addition, EVE has comprehensive online help for all commands, keys, and other topics.

---

### 5.4.1 New Commands

This section describes the new commands in this version of EVE, both on character-cell terminals and on DECwindows. For more detailed information about these commands, use the online help in EVE, which provides examples or lists of steps for each command.

## Documentation Information

### 5.4 New EVE Commands and Features

#### **COPY Command**

Same as the STORE TEXT command: copies a select range or found range, without removing it, so you can insert it elsewhere. The copied text is stored in the Insert Here buffer, replacing what you previously cut or copied.

With DECwindows, you can use the SET CLIPBOARD command to have the copied text stored in the DECwindows clipboard to transfer text between EVE and other DECwindows applications running concurrently.

#### **CUT Command**

Same as the REMOVE command: removes a select range or found range, so you can insert it elsewhere. The removed text is stored in the Insert Here buffer, replacing what you previously cut or copied. If you are in the Buffer List buffer, CUT deletes the buffer whose name the cursor is on (similar to DELETE BUFFER).

With DECwindows, you can use the SET CLIPBOARD command to have the removed text stored in the DECwindows clipboard to transfer text between EVE and other DECwindows applications running concurrently.

#### **PASTE Command**

Same as the INSERT HERE command: inserts the text you previously removed or copied. The text is pasted from the Insert Here buffer. The text is inserted whether the mode of the buffer is insert or overstrike.

With DECwindows, you can use the SET CLIPBOARD command to copy, cut, and paste text between EVE and other DECwindows applications running concurrently.

#### **FIND NEXT Command**

Searches for another occurrence of a string already entered with the FIND, REPLACE, or WILDCARD FIND command. The effect is the same as pressing the FIND key twice. For example, the following commands search for the word "digital" and then for the next occurrence of the word:

```
Command FIND digital
      .
      .
      .
Command FIND NEXT
```

In typing the FIND command, if you want to find the word "NEXT," put the word in quotes or let EVE prompt you for it.

#### **FIND SELECTED Command**

Searches for text you have selected, rather than your having to type the search string. This is particularly useful to find a lengthy mixed-case string without having to type it exactly.

## Documentation Information

### 5.4 New EVE Commands and Features

#### Steps:

- 1 Select the text you want to find.
- 2 Enter the FIND SELECTED command.

EVE then cancels the select range and searches for the text in the same way as the FIND command.

If you select more than one line, you might want to use the SET FIND WHITESPACE command so that the search spans up to one line break between words.

In typing the FIND command, if you want to find the word "SELECTED," put the word in quotes or let EVE prompt you for it.

#### NEXT BUFFER Command

Puts your next buffer (if any) in the current EVE window, returning the cursor to your last position in that buffer. (It does *not* create a new buffer.) Thus, you can toggle between or cycle through two or more buffers without having to type the buffer names or enter the SHOW BUFFERS command.

#### NEW Command

Creates a new buffer named MAIN, putting it in the current EVE window. If a buffer named MAIN already exists, EVE asks for the name of the new buffer to create. (Pressing RETURN at the prompt cancels the operation.)

#### OPEN Command

Same as the GET FILE command: puts a file you specify into the current EVE window, creating a new buffer, if necessary, with the same name as the file. If the file you specify is one you have already opened in the editing session and if the buffer for the file still exists, EVE returns you to your last position in that buffer.

#### OPEN SELECTED Command

Opens the file whose name you have selected or found, rather than your having to type the file specification. On DECwindows, the select range can be in EVE or in another DECwindows application running concurrently.

#### Steps:

- 1 Select or find the name of the file you want to edit or create.
- 2 Enter the OPEN SELECTED command.

#### PAGINATE Command

Inserts a "soft" page break for a 54-line page. A soft page break (in WPS terms, a page marker) is a form feed followed by the null character—appearing as a small  $F_N$  on a line by itself. Soft page breaks are erased or relocated by subsequent PAGINATE commands. A "hard" page break is a form feed only, typically done with the INSERT PAGE BREAK command (CTRL/L key).



# Documentation Information

## 5.4 New EVE Commands and Features

### Steps:

- 1 Use the PAGINATE command or with the WPS keypad, press GOLD-PF2.

EVE then moves back to the previous page break (if any) and checks ahead for page breaks within the next 54 lines. If soft breaks are found within the 54 lines, EVE erases them, then moves down 54 lines (if there are that many), inserts a soft break, and puts the cursor on the next line.

- 2 You can then repeat the PAGINATE command to continue paginating the buffer.

If a hard break is found, EVE stops on the line after the page break, in case you want to erase it.

### RESTORE SELECTION Command

Reinserts (undeletes) the text erased by a "pending delete." (See the description of the SET PENDING DELETE command.) The text is reinserted whether the mode of the buffer is insert or overstrike. Effectively, the pending delete feature and RESTORE SELECTION give you an additional way to cut and paste text.

### SAVE FILE Command

Writes out the current buffer using the output file associated with that buffer or, if necessary, prompts you for a file specification. Effectively the same as the WRITE FILE command, except you cannot type a file specification on the command line. For example, the following commands let you edit a file called MEMO.TXT and then save your edits in a new version of that file (or as version 1 if you are creating a new file):

```
Command GET FILE memo.txt
.
.
.
Command SAVE FILE
10 lines written to file DISK$1:[GEOFF]MEMO.TXT;2
```

### SAVE FILE AS Command

Writes out the current buffer to the file you specify. Thus, if you are editing a file called ROUGH.DAT, you can save it as a file called FINAL.TXT. The SAVE FILE AS command is effectively the same as the WRITE FILE command, but it always prompts for a file specification. Using SAVE FILE AS does not change the buffer name, but it might change the output file specification associated with that buffer for subsequent SAVE FILE or WRITE FILE commands or for exiting.

### SELECT ALL Command

Selects all of the current buffer without changing your cursor position.

## Documentation Information

### 5.4 New EVE Commands and Features

#### SET BUFFER Command

Sets the editing status of the current buffer—that is, whether you can modify the buffer and whether EVE saves (writes out) the buffer on exiting. You can specify one of the following keywords per command:

Keyword	Effects
MODIFIABLE	(Default setting.) Buffer can be edited (modified). Also, restores the previous mode of the buffer (insert or overstrike), if the buffer was set to unmodifiable.
READ_ONLY	Buffer is <i>not</i> saved (written out) on exiting, even if it has been modified (opposite of WRITE). Also, sets the buffer to unmodifiable; however, you can then set it to modifiable.
UNMODIFIABLE	Buffer cannot be edited (modified). Also, in the status line, Unmodifiable replaces Insert or Overstrike.
WRITE	(Default setting.) Buffer is saved (written out) on exiting if it has been modified (opposite of READ_ONLY). If a buffer is read-only or unmodifiable, using SET BUFFER WRITE makes it modifiable and restores its previous mode (insert or overstrike).

Typically you set a buffer to READ\_ONLY, UNMODIFIABLE, or both if it contains information that you do not want to change inadvertently, such as reference data or a previous draft you want to keep intact.

Similarly, if you create a “scratchpad” buffer as a temporary storage area or for tests and demonstrations, you might want to use the following commands to set the buffer to read-only (which implies unmodifiable) and modifiable:

```
Command: SET BUFFER READ_ONLY  
Command: SET BUFFER MODIFIABLE
```

Thus, you can edit (modify) the buffer, but EVE will not save (write out) that buffer upon exiting.

#### SET CLIPBOARD Command

Enables the DECwindows clipboard for copy, cut, and paste operations, so you can transfer text between EVE and other DECwindows applications. The default setting is NOCLIPBOARD. Copy, cut, and paste operations use EVE’s Insert Here buffer, the same as on a character-cell terminal. The setting affects the following commands and EDT keypad functions:

- COPY or STORE TEXT
- CUT or REMOVE
- PASTE or INSERT HERE
- EDT Append key (KP9)
- EDT Replace key (GOLD\_KP9)
- EDT Subs key (GOLD-ENTER)

**Note:** You cannot enable the clipboard unless you are running EVE on DECwindows. Also, WPS keypad functions do *not* use the clipboard. If you want the clipboard enabled for all or most editing sessions on DECwindows, put the SET CLIPBOARD command in your EVE initialization file.

## Documentation Information

### 5.4 New EVE Commands and Features

#### SET PARAGRAPH INDENT Command

Sets the number of spaces to be added or subtracted from the first line in paragraphs you create or reformat. It does not by itself affect existing text.

For example, if you want the first line of each paragraph to be indented five spaces, enter the following command:

```
Command: SET PARAGRAPH INDENT 5
```

Thus, when you create a new paragraph (typically by pressing RETURN twice) or when you reformat text (by using FILL commands), EVE adds five spaces to the start of the first line of the paragraph. The other lines of the paragraph start at the current left margin.

Paragraph indent is relative to the left margin of the buffer. The default setting is 0 (that is, no indent). Also, paragraph indent is a buffer-specific setting; you can have a paragraph indent for each buffer you edit. At startup, SET PARAGRAPH INDENT in an initialization file applies to a system buffer named \$DEFAULTS\$, so that each buffer you create has the same setting. (See EVE help on Defaults.)

You can specify a negative value to produce a “hanging” paragraph (first line to the left of the left margin), so long as the sum of the paragraph indent and the left margin is at least 1. Hanging paragraphs are useful if you want to make a bulleted list. You can use the following procedure to create a bulleted list:

- 1 Assuming your left margin is ordinarily column 1, set the left margin to 4, and set the paragraph indent to -3, which allows for a bullet character (usually a lowercase *o*) and two spaces:

```
Command: SET LEFT MARGIN 4  
Command: SET PARAGRAPH INDENT -3
```

- 2 Make each list item a separate paragraph (by using blank lines between each item). The counter or bullet at the start of each list item will be three columns left of the left margin (in this case, column 1). The other lines of a list item will begin at the left margin (in this case, column 4).
- 3 When you finish making the list, restore your previous settings. (Because EVE algebraically adds the paragraph indent to the left margin, the order in which you enter the commands might be important.)

```
Command: SET PARAGRAPH INDENT 0  
Command: SET LEFT MARGIN 1
```

#### SET PENDING DELETE Command

Enables pending delete of a select range. **Pending delete** means that, if you select text, you can erase the select range by using DELETE or by typing new text. This is useful for quickly erasing or replacing a block of text. To put back what you deleted, use the RESTORE SELECTION command.

## Documentation Information

### 5.4 New EVE Commands and Features

The default setting is NOPENDING DELETE. Using DELETE erases the character left of the cursor, whether text is selected or not, and typing new text inserts those characters. If you want pending delete enabled for all or most editing sessions, add the SET PENDING DELETE command to your EVE initialization file.

---

#### 5.4.2 New Features

The following sections describe the new EVE features other than new commands.

##### Enhanced Choices Buffer

The **Choices buffer** is an EVE system buffer that is displayed when you enter an ambiguous command. For example, you enter the following command to open a file with the type TXT:

```
Command: GET FILE *.txt
```

Then, if more than one file matches your request, EVE recalls the command line and displays the Choices buffer in a second window. The Choices buffer lists the matching files. For example, there might be two files, LETTER.TXT and MEMO.TXT. To scroll through the list, if necessary, press the NEXT SCREEN and PREV SCREEN keys. To complete the command, type your choice—for example, L\*.TXT or M\*.TXT—and press RETURN.

If the Choices buffer is displayed, you can put the cursor in the Choices buffer and select the choice you want without having to edit or retype the command line:

- With DECwindows, point with the mouse to the choice you want and click MB1 twice. The first click copies the choice onto the command line; the second click executes the command.
- Press a key defined as NEXT WINDOW or PREVIOUS WINDOW to put the cursor in the Choices buffer. Then move the cursor to the choice you want to use, and press DO.
- Put the cursor in the Choices buffer. Then, press SELECT to copy the choice onto the command line. Move back to the command window. If necessary, use command-line editing. Then press RETURN to execute the command.

**Note:** The Choices buffer is displayed if you enter an ambiguous HELP topic, but you cannot select a HELP topic from the Choices buffer. Instead, you must enter the topic name.

##### Default Key Definitions

Setting the GOLD key—by itself or by setting the EDT or WPS keypad—automatically defines GOLD-PREV SCREEN and GOLD-NEXT SCREEN as PREVIOUS WINDOW and NEXT WINDOW respectively, as well as other GOLD key combinations, unless you have already defined the keys otherwise. For more information on default GOLD key combinations, see EVE help on Gold Keys.

## Documentation Information

### 5.4 New EVE Commands and Features

The PF4 key is defined as DO in both standard EVE keypads (Numeric and VT100). You can redefine either of these keys—for example, setting the EDT or WPS keypad redefines PF4. You must have a key defined as DO.

#### WPS Keypad Ruler Key

Using the WPS Ruler key (GOLD-R) lets you set the paragraph indent (P or I) and wrap point (W), as well as margins and tab stops. Paragraph indent set with the ruler is independent of the left margin; whereas, if it is set by command, it is relative to the left margin. For more information, see EVE help on Ruler Keys.

#### Status Line

The EVE status line includes fields or indicators for the buffer's read-write status and modification status, as well as the mode and direction. Typically, the status line for a buffer (in this example, called MAIN), looks like the following example, appearing in reverse video on your screen:

```
Buffer: MAIN          | Write | Insert | Forward
```

If you enter the SET BUFFER READ\_ONLY command, the status line looks like the following example:

```
Buffer: MAIN          | Read-only | Unmodifiable | Forward
```

If you invoke EVE on DECwindows, even running the character-cell version of EVE, the status line is “hot” or mouse-sensitive—that is, it comprises a menu as well as an informational display. Clicking MB1 on the buffer name in the status line is the same as entering the NEXT BUFFER command; clicking MB1 on the other fields toggles the respective setting, for example, to change the mode of the buffer from insert to overstrike. For more information, see EVE help on Status Line.

---

### 5.5 SYSMAN DISKQUOTA REMOVE Command—Correction

The DISKQUOTA REMOVE function does not exist as documented in the *VMS SYSMAN UTILITY MANUAL*. Instead, enter the command DISKQUOTA DELETE to remove an entry from a quota file. DISKQUOTA DELETE works the same way that DISKQUOTA REMOVE is documented.

DISKQUOTA REMOVE will function in VMS Version 5.2. DISKQUOTA DELETE will continue to function in future releases as well.

---

### 5.6 VAX Text Processing Utility Manual—Omissions

The VMS Version 5.0 *VAX Text Processing Utility Manual* did not document that the major ID of the TPUSHR.EXE shareable image was changed in the version of VAXTPU shipped with VMS Version 5.0. The major ID is a field in the image header of a shareable image.

If you run or activate an image that was linked using the old version of TPUSHR.EXE, the image activator aborts with an ID mismatch message. To create an application or other image that works with the new TPUSHR.EXE, relink the application or image.

### 5.6 VAX Text Processing Utility Manual—Omissions

The change in TPUSHR was made necessary by the fact that the new version of VAXTPU removed the following universal symbols:

- TPU\$FACILITY—The VAXTPU facility number. The VAXTPU facility number is now available using the symbol TPU\$\_FACILITY.
- TPU\$MESSAGE\_FLAGS—An address that contained the address of a byte containing the current message flags.
- TPU\$FACILITY\_NAME—An address that contained the address of a descriptor for the facility name.
- TPU\$COMMAND\_TABLE—An address that contained the address of the Command Line Interpreter (CLI) command table used to parse the EDIT/TPU command.

The access to these data areas was not vectored, so applications that used these addresses in the TPUSHR image had to be relinked with each release of VAXTPU. This change in the callable interface made it necessary to change the major ID so any application that attempted to use these addresses would be found.

---

#### 5.6.1 VAX Text Processing Utility Manual EVE Source File Location

The *VAX Text Processing Utility Manual* for VMS Version 5.0 neglects to mention that the EVE source files in Version 5.0 are stored in the location pointed to by the logical name SYS\$EXAMPLES, not in the location pointed to by the logical name SYS\$LIBRARY where they were stored in VMS Version 4.7.

---

#### 5.7 VMS RTL Library (LIB\$) Manual—Documentation Error

In the description of the LIB\$CREATE\_VM\_ZONE routine, an extraneous argument named **number-of-areas** is documented. This argument is not valid. The total number of arguments is 13, not 14 as shown in the manual.

---

#### 5.8 Correction to VMS RTL Parallel Processing (PPL\$) Manual

All of the **flags** arguments in the RTL PPL\$ routines are passed by reference, not by value. The following routines *incorrectly* show the **flags** argument as passed by value:

```
PPL$CREATE_SHARED_MEMORY
PPL$DELETE_SHARED_MEMORY
PPL$FLUSH_SHARED_MEMORY
PPL$TRIGGER_EVENT
```

## Documentation Information

### 5.9 Documentation for LIB\$SYS\_TRNLOG

#### 5.9

---

### Documentation for LIB\$SYS\_TRNLOG

The RTL routine LIB\$SYS\_TRNLOG was removed from the VMS Version 5.0 *VMS RTL Library (LIB\$) Manual* because the system service that it calls, \$TRNLOG, is obsolete. However, the RTL routine LIB\$SYS\_TRNLOG is not obsolete. Following is the routine description for LIB\$SYS\_TRNLOG that should have appeared in the RTL LIB\$ manual for VMS Version 5.0.

---

## LIB\$SYS\_TRNLOG

### Invoke \$TRNLOG System Service to Translate Logical Name

The Invoke \$TRNLOG System Service to Translate Logical Name routine uses the system service \$TRNLOG to translate a logical name. LIB\$SYS\_TRNLOG returns the logical name's translation using the semantics of the caller's string.

---

<b>FORMAT</b>	<b>LIB\$SYS_TRNLOG</b>	<i>logical-name</i> <i>,[word-integer-dest-length]</i> <i>,destination-string</i> <i>[,byte-integer-table]</i> <i>[,access-mode]</i> <i>[,byte-integer-disable-mask]</i>
---------------	------------------------	---

---

<b>RETURNS</b>	VMS usage: <b>cond_value</b> type: <b>longword (unsigned)</b> access: <b>write only</b> mechanism: <b>by value</b>
----------------	---

---

<b>ARGUMENTS</b>	<b><i>logical-name</i></b> VMS usage: <b>logical_name</b> type: <b>character string</b> access: <b>read only</b> mechanism: <b>by descriptor</b> Logical name. The <b>logical-name</b> argument contains the address of a descriptor pointing to this logical name string.
	<b><i>word-integer-dest-length</i></b> VMS usage: <b>word_unsigned</b> type: <b>word (unsigned)</b> access: <b>write only</b> mechanism: <b>by reference</b> Number of characters written into <b>destination-string</b> , not counting padding in the case of a fixed-length string. The <b>word-integer-dest-length</b> argument contains the address of an unsigned word integer that is this number.  If the input string is truncated to the size specified in the <b>destination-string</b> descriptor, <b>word-integer-dest-length</b> is set to this size. Therefore, <b>word-integer-dest-length</b> can always be used by the calling program to access a valid substring of <b>destination-string</b> .



## Documentation Information

### LIB\$SYS\_TRNLOG

#### ***destination-string***

VMS usage: **char\_string**  
type: **character string**  
access: **write only**  
mechanism: **by descriptor**

Destination string into which LIB\$SYS\_TRNLOG writes the logical name translation. The **destination-string** argument contains the address of a descriptor pointing to this destination string.

#### ***byte-integer-table***

VMS usage: **byte\_signed**  
type: **byte integer (signed)**  
access: **write only**  
mechanism: **by reference**

Logical name table number. The **byte-integer-table** argument contains the address of a signed byte integer that is this table number.

#### ***access-mode***

VMS usage: **access\_mode**  
type: **byte integer (unsigned)**  
access: **write only**  
mechanism: **by reference**

Access mode of entry (process table only). The **access-mode** argument contains the address of a signed byte integer that is this access mode.

The access modes, their numeric values, and symbolic names are as follows:

Mode	Value	Symbolic Name
Kernel	0	PSL\$C_KERNEL
Executive	1	PSL\$C_EXEC
Supervisor	2	PSL\$C_SUPER
User	3	PSL\$C_USER

#### ***byte-integer-disable-mask***

VMS usage: **mask\_byte**  
type: **byte (unsigned)**  
access: **read only**  
mechanism: **by reference**

Table search disable mask. The **byte-integer-disable-mask** argument contains the address of a mask byte that is this mask.

The argument **byte-integer-disable-mask** is passed to this routine by reference and is changed to value for use by \$TRNLOG.

## Documentation Information

### LIB\$SYS\_TRNLOG

The mask-bit settings and their resultant actions are as follows:

Bit	Action
0	Do not search system logical name table
1	Do not search group logical name table
2	Do not search process logical name table

---

<b>DESCRIPTION</b>	See the <i>VMS System Services Reference Manual</i> for a complete description of \$TRNLOG.
--------------------	---

---

<b>CONDITION VALUES RETURNED</b>	SS\$_NORMAL	Routine successfully completed.
	SS\$_NOTRAN	Successfully completed, but the input logical name string was placed in the output buffer because no equivalence name was found.
	LIB\$_STRTRU	Successfully completed, but the source string was truncated on copy.
	SS\$_ACCVIO	The caller cannot read the logical name string or the string descriptor, or cannot write the output length, output buffer, table, or access mode field.
	SS\$_IVLOGNAM	The specified logical name string has a length of zero or is greater than 255 characters. String was placed in the destination string buffer because no equivalence name was found.
	SS\$_RESULTOVF	The destination string buffer has a length of zero, or it is smaller than the resultant string.
	LIB\$_FATERRLIB	Fatal internal error. An internal consistency check has failed. This usually indicates an internal error in the Run-Time Library and should be reported to DIGITAL in a Software Performance Report (SPR).
	LIB\$_INVSTRDES	Invalid string descriptor. A string descriptor has an invalid value in its DSC\$_CLASS field.
	LIB\$_INSVIRMEM	Insufficient virtual memory. A call to LIB\$GET_VM has failed because your program has exceeded the image quota for virtual memory.

## Documentation Information

### 5.10 *VMS DECwindows Toolkit Routines Reference Manual—Addition*

---

#### 5.10 ***VMS DECwindows Toolkit Routines Reference Manual—Addition***

The low-level routines ATTACHED DIALOG BOX CREATE and ATTACHED DIALOG BOX POPUP CREATE have an additional constraint attribute, **resizable**. In the VAX binding the attribute name is DWT\$C\_NRESIZABLE; in the C binding the attribute name is DwtNresizable. This is a Boolean attribute that, if true, allows the attached dialog box change the size of its child widgets. The default is true.

---

#### 5.11 ***VMS DECwindows Xlib Routines Reference Manual—Corrections***

This section describes corrections to the *VMS DECwindows Xlib Routines Reference Manual*. The corrections are as follows:

- The **data** argument in X\$CREATE\_IMAGE should be passed by reference, not by descriptor.
- The **buff\_return** argument in X\$LOOKUP\_STRING should be passed by descriptor, not by reference.
- The **name\_len\_return** argument in X\$GET\_ATOM\_NAME uses a longword instead of of a word.
- The **len\_return** argument in X\$LIS\_FONTS uses a longword instead of a word.
- The documentation for X\$DELETE\_MODIFIERMAP\_ENTRY is incorrect. X\$DELETE\_MODIFIERMAP\_ENTRY returns a status that indicates whether the routine completed successfully. See Section 5.11.1 for the corrected routine description.

---

##### 5.11.1 **Corrected X\$DELETE\_MODIFIERMAP\_ENTRY Routine**

The corrected version of the X\$DELETE\_MODIFIERMAP\_ENTRY routine is as follows:

## DELETE MODIFIERMAP ENTRY

Deletes an entry from a modifier key map structure.

**VAX FORMAT**     *status\_return =*  
**X\$DELETE\_MODIFIERMAP\_ENTRY**  
*(modifier\_keys, keycode\_entry, modifier,*  
*modifier\_keys\_return)*

**argument  
information**

Argument	Usage	Data Type	Access	Mechanism
status_return	cond_value	uns longword	write	value
modifier_keys_return	record	x\$modifier_keymap	write	reference
modifier_keys	record	x\$modifier_keymap	read	reference
keycode_entry	identifier	uns longword	read	reference
modifier	longword	uns longword	read	reference
modifier_keys_return	record	x\$modifier_keymap	write	reference

**MIT C FORMAT**     *modifier\_keys\_return = XDeleteModifiermapEntry*  
*(modifier\_keys, keycode\_entry, modifier)*

**argument  
information**

```
XModifierKeymap XDeleteModifiermapEntry(modifier_keys,
                                         keycode_entry,
                                         modifier)

XModifierKeymap *modifier_keys;
KeyCode keycode_entry;
int modifier;
```

**RETURNS**

***status\_return (VAX only)***  
Whether the routine completed successfully.

***modifier\_keys\_return (MIT C only)***  
A pointer to a modifier keys structure. DELETE MODIFIER MAP ENTRY returns the revised modifier key map structure to this client-supplied structure.

## Window and Session Manager Routines

### DELETE MODIFIERMAP ENTRY

---

#### ARGUMENTS

##### *modifier\_keys*

A pointer to the modifier key map structure from which you want to delete an entry.

##### *keycode\_entry*

The key code that is to be deleted.

##### *modifier*

The modifier for which you want to delete a key symbol. There are eight modifiers in the order (starting from zero) shift, lock, control, mod1, mod2, mod3, mod4, and mod5. You can pass the integer value or one of the following constants:

VAX	MIT C
X\$C_SHIFT_MAP_INDEX	Shift
X\$C_LOCK_MAP_INDEX	Lock
X\$C_CONTROL_MAP_INDEX	Control
X\$C_MOD1_MAP_INDEX	Mod1
X\$C_MOD2_MAP_INDEX	Mod2
X\$C_MOD3_MAP_INDEX	Mod3
X\$C_MOD4_MAP_INDEX	Mod4
X\$C_MOD5_MAP_INDEX	Mod5

##### *modifier\_keys\_return (VAX only)*

A pointer to a modifier keys structure. DELETE MODIFIER MAP ENTRY returns the revised modifier key map structure to this client-supplied structure.

---

#### DESCRIPTION

DELETE MODIFIERMAP ENTRY deletes the specified key code from the set that controls the specified modifier. DELETE MODIFIERMAP ENTRY returns the resulting modifier key map structure.

The modifier map is not shrunk if all of the rows in a column are zero and the number of keys per modifier is 1. See the INSERT MODIFIERMAP ENTRY routine for more information.

---

## 5.12 VMS DECwindows User Interface Language Reference Manual—Corrections

This section describes corrections to the *VMS DECwindows User Interface Language Reference Manual*.

UIL incorrectly allows the specification of three character sets that are not supported in XUI. The unsupported character sets are as follows:

```
DEC_MCS
DEC_HEBREW
DEC_HEBREW_LR
```

Future versions of the UIL compiler will issue an error message when these character sets are encountered.

Support for international applications is provided through use of other supported character sets. In particular, support for Hebrew applications is provided by the ISO\_HEBREW character set. Support for applications using the DEC\_MCS character set is provided by the ISO\_LATIN1 (the default character set in UIL).

The following sections of the *VMS DECwindows User Interface Language Reference Manual* are affected:

- Table 2–5, Examples of String Literal Syntax. Substitute ISO\_HEBREW for DEC\_HEBREW.
- First paragraph in Section 2.4.1.1, Compound String Literals. Substitute ISO\_HEBREW for DEC\_HEBREW.
- Section 2.4.1.1, code example. Substitute ISO\_HEBREW for DEC\_HEBREW.
 

```
#DEC_HEBREW"txet werbeh"&#ISO_LATIN8"latin text"
```
- Table 2–7, UIL-Supported Character Sets. Eliminate DEC\_MCS, DEC\_HEBREW and DEC\_HEBREW\_LR.
- Section 2.4.1.3, code example following Table 2–7. Substitute ISO\_HEBREW for DEC\_HEBREW.
 

```
#DEC_HEBREW"tfel ot thgir morf og sretcarahc"
```
- Table 2–8, Parsing Rules for Character Sets. Eliminate DEC\_MCS, DEC\_HEBREW and DEC\_HEBREW\_LR.
- Section 2.4.1.3, fourth paragraph following Table 2–8. Substitute ISO\_HEBREW for DEC\_HEBREW.

---

## 5.13 VMS Compound Document Architecture Manual—Corrections

Chapter 5 of the *VMS Compound Document Architecture Manual* describes the guidelines for creating user-written front and back ends that work with the Compound Document Architecture (CDA) Converter to translate various file formats to and from a CDA in-memory representation. Both the DDIF\$READ\_format routine (for a front end) and the DDIF\$WRITE\_format routine (for a back end) support an argument called

## Documentation Information

### 5.13 *VMS Compound Document Architecture Manual—Corrections*

**standard-item-list.** This argument identifies the document source and can also contain processing options.

The documentation does not mention that DIGITAL recommends that front and back ends, when they encounter unrecognizable items in the **standard-item-list** parameter, should not return an error. Instead, the unrecognized item should be ignored and processing should continue.

The documentation of the PostScript back end in Chapter 2 of the *VMS Compound Document Architecture Manual* does not mention that, in a PostScript processing option file, the processing option keyword can be preceded by a slash (/). For example:

```
PS/PAPER_ORIENTATION LANDSCAPE
```

---

### 5.14 **Documentation Corrections from the VMS Version 5.0-2 Release Notes**

This section contains additions and corrections to the VMS Documentation Set from the VMS Version 5.0-2 Release Notes.

---

#### 5.14.1 ***VMS Developer's Guide to VMSINSTAL***

Developers of installation procedures using the VMSINSTAL callback SET STARTUP can specify only a file name for P3. The documentation incorrectly states that developers can specify SYSMAN for P3.

---

#### 5.14.2 ***VMS System Dump Analyzer Utility Manual—Requirements for VIRTUALPAGECNT System Parameter***

The Version 5.0 *VMS System Dump Analyzer Utility Manual* recommends that you set the system parameter VIRTUALPAGECNT to the size of the system dump file plus 3000 in order to do the following:

- Maintain sufficient virtual address space for the System Dump Analyzer to map a dump
- Load any required symbol tables
- Store stack information

DIGITAL now recommends that you set VIRTUALPAGECNT to at least the size of the system dump file plus 4700. If your SDA sessions require many symbols (and invoke the READ/EXECUTIVE command), you should set VIRTUALPAGECNT to the size of the dump file plus 5750.

---

### 5.15 **Documentation Corrections from the VMS Version 5.0-1 Release Notes**

This section contains additions and corrections to the VMS Documentation Set from the VMS Version 5.0-1 Release Notes.

### 5.15 Documentation Corrections from the VMS Version 5.0-1 Release Notes

#### 5.15.1 Overview of VMS Documentation

In the *Overview of VMS Documentation*, the order numbers for the following documentation were inadvertently omitted:

- VMS Documentation Release Notes Kit (order number QA-001AY-GZ). When ordering the VMS Documentation Base Set or the VMS Documentation Extended Set, you must also specify the order number for the VMS Documentation Release Notes Kit.
- VMS Documentation Extended Set without Base Set (order number QA-001A4-GZ). This kit includes the following:
  - General User Subkit
  - System Management Subkit
  - Programming Subkit
  - Obsolete Features Kit

In addition, the order number for the VMS Documentation Base Set was incorrect. The correct order number for the VMS Documentation Base Set is QA-09SAA-GZ.

#### 5.15.2 VMS Device Support Manual

The description of the operating system routines EXE\$DEBIT\_BYTCNT\_ALO and EXE\$DEBIT\_BYCNT\_BYTLM\_ALO in the *VMS Device Support Manual* contain incorrect information in the description of their synchronization method.

Each of these routines returns control to its caller at IPL\$\_ASTDEL and *not* at the caller's IPL as stated in the manual. (Note that the similar routines EXE\$DEBIT\_BYTCNT(\_NW) and EXE\$DEBIT\_BYTCNT\_BYTLM(\_NW) *do* return control to the caller at the caller's IPL as documented.)

A future revision of the *VMS Device Support Manual* will contain this correction.

#### 5.15.3 VMS VAXcluster Manual

The following command example in Section 5.5.1 of the *VMS VAXcluster Manual* is incorrect:

```
$ MOUNT/CLUSTER $1$DUS111: /SHADOW=VMS08JUL
```

The correct command is as follows:

```
$ MOUNT/CLUSTER $1$DUS111: /SHADOW=( $1$DUA111, $1$DUA151 ) VMS08JUL
```

A future revision of the *VMS VAXcluster Manual* will contain this correction.



## Documentation Information

### 5.16 VMS Version 5.0 Release Notes—Corrections

---

#### 5.16 VMS Version 5.0 Release Notes—Corrections

This section contains additions and corrections to the *VMS Version 5.0 Release Notes*.

---

##### 5.16.1 RQDX3 Microcode Revision Level Error

Section 8.30 of the *VMS Version 5.0 Release Notes*, which describes a restriction for RQDX3 controllers, identifies an incorrect microcode revision level. The corrected note follows:

###### 8.30 Restriction for RQDX3 Controllers

If you are using RQDX3 controllers on a system that serves disks in a Local Area VAXcluster or Mixed-Interconnect cluster and the RQDX3 controller does not contain a microcode revision level of 3.0 or later, you might see frequent controller resets. If your error log shows frequent controller resets during satellite booting, you should contact your local DIGITAL Field Service representative to obtain the latest microcode.

You can determine the controller type and microcode revision level by entering the command `ANALYZE/ERROR_LOG`.

---

##### 5.16.2 Layered Product Availability

In the *VMS Version 5.0 Release Notes*, Tables 1–1 and 1–2 contained inaccurate information regarding the availability of some DIGITAL layered products.

Tables 5–3 and 5–4 summarize the availability of the affected products for Version 5.0-2 of the VMS operating system. The information in these tables supersedes the availability information about these products that is in the *VMS Version 5.0 Release Notes*.

**Table 5–3 Currently Available Products**

Layered Product	Version Number
DECnet/SNA VMS Application Programming Interface	2.2
DECnet/SNA VMS 3270 Data Stream Programming Interface	1.3
DECnet/SNA VMS 3270 Terminal Emulator	1.4
DECnet/SNA VMS APPC/LU6.2	2.0
DECnet/SNA VMS Printer Emulator	1.1
DECnet/SNA VMS Remote Job Entry	1.3
DECnet/SNA Gateway for Synchronous Transport	1.0

(continued on next page)

## Documentation Information

### 5.16 VMS Version 5.0 Release Notes—Corrections

**Table 5–3 (Cont.) Currently Available Products**

Layered Product	Version Number
DECnet/SNA Gateway for Channel Transport	1.0
DECnet/SNA VMS Gateway Management	2.0
VAX 3271 Protocol Emulator	2.4
VMS/SNA	1.3

**Table 5–4 Future Supported Products**

Layered Product	Version Number
EDE-W Document Exchange	2.0
EDE with IBM DISOSS	2.1
VAX Message Router/P Gateway	1.1
VAX Message Router/S Gateway	1.1
DECnet/SNA VMS Distributed Host Command Facility	1.2
DECnet/SNA VMS DISOSS Document Exchange Facility	1.4
DECnet/SNA VMS Data Transfer Facility Server	2.0
DECnet/SNA VMS Data Transfer Facility Utilities	2.0
DECnet/SNA Gateway	1.5
DECnet/SNA VMS Gateway Management for DECSA	1.5
VAX 2780/3780 Protocol Emulator	1.7



# 6

---

## VMS Version 5.0-2 and Version 5.0-1 Release Notes

This chapter contains release notes for VMS Version 5.0-2 and Version 5.0-1. Note that many of the problems and restrictions described in this chapter have been removed in Version V5.1. You should read the notes described in this chapter in conjunction with the Version 5.1 notes.

---

### 6.1 VMS Version 5.0-2 Release Notes

This section contains the release notes for VMS Version 5.0-2.

---

#### 6.1.1 VMS Version 5.0-2 General User Release Notes

This section contains information about the VMS Version 5.0-2 operating system that is of interest to general users.

---

##### 6.1.1.1 SET ENTRY/REQUEUE Problem Corrected

In Versions 5.0 and 5.0-1 of the VMS operating system, a print job or restartable batch job that was submitted to queue "A" and then requeued to queue "B" would incorrectly execute from queue "A."

Version 5.0-2 of the VMS operating system corrects this problem.

---

##### 6.1.1.2 SHOW QUEUE/SUMMARY Now Displays Correct Job Counts

In Versions 5.0 and 5.0-1 of the VMS operating system, the pending, holding, and retained job counts for a queue reported by the DCL command SHOW QUEUE/SUMMARY were sometimes greater than the actual number of jobs in each of these states. The job counts returned by the \$GETQUI system service, the F\$GETQUI lexical function, and the LIB\$GETQUI Run-Time Library routine were also affected. This problem was documented in Section 7.1.8 of the *VMS Version 5.0 Release Notes*.

Version 5.0-2 of the VMS operating system fixes this problem. Job counts that are accurate on a Version 5.0-2 system should remain accurate. However, inaccurate counts can only be corrected by deleting and then recreating the queue.

---

#### 6.1.2 VMS Version 5.0-2 System Manager Release Notes

This section contains information about Version 5.0-2 of the VMS operating system that is of interest to system managers.

# VMS Version 5.0-2 and Version 5.0-1 Release Notes

## 6.1 VMS Version 5.0-2 Release Notes

---

### 6.1.2.1 **SYSGEN Problem with Non-Workstation Q-bus and UNIBUS Systems Corrected**

In Version 5.0 and Version 5.0-1 of the VMS operating system, there was a problem with SYSGEN (System Generation Utility) that affected non-workstation Q-bus and UNIBUS systems.

If the SYSGEN parameter WINDOW\_SYSTEM was set to zero on a Q-bus or UNIBUS system that was not a workstation, the system did not load the drivers for the devices listed in Table 6-1.

**Table 6-1 Devices Affected by SYSGEN Problem**

---

AAV11-C	DTC04	IEQ11
AAV11-D	DTQNA	KWV11-C
ADQ32	IAV11-A	MIRA (M7763)
ADV11-C	IAV11-B	QPSS (LNV21)
ADV11-D	IBQ01	VS31
AXV11-C	IDV11-A	VSV21
CSAM	IDV11-B	VSV24
DRQ32B	IDV11-C	
DSV11	IDV11-D	

---

The workaround was to do a conversational boot and set the WINDOW\_SYSTEM parameter to 1. Version 5.0-2 of the VMS operating system fixes this problem.

---

### 6.1.2.2 **VAX 6200 Series Computers—Problems Booting Standalone BACKUP from a TK50**

On VAX 6200 series systems with 192Mb or more of memory, standalone BACKUP fails to boot from a TK50 and displays the following message:

```
%SYSBOOT-F-PFN allocation overwrites CI ucode
?06 halt instruction executed
PC = xxxxxxxx
```

This message indicates that the system was unable to continue booting because all its memory was allocated. This problem occurs because standalone BACKUP limits the memory configurations to 4Mb. The workaround is to do a conversational boot on the system (see the installation and operations guide for your VAX computer) and enter the following commands at the SYSBOOT> prompt:

```
SYSBOOT> SET PHYSICALPAGES 200000
SYSBOOT> CONTINUE
```

The default value of the PHYSICALPAGES parameter for standalone BACKUP will be raised in a future release of the VMS operating system.

# VMS Version 5.0-2 and Version 5.0-1 Release Notes

## 6.1 VMS Version 5.0-2 Release Notes

- 
- 6.1.2.3 VAXINFO REMOTE—Installation Problem Corrected**  
VAXINFO REMOTE Version 1.4 did not install correctly with Versions 5.0 and 5.0-1 of the VMS operating system. Version 5.0-2 of the VMS operating system corrects this problem. Your system must be running Version 5.0-2 to install VAXINFO REMOTE Version 1.4.
- 
- 6.1.2.4 VMS Batch/Print Facility**  
This section contains notes about the VMS Batch/Print Facility.
- 
- 6.1.2.4.1 Generic Queue Restriction Now Enforced**  
The VMS Batch/Print Facility is designed to allow one execution queue to be the target of no more than eight generic queues. In Versions 5.0 and 5.0-1 of the VMS operating system, this restriction was not enforced, resulting in unpredictable behavior if more than eight generic queues directed jobs to one execution queue.  
  
Beginning in Version 5.0-2 of the VMS operating system, this restriction is enforced.
- 
- 6.1.2.4.2 Job Controller Now Deallocates Virtual Memory**  
In Versions 5.0 and 5.0-1 of the VMS operating system, the job controller failed to deallocate all of the virtual memory it allocated to process a batch or a print job request. This sometimes caused the job controller process to abort with an “insufficient virtual memory” error after processing several thousand jobs.  
  
Version 5.0-2 of the VMS operating system corrects this problem.

### 6.1.3 VMS Version 5.0-2 Programmer Release Notes

This section contains information about Version 5.0-2 of the VMS operating system that is of interest to programmers.

- 
- 6.1.3.1 ACMS V3.0 Queued Task Facility—Restriction Removed**  
Prior to Version 5.0-2 of the VMS operating system, the ACMS Queued Task Facility was subject to a restriction described in the ACMS Version 3.0 Release Notes. Version 5.0-2 of the VMS operating system removes this restriction and supports the full functionality of the ACMS Queued Task Facility.
- 
- 6.1.3.2 VAX Ada CALENDAR Function—Restrictions Removed**  
In previous versions of the VMS operating system, the range of the VAX Ada function CALENDAR."-(LEFT,RIGHT: TIME) was incorrectly restricted to values in the range of the type DAY\_DURATION, rather than the correct (and larger) range of the type DURATION. Also, the limit on the largest possible value of the type TIME was incorrectly specified in the VAX Ada Run-Time Library to be one second earlier than the proper value.  
  
Version 5.0-2 of the VMS operating system corrects both of these problems.

## VMS Version 5.0-2 and Version 5.0-1 Release Notes

### 6.1 VMS Version 5.0-2 Release Notes

---

#### 6.1.3.3 VMS RTL—Problems with MTH\$DCOSD and MTH\$DSIND Functions Corrected

In Version 5.0-1 of the VMS operating system, the VMS Run-Time Library (RTL) math routine MTH\$DCOSD did not return 0 when the input argument was exactly 90 or 270 degrees. Also, the routine MTH\$DSIND did not return 0 when the input argument was exactly 0 or 180 degrees.

Version 5.0-2 of the VMS operating system corrects both of these problems.

---

## 6.2 VMS Version 5.0-1 Release Notes

This section contains the release notes for VMS Version 5.0-1.

---

### 6.2.1 VMS Version 5.0-1 General User Release Notes

This section contains information about the VMS Version 5.0-1 operating system that is of interest to general users.

---

#### 6.2.1.1 SET ACL Command—Use of the /LIKE Qualifier

In Version 5.0-1 of the VMS operating system, use of the /LIKE qualifier with the SET ACL command no longer requires exclusive access to the source object.

---

#### 6.2.1.2 SET ACCOUNTING Facility

In Versions 4.x of the VMS operating system, the following command turned on all types of accounting except for IMAGE accounting:

```
$ SET ACCOUNTING/ENABLE
```

In Versions 5.0 and 5.0-1 of the VMS operating system, the previous command now turns on IMAGE accounting along with the other types of accounting (as documented in the *VMS DCL Dictionary*). Note that turning on all types of accounting including IMAGE accounting can create very large ACCOUNTNG.DAT files. If you want to maintain the Version 4.x behavior, enter the following command:

```
$ SET ACCOUNTING/ENABLE/DISABLE=IMAGE
```

---

### 6.2.2 VMS Version 5.0-1 System Manager Release Notes

This section describes problems that are corrected in Version 5.0-1 of the VMS operating system.

---

#### 6.2.2.1 Backup Utility

This section describes Backup Utility problems that are fixed.

# VMS Version 5.0-2 and Version 5.0-1 Release Notes

## 6.2 VMS Version 5.0-1 Release Notes

---

### 6.2.2.1.1 Data Overwrite Problem with MOUNT/NOUNLOAD

In Version 5.0 of the VMS operating system, if you mounted a tape using the MOUNT/NOUNLOAD command and then used the tape in a backup operation that required multiple tapes, the Backup Utility could not unload the first tape. Instead of prompting you to load a new tape, the Backup Utility continued processing.

**Note: In a save operation, this problem caused data on the first tape to be overwritten by data that was intended for the second tape.**

Version 5.0-1 of the VMS operating system fixes this problem. The Backup Utility can now successfully unload the tape, regardless of how it was mounted.

---

### 6.2.2.1.2 Errors Using TU81-Plus Tape Drives

When the Backup Utility writes to a faulty magnetic tape, it recovers by positioning the tape after the bad block and rewriting the data on a good block. However, in Version 5.0 of the VMS operating system, a tape written in this manner could not be read by a TU81-Plus tape drive. When a TU81-Plus tape drive attempted to read the tape, the following error message resulted:

```
%BACKUP-E-FATALERR, fatal error on MUA0:[ ]TEST.;  
-SYSTEM-F-TAPEPOSLOST, magnetic tape position lost  
%BACKUP-I-SPECIFY, specify option (QUIT, CONTINUE or RESTART)
```

Choosing the CONTINUE or RESTART option caused the error to recur.

Version 5.0-1 of the VMS operating system fixes this problem. When a backup operation encounters a bad block and recovers, a recoverable media error is reported in the error log. A tape with a bad block can be read by a TU81-Plus tape drive.

---

### 6.2.2.1.3 Errors Using Degaussed Tapes on HSC Drives

In Version 5.0 of the VMS operating system, a backup operation to a degaussed tape on an HSC drive failed and displayed the following error message:

```
BACKUP-F-LABELERR, error in tape label processing on 'device'  
SYSTEM-?-NOMSG Message number 8026.
```

Version 5.0-1 of the VMS operating system fixes this problem.

---

### 6.2.2.2 MSCP Server Problem

In Version 5.0 of the VMS operating system, the buffer segmentation algorithm used by the MSCP (Mass Storage Control Protocol) Server did not correctly account for transfers exhausting the local controller's maximum byte count. Under certain circumstances, this resulted in a port driver deadlock.

Version 5.0-1 of the VMS operating system fixes this problem.



# VMS Version 5.0-2 and Version 5.0-1 Release Notes

## 6.2 VMS Version 5.0-1 Release Notes

---

### 6.2.2.3 Print Symbiont Virtual Memory Deallocation

In previous versions of the VMS operating system, the amount of virtual memory allocated to the print symbiont process increased in relation to the number of print jobs handled by that symbiont. This problem occurred because virtual memory was being allocated in several symbiont routines without subsequent deallocation.

In Version 5.0-1 of the VMS operating system, virtual memory is deallocated in these symbiont routines. This prevents the continual growth of the print symbiont's virtual page count under normal operation.

---

### 6.2.2.4 TU81-Plus Tape Drives—Data Loss

Version 5.0 of the VMS operating system contained a problem with RMS that caused data to be lost without reporting an error message on some systems. The data loss occurred when an RMS \$GET operation was issued to a TU81-Plus tape drive on a VAX 82xx or VAX 83xx computer. The problem did not affect the Backup Utility, but it did affect other utilities such as the COPY and CONVERT utilities.

Version 5.0-1 of the VMS operating system fixes this problem.

---

### 6.2.2.5 VAX 6200 Series Computers

This section describes problems with VAX 6200 series computers that have been fixed.

---

#### 6.2.2.5.1 LP11 Printer Not Supported when Standalone Backup is Booted from a TK50 on VAX 6210 and 6220 Systems

VAX 6210 and 6220 systems that are configured with both a UNIBUS and an LP11 line printer displayed the following message when Version 5.0 standalone BACKUP was booted from a TK50 tape cartridge:

```
SYSGEN%W-NOSUCHFILE, file not found  
[SYSEXE]LPDRIVER.EXE
```

If you received this message, you could not direct the output of a BACKUP/LIST operation to an LP11 printer.

Version 5.0-1 of the VMS operating system fixes this problem.

---

#### 6.2.2.5.2 Power Failure Support Added

Version 5.0-1 of the VMS operating system permits VAX 6200 series computers to recover from a power failure, provided they are equipped with the optional battery backup unit (H7231-N) and a power and logic unit (H7206) of hardware revision E6 or later. Customers who have the battery backup option (H7231-N) but do not have the proper power and logic unit should contact DIGITAL Field Service for a free H7206 upgrade.

---

### 6.2.2.6 VAX Volume Shadowing—Shadow Set Copy Operation Problems

In previous versions of the VMS operating system, shadow set copy operations sometimes caused a CI node to crash with a DSDRIVER BUGCHECK. Analysis of the crash revealed an internal shadowing error of LockNotQueued. This problem was caused by a multinode race condition in the shadowing lock protocols.

Version 5.0-1 of the VMS operating system fixes this problem.

# VMS Version 5.0-2 and Version 5.0-1 Release Notes

## 6.2 VMS Version 5.0-1 Release Notes

### 6.2.2.7

#### DECnet-VAX Session Incompatibility with Phase IV Implementations

Incompatibilities have been found between the Phase IV Session Control Architecture and the Phase IV+ Session Control Architecture. DECnet-VAX V5.0 implements Phase IV+ of the DNA architecture, which means it is affected by the incompatibilities. One of the following two problems can occur when attempting a connection from a Phase IV+ node to a Phase IV node.

- The Phase IV session control architecture defines the `invokeProxy` bit in the connect initiate message as being reserved. It also states that any bit defined as reserved must be set to zero unless otherwise specified. Some Phase IV implementations expect the `invokeProxy` bit to be zero and will reject the connection with a protocol error if it is nonzero. Others do not check this field because it is not used in Phase IV. Because proxy is architected as part of Phase IV+, the `invokeProxy` bit is now nonzero. This causes connections initiated from Phase IV+ implementations to be rejected by Phase IV implementations with a protocol error.
- Phase IV implementations expect the session version number in the connect initiate message to be zero because this is the value for Session V1.0. The session version is now 1 to reflect Session V2.0. The connection will be rejected because of version skew.

Table 6–2 lists the DECnet implementations and the version/update required to resolve the compatibility problems.

**Table 6–2 Patches for DECnet Compatibility Problems**

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

<sup>1</sup>Installation of these patches were optional. Some customers may have elected not to install the patches provided and may be running the correct version without the patches.

# VMS Version 5.0-2 and Version 5.0-1 Release Notes

## 6.2 VMS Version 5.0-1 Release Notes

---

### 6.2.2.8 LMF—New LMF\$CONFIG Command File

If you are an existing service contract customer that installed or upgraded to Version 5.0 of the VMS operating system, the License Management Facility (LMF) used the command file LMF\$CONFIG to generate automatically a VMS Product Authorization Key (PAK) and register it for you. The Version 5.0-1 update provides a new version of the LMF\$CONFIG file.

This new version of LMF\$CONFIG runs automatically on your system if you immediately update to VMS Version 5.0-1 after upgrading or installing VMS Version 5.0. If you are a VAXcluster member, the new version of LMF\$CONFIG also runs on nodes that use the updated system disk as they reboot (as long as they were not rebooted after installing or upgrading to VMS Version 5.0).

If you reboot the nodes that use the system disk before updating to VMS Version 5.0-1, LMF\$CONFIG will run automatically on the updated system disk. However, it will not run automatically on any other nodes. You should run LMF\$CONFIG on all other nodes as soon as possible after updating to Version 5.0-1. Running LMF\$CONFIG does not disrupt a running system; it generates a PAK, which is picked up when the node reboots.

**Note:** This information does not apply if you are a new customer or if you are not a service contract customer.

---

### 6.2.2.9 SYSGEN Problem with Non-Workstation Q-bus and UNIBUS Systems

In Version 5.0 and Version 5.0-1 of the VMS operating system, there is a problem with SYSGEN (System Generation Utility) that affects non-workstation Q-bus and UNIBUS systems.

If the SYSGEN parameter WINDOW\_SYSTEM is set to zero on a Q-bus or UNIBUS system that is not a workstation, the system does not load the drivers for the devices listed in Table 6-3.

**Table 6-3 Devices Affected by SYSGEN Problem**

---

AAV11-C	DTC04	IEQ11
AAV11-D	DTQNA	KWV11-C
ADQ32	IAV11-A	MIRA (M7763)
ADV11-C	IAV11-B	QPSS (LNV21)
ADV11-D	IBQ01	VS31
AXV11-C	IDV11-A	VSV21
CSAM	IDV11-B	VSV24
DRQ32B	IDV11-C	
DSV11	IDV11-D	

---

To make sure the system loads the drivers for these devices, do a conversational boot on the system (see the installation and operations guide for your VAX computer) and enter the following commands at the SYSBOOT> prompt:

# VMS Version 5.0-2 and Version 5.0-1 Release Notes

## 6.2 VMS Version 5.0-1 Release Notes

```
SYSBOOT> SET WINDOW_SYSTEM 1
SYSBOOT> CONTINUE
```

This restriction will be removed in a future maintenance release of the VMS operating system.

---

### 6.2.2.10

#### Starting the Queue Manager in SYSTARTUP\_V5.COM

The following command is used as the example in SY\$MANAGER:SYSTARTUP\_V5.COM for starting the Batch/Print Queue Manager:

```
$ START/QUEUE/MANAGER/BUFFER_COUNT=10/EXTEND_QUANTITY=25
$ SYS$COMMON:[SYSEXE]JBCSYSQUE.DAT
```

The local buffer count value (specified with /BUFFER\_COUNT=10), the initial allocation, and the subsequent file extension value (specified with /EXTEND\_QUANTITY=25) are appropriate for a small system with one batch and one print queue. These values should be increased for standalone time-sharing and clustered machines that support many queues.

The default values for /BUFFER\_COUNT and EXTEND\_QUANTITY are 50 and 100, respectively. These values are generally adequate to support 5 to 20 queues where the total number of concurrent jobs is typically less than 50. To efficiently support more queues and jobs, DIGITAL recommends specifying larger values for these qualifiers when starting the queue manager. Note that the value for the /EXTEND\_QUANTITY qualifier should be the same for all nodes in a cluster.

Increasing the local buffer count decreases the number of direct I/O reads on the queue file required to perform Batch/Print operations at the expense of job controller working set size and locking activity. When memory is available, a large number of local buffers increases system performance. However, if a small amount of memory is available, using 100 or more local buffers can decrease performance by causing excessive page faulting of the job controller process.

The value for extend quantity should be at least 20 percent of the size of the queue file when the queue file is in a steady state. If the value for the extend quantity is too small, fragmentation of the queue file can occur as a result of the many file extend operations being performed on the disk.

---

### 6.2.2.11

#### VAX 6200 Series Computers—Problem Booting Standalone BACKUP Over a CI from a Disk

On VAX 6200 series systems, standalone BACKUP fails to boot over a CI from a disk and displays the following message:

```
UNABLE TO TAKE BUGCHECK AT THIS POINT IN BOOTSTRAP
BUGCHECK CODE IS 0000063C
```

This message indicates that the system performed a bugcheck with resources exhausted because the default value of the SYSGEN parameter NPAGEDYN is inadequate for the operation. The workaround is to do a conversational boot on the system (see the installation and operations guide for your VAX computer) and enter the following commands at the SYSBOOT> prompt:

## VMS Version 5.0-2 and Version 5.0-1 Release Notes

### 6.2 VMS Version 5.0-1 Release Notes

```
SYSBOOT> SET NPAGEDYN 300000  
SYSBOOT> CONTINUE
```

The default value of the NPAGEDYN parameter for standalone BACKUP will be raised in a future release of the VMS operating system.

### 6.2.3 VMS Version 5.0-1 Programmer Release Notes

This section contains information about Version 5.0-1 of the VMS operating system that is of interest to programmers.

#### 6.2.3.1 NCS Utility—Diacriticals Collation Problem

In Version 5.0 of the VMS operating system, the languages in the NCS (National Character Set) library SYS\$COMMON:[SYSLIB]NCS\$LIBRARY.NLB (for example, Multinational, Dutch, English, and so forth) had the diacriticals collating in the wrong order.

Version 5.0-1 of the VMS operating system fixes this problem; however, DIGITAL recommends that you avoid using the following syntax with the MODIFICATIONS keyword:

```
RANGE > STRING
```

For example:

```
CF=_IDENTITY  
MODIFICATIONS=(  
%XC0-%XC4 > "a"
```

The restriction on using this syntax will be removed in a future release of the VMS operating system.

#### 6.2.3.2 RMS \$FREE Restriction

Version 5.0 of the VMS operating system did not permit the \$FREE service (which releases all RMS locks held by the specified stream) to return the RMS\$\_RNL (record not locked) warning status when no records were locked by the specified stream. Instead, the \$FREE service returned the RMS\$\_NORMAL success status.

This restriction has been removed in Version 5.0-1 of the VMS operating system, allowing the \$FREE service to return the RMS\$\_RNL status when no records are locked by the specified stream.

#### 6.2.3.3 VMS RTL

This section describes problems with the VMS Run-Time Library (RTL) that are fixed.

##### 6.2.3.3.1 LIB\$GET\_VM Routine Performance Degradation

In Version 5.0 of the VMS operating system, performance degradation in the LIB\$GET\_VM routine occurred under the following conditions:

- The program created a zone, defaulting all the parameters.
- The program allocated many small pieces of memory that totaled a large portion of memory.

## VMS Version 5.0-2 and Version 5.0-1 Release Notes

### 6.2 VMS Version 5.0-1 Release Notes

- The program made few calls to LIB\$FREE\_VM.

Version 5.0-1 of the VMS operating system fixes this performance degradation problem.

---

#### 6.2.3.3.2 MTH\$DCOSD, MTH\$DSIND, and MTH\$DTAND Functions

The VMS Version 5.0 RTL math routines MTH\$DCOSD, MTH\$DSIND, and MTH\$DTAND erroneously returned 0 when the return value was close to 0 but could still be represented as a D\_float number.

Version 5.0-1 of the VMS operating system fixes this problem by returning the actual D\_floating value when these routines are called.

---

#### 6.2.3.4 NCS Callable Routines Unavailable

In Version 5.0 and Version 5.0-1 of the VMS operating system, the NCS callable routines NCS\$RESTORE\_CF and NCS\$SAVE\_CF are unavailable. These routines are documented in the *VMS National Character Set Utility Manual* and will be included in a future maintenance release of the VMS operating system.

---

#### 6.2.3.5 VMSINSTAL—CHECK\_VMS\_VERSION Callback Enhancement

The CHECK\_VMS\_VERSION callback in VMSINSTAL has been enhanced for Version 5.0-1 of the VMS operating system. This callback still functions as described in the *VMS Developer's Guide to VMSINSTAL*, but now allows you to specify a maintenance release for the minimum\_version and maximum\_version parameters.

Although the format for specifying versions to this callback has changed, products that use this callback are not affected because the old format is still supported. However, DIGITAL recommends that you convert products that use the old format to the new format with the next release of the product.

Also note that, if you need to specify a particular maintenance release when passing the minimum and maximum versions for a product, you must use the new format. The new format for expressing VMS versions is as follows:

vv.u-mh

where:

- *vv* indicates a version
- *u* indicates an update
- *m* indicates a maintenance level
- *h* indicates a limited hardware release (LHR)

For example, 5.0-1 indicates that the version of the product is 5, the update is 0, and the maintenance level is 1.

The CHECK\_VMS\_VERSION callback has the following format:

CHECK\_VMS\_VERSION symbol minimum\_version [option] [maximum\_version]

# VMS Version 5.0-2 and Version 5.0-1 Release Notes

## 6.2 VMS Version 5.0-1 Release Notes

The parameters on the command line indicate the following:

- *symbol* is the name of a global symbol that will be defined with a TRUE/FALSE Boolean value that indicates the results of the version check.
- *minimum\_version* is the minimum VMS version required to install the product. This parameter is passed in the form "vv.u[-mh]" (for example 5.0 or 5.0-1). To pass a maintenance release, you must use this format.

The format "vvu" is also supported (for example 050) and is the minimum value that you can provide for this parameter.

- *option* is used to limit the product installation to a field test version of the VMS operating system. If you specify F on the command line, the product is restricted to the specific field test version of the VMS operating system that is specified by the *minimum\_version* parameter.
- *maximum\_version* is the maximum VMS version required to install the product. Use this parameter if a product will not function above a certain version. This parameter uses the same format as the *minimum\_version* parameter.

For example, to restrict a product installation to VMS Versions 4.6 to 5.0, you could use the following command line:

```
$ VMI$CALLBACK CHECK_VMS_VERSION for$version 046 "" 050
```

However, if you want to restrict the product installation to VMS Versions 4.6 to 5.0-1, you need to use the new format:

```
$ VMI$CALLBACK CHECK_VMS_VERSION for$version 4.6 "" 5.0-1
```

# 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.

This appendix describes the performance improvements associated with running DECwindows applications remotely.

---

### A.1 Recommended Minimum Memory Configurations for DECwindows

Workstation memory configurations of 4 megabytes are supported in a non-clustered DECwindows environment. DIGITAL recommends that, at a minimum, your workstation be configured with at least 6 megabytes of memory for non-clustered use and 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 accept input from both the keyboard and mouse. You need to customize your local 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.



## VMS DECwindows Performance Considerations

### A.2 Running VMS DECwindows Applications Remotely

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

A very busy CPU or network can lead to unpredictable application performance. Conversely, the performance experienced by users logged directly into the remote node depends on the amount of DECwindows work demanded of it.

Applications that have minimal communication with the workstation server generally run very well remotely. 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 application.

#### A.2.1 Suggestion for Running Applications Remotely

The simplest method for running applications remotely is to bring up the FileView application remotely and then initiate other applications through 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, submit the following commands in a batch job:

```
$ SET DISPLAY /CREATE /NODE=display_node  
$ @SYS$MANAGER:DECW$STARTVUE
```

When the FileView application comes up, you can run other remote applications by selecting them from the Applications menu item. You can continue to enter commands in the DECterm window.

# B

## DECwindows Ada Programming Interfaces

---

DECwindows provides programming interface definitions for the Ada language. When you select Ada support at the time of the DECwindows kit installation, four Ada package source files are placed in the SYS\$LIBRARY: directory of your system. These files are:

- CDA\$CDA\_ADA—Package CDA—Compound Document Architecture
- DDIF\$DDIF\_ADA—Package DDIF—Digital Document Interchange Format
- DECW\$DWT\_ADA—Package DWT—DECwindows Toolkit
- DECW\$X\_ADA—Package X—Xlib

These package source files can be individually compiled into your Ada program libraries or compiled into the systemwide Ada predefined library. To make the packages available systemwide, the command file SYS\$UPDATE:DECW\$COMPILE\_ADA\_UNITS.COM is provided.

This command procedure compiles all four packages into the predefined Ada library, and, if the VAX Source Code Analyzer (SCA) product is present, loads SCA analysis data for the packages into the SCA library for the predefined library. The command procedure should be run as a batch job and should have available a minimum of 2000 pages in the working set; however, 3000 pages is preferable. A page file quota of at least 30000 pages is suggested.

Once the units are compiled into the predefined Ada library, you must execute the following Ada program library manager command to make the units visible:

```
$ ACS ENTER UNIT ADA$PREDEFINED CDA,DDIF,DWT,X
```

You need only to do this once. This step is also performed automatically for all Ada program libraries created after the DECwindows units are compiled into the predefined library.

Future installations of DECwindows might replace the Ada packages. If so, the new packages must be compiled as shown. If you have already entered the units into your own library, you must then execute the following command to make your library current:

```
$ACS REENTER *
```

Future installations of VAX Ada might replace the Ada predefined library and remove the DECwindows units. If this occurs, reexecute the DECW\$COMPILE\_ADA\_UNITS.COM command procedure.

If you want to compile the units into your program libraries directly, you must execute the following commands after compilation of packages DWT and X:

## DECwindows Ada Programming Interfaces

```
$ ACS ENTER FOREIGN SYSS$SHARE:DECW$DWTLIBSHR/SHAREABLE DWT
$ ACS ENTER FOREIGN SYSS$SHARE:DECW$XLIBSHR/SHAREABLE X
```

This step is not necessary if the units are entered from the predefined library.

Once the units are entered into your program library, applications that use the DECwindows packages are linked in the normal manner using ACS LINK. It is not necessary to explicitly specify the shareable images when linking.

---

### B.1 Using the Packages

Each package, CDA, DDIF, DWT, and X, contains definitions of constants, structures, status codes, and routines for each facility. All four packages observe certain common conventions for naming and use; these conventions are outlined as follows:

- In each package, the facility prefix (CDA\_, DDIF\_, DWT\_, X\_) has been removed from all the symbols defined in that package. It is intended that the Ada USE clause not be used with these packages. This encourages clarity in the application source and also improves compiler efficiency. For example:

```
with DWT;
procedure CALLBACK (
WIDGET: in DWT.WIDGET_TYPE) is
...
ARGLIST: DWT.ARG_ARRAY_TYPE (0..0);
CSTRING: DWT.COMP_STRING_TYPE;
...
begin
DWT.LATIN1_STRING (.....);
```

In some packages, symbols defined with other facility prefixes are present; these have not been removed from the symbol names. For example, routine XT\$INITIALIZE is DWT.XT\_INITIALIZE.

- When a symbol would conflict with an Ada reserved word or predefined identifier, the last letter of the symbol name is removed. For example, the routine DWT\$STRING is DWT.STRIN. See the individual package descriptions for a list of affected identifiers.
- Unconstrained array types are defined as name\_ARRAY\_TYPE for an array of name\_TYPE elements. The DECwindows documentation sometimes uses name\_LIST for such arrays; in the Ada packages, these names are used when the address of an array is desired, most commonly as an element of a structure.
- All functions are defined as “valued procedures.” The function return value is usually named STATUS or RESULT, depending on the type of value returned.
- The null-terminated strings required by some procedures can be created by concatenating the string with ASCII.NUL. Further information about the interfaces can be found by examining the package sources provided in SYS\$LIBRARY:, as described above.

Usage information for the specific packages is given in following sections.

---

**B.1.1 Package CDA**

This package defines constants and types for the Compound Document Architecture facility. There are no package-specific usage comments for package CDA.

---

**B.1.2 Package DDIF**

This package defines constants and types for the Digital Document Interchange Format facility. There are no package-specific usage comments for package DDIF.

---

**B.1.3 Package DWT**

This package defines constants, types, and procedures for the XUI Toolkit facility. The following usage comments are specific to package DWT:

- The procedure `STRING` is renamed `STRIN` to avoid conflict with the predefined type.
- The parameter `ADDRESS` of procedure `XT_FREE` is renamed `ADDRES` to avoid conflict with the predefined type.
- The following subtype definitions are provided that rename types from package DWT:

**Table B-1 Subtype Definitions—Package DWT**

Subtype	Definition
<code>DISPLAY_TYPE</code>	<code>X.DISPLAY_TYPE</code>
<code>EVENT_TYPE</code>	<code>X.EVENT_TYPE</code>
<code>GC_TYPE</code>	<code>X.GC_ID_TYPE</code>
<code>PIXMAP_TYPE</code>	<code>X.PIXMAP_ID_TYPE</code>
<code>TIME_TYPE</code>	<code>X.TIME_TYPE</code>
<code>SCREEN_TYPE</code>	<code>X.SCREEN_ID_TYPE</code>
<code>WINDOW_TYPE</code>	<code>X.WINDOW_ID_TYPE</code>
<code>XRMDATABASE_TYPE</code>	<code>X.DATABASE_ID_TYPE</code>

- The types `INTEGER_ARRAY` and `ADDRESS_ARRAY` are defined for use with procedures in package DWT, being unconstrained arrays of `INTEGER` and `ADDRESS`, respectively.
- The type `DESCRIPTOR_TYPE` is defined for constructing string descriptors required by certain procedures.

# DECwindows Ada Programming Interfaces

## B.1 Using the Packages

### B.1.4 Package X

This package defines types, structures, and procedures for the Xlib facility. The following usage comments are specific to package X:

- The subtype definitions listed in Table B-2 are provided.

**Table B-2 Subtype Definitions—Package X**

Subtype	Definition
ATOM_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
BITMAP_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
CLASS_LIST_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
COLORMAP_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
CURSOR_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
DATABASE_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
DISPLAY_ID_TYPE	SYSTEM.ADDRESS
DISPLAY_TYPE	SYSTEM.ADDRESS
DRAWABLE_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
FONT_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
GC_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
KEYSYM_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
NAME_LIST_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
PIXMAP_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
PROPERTY_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
REGION_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
SEARCH_LIST_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
SELECTION_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
SCREEN_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
TARGET_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
TIME_TYPE	SYSTEM.UNSIGNED_LONGWORD
TYPE_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD
WINDOW_ID_TYPE	SYSTEM.UNSIGNED_LONGWORD

- The argument `EVENT_TYPE` of procedures `CHECK_TYPED_EVENT` and `CHECK_TYPED_WINDOW_EVENT` has been renamed to `EVENT_TYP` to avoid conflict with the `EVENT_TYPE` type definition.

# DECwindows Ada Programming Interfaces

## B.1 Using the Packages

- `EVENT_TYPE` has been defined as a variant record; subtypes for specific event types are also defined as specific instances of the variant record. The discriminant for `EVENT_TYPE` is the field `EVNT_TYPE`; each variant uses its own prefixes for the field names, for example, `KYEV_DISPLAY` for a key event.

When you declare a variable as being type `EVENT_TYPE`, Ada automatically allocates the maximum possible event size for the variable. When examining event variables, be sure to use only the correct fields for the variant defined by `EVNT_TYPE`; otherwise an Ada constraint error can be generated. For example, the following code is correct:

```
if EVENT.EVNT_TYPE = X.C_EXPOSE then
if EVENT.EXEV_WINDOW = WINDOW_2 then
.
.
.
```

while the following is incorrect:

```
if EVENT.EVNT_TYPE = X.C_EXPOSE and
EVENT.EXEV_WINDOW = WINDOW_2 then
.
.
.
```

The second code fragment would raise a constraint error on the reference to `EVENT.EXEV_WINDOW` if the value of the discriminant (`EVNT_TYPE`) was not `C_EXPOSE`.

---

## B.2 Callbacks

Ada procedures that are to be used as callback routines must be made visible by means of the `EXPORT_PROCEDURE` pragma. This requires that the procedure be a library unit or be declared in the outermost declarative part of a library package. See the section on Exporting Subprograms in the *VAX Ada Language Reference Manual* for more details.

Be aware that `EXPORT_PROCEDURE` implicitly declares the procedure name as a global symbol. If the same procedure name is used in multiple packages, you should specify an “external designator” as the second argument of pragma `EXPORT_PROCEDURE` to give the procedure a unique external name.

Callback routines used in tasking applications must also specify pragma `SUPPRESS_ALL`. This suppresses the task stack check that might fail for routines called from outside the context of an Ada task.

# DECwindows Ada Programming Interfaces

## B.3 Tasking Considerations

---

### B.3 Tasking Considerations

Ada programs that use tasking can call DECwindows routines, but applications designers should be aware that the DECwindows design philosophy is oriented towards event polling and not asynchronous notification of events. A mechanism is available to queue an AST when certain events occur, but this is specific to VMS and should be used cautiously in applications intended to be portable.

An important consideration is that the routines that wait for an event, such as `X.NEXT_EVENT`, block the process until the event occurs. In a tasking program, this means that all tasks are blocked, even if a task of a higher priority is eligible for execution. However, if time slicing is enabled with pragma `TIME_SLICE`, other tasks of equal priority will run at the end of each time slice; but when the stalled task is again scheduled, it will block until its time slice has expired. Tasks of lower priority will not run. However, tasks of higher priority that become runnable by means of an AST completion (such as by using one of the routines in package `TASKING_SERVICES`) will run immediately.

---

### B.4 Examples

There are three Ada language examples provided in the `DECW$EXAMPLES` directory:

- 1 `HELLOWORLD.ADA` is a simple example of using the DECwindows Toolkit and Resource Manager.
- 2 `DECBURGER.ADA` is a more complex example of using various predefined widgets in the DECwindows Toolkit and demonstrates the use of callbacks, as well as more intensive use of the Resource Manager and access to UIL definitions.
- 3 `XLIBINTRO.ADA` demonstrates the use of the Xlib interface and responding to events.

The first two example programs require that the appropriate UIL file from the directory `DECW$EXAMPLES` be compiled using the UIL compiler before running the programs. See the command procedure `DECW$EXAMPLES:DEMO_BUILD.COM` for details on compiling and linking the example applications.

---

# Index

---

## A

---

ACCOUNTNG.DAT file • 6-4  
ACMS Queued Task Facility • 6-3  
Ada  
    programming interfaces • B-1  
    See also VAX Ada • B-1  
ATTACH command • 2-7  
AUTOGEN • 3-2  
Automatic tape labeling feature • 3-1

## B

---

Backup Utility  
    errors • 6-5  
Backup Utility (BACKUP)  
    using with compound document files • 2-2  
Batch print facility  
    See VMS Batch/Print facility  
Batch queue • 6-9  
Bookreader  
    restriction • 2-10  
/BUFFER\_COUNT qualifier • 6-9

## C

---

Cardfiler • 2-6  
CHECK\_VMS\_VERSION callback • 6-11  
    format of • 6-11  
Clock • 2-6  
Color map • 2-5  
Compound documents • 1-1  
CONFIGURE • 3-2  
Console media  
    update after upgrade for VAX 8530, 8550, 8700,  
    8800 • 3-6 to 3-7

## D

---

DAP (DECnet file access protocol)  
    extensions • 4-9

DDIF files  
    mailing • 2-1  
DDIF files  
    using with MAIL utility • 2-1  
DDIF viewer  
    restrictions • 2-6  
DECnet-VAX  
    incompatibility • 6-7  
DECTerm • 2-4  
DECTERM\_PORT routine • 4-14, 4-15  
    creating high level widget • 4-16  
DECwindows • 2-2  
    minimum memory • A-1  
DECwindows applications  
    performance • A-1  
    running remotely • A-1, A-2  
DECwindows Bookreader  
    restrictions • 2-10  
DECwindows Mail  
    features • 2-8  
    INBOX folder problem • 2-9  
    logical names • 2-8  
    online help • 2-8  
    problems and restrictions • 2-9  
DECwindows programming • 4-10 to 4-17  
DECwTermPort routine • 2-4  
DEFINE KEY command • 2-8  
DELETE MODIFIERMAP ENTRY routine • 5-17  
Delta/XDelta Utility (DELTA/XDELTA)  
    commands  
        executing string • 4-8  
        depositing command string in system patch space  
        for use by • 4-8  
Desktop applications • 2-6  
Detached recovery  
    performance • 2-2  
Device reference count • 3-3  
Disk class drivers  
    correction • 3-2  
DISKQUOTA REMOVE function • 5-10  
DSDRIVER disk class driver • 3-2  
DUDRIVER disk class driver • 3-2



## Index

---

# E

---

### Error

during console update • 3–7

### EVE • 5–3

See also EVE commands

Choices buffer • 5–9

command completion • 5–9

commands • 5–3 to 5–9

COPY command • 5–4

CUT command • 5–4

### DECwindows

clipboard support • 2–7

FIND NEXT command • 5–4

FIND SELECTED command • 5–4

GOLD-NEXT SCREEN key • 5–9

GOLD-PREV SCREEN key • 5–9

HELP • 2–7

journaling • 2–7

Keys • 5–9

NEW command • 5–5

NEXT BUFFER command • 5–5, 5–10

NEXT WINDOW command • 5–9

obsolete commands • 2–2

OPEN command • 5–5

OPEN SELECTED command • 5–5

PAGINATE command • 5–5

Paragraph indent • 5–8, 5–10

PASTE command • 5–4

PF4 key • 5–10

PREVIOUS WINDOW command • 5–9

recovery • 2–7

RESTORE SELECTION command • 5–6

restrictions • 2–7

SAVE FILE AS command • 5–6

SAVE FILE command • 5–6

SELECT ALL command • 5–6

SET BUFFER command • 5–7

SET CLIPBOARD command • 5–7

SET GOLD KEY command • 2–2

SET NOCLIPBOARD command • 5–7

SET NOPENDING DELETE command • 5–8

SET PARAGRAPH INDENT command • 5–8

SET PENDING DELETE command • 5–8

source file location • 5–11

status line • 5–10

WPS Ruler key • 5–10

EVE (Extensible VAX Editor) • 2–7 to 2–8

### EVE commands

ATTACH • 2–7

### EVE commands (cont'd.)

DEFINE KEY • 2–8

REPEAT • 2–8

SET CLIPBOARD • 2–7

SET SCROLL MARGINS • 2–7

SPAWN • 2–7

SPELL • 2–7

EXE\$DEBIT\_BYTCNT\_ALO routine • 5–21

EXE\$DEBIT\_BYTCNT\_BYTLM\_ALO routine • 5–21

/EXTEND\_QUANTITY qualifier • 6–9

### Extensible VAX Editor

See EVE

---

# F

---

F\$GETQUI lexical function • 2–1, 6–1

### FileView

profile file • 2–3

FileView process quotas • 2–3

### Folder name parameters

in Mail • 2–1

---

# G

---

General user • 2–1 to 2–10

\$GET operation • 6–6

\$GETQUI system service • 2–1, 6–1

---

# H

---

Highwater marking • 3–3

---

# I

---

### IPL (interrupt priority level)

lowering • 4–7

raising • 4–5

---

# J

---

Job controller • 6–3

---

**L**

---

LATSYM print symbiont • 3-2  
 Layered Product  
   availability • 5-22  
 Layered product caution • 1-1  
 Layered products  
   available • 1-2 to 1-7  
   list of • 1-2  
   supported • 1-7, 1-8  
 LIB\$FREE\_VM routine • 6-11  
 LIB\$GETQUI routine • 2-1, 6-1  
 LIB\$GET\_VM routine • 6-10  
 /LIKE qualifier  
   use with SET ACL command • 6-4  
 LMF\$CONFIG command file • 6-8  
 LMF (License Management facility) • 6-8  
 LOCK\_SYSTEM\_PAGES macro • 4-5

---

**M**

---

MA780 (Multiport Shared Memory) • 3-7  
 Mail  
   See DECwindows Mail  
   See VMS Mail Utility  
 Mass Storage Control Protocol (MSCP) • 6-5  
 Memory  
   minimum recommended for DECwindows • A-1  
 MicroVAX 3400 Series Systems  
   Q-bus devices • 3-4  
 MODIFICATIONS keyword • 6-10  
 Modifier map  
   deleting entry • 5-17  
 MOUNT command • 5-21  
   /AUTOMATIC qualifier • 3-1  
   /NOUNLOAD qualifier • 6-5  
 Mount Utility • 3-1  
 MTH\$DCOSD function • 6-4, 6-11  
 MTH\$DSIND function • 6-4, 6-11  
 MTH\$DTAND function • 6-11  
 Multiport shared memory  
   See MA780

---

**N**

---

National Character Set Utility (NCS) • 6-10

National Character Set Utility (NCS) (cont'd.)  
   MODIFICATIONS keyword • 6-10  
   routines • 6-11  
 NCS\$RESTORE\_CF routine • 6-11  
 NCS\$SAVE\_CF routine • 6-11  
 /NOUNLOAD qualifier  
   use with MOUNT command • 6-5  
 NPAGEDYN parameter • 6-9

---

**O**

---

Online help in DECwindows Mail • 2-8

---

**P**

---

PAT\$\_NONPAGED • 4-8  
 PAT\$\_NONPGD  
   See PAT\$\_NONPAGED  
 Patch space • 4-8  
 Per-process page  
   locking in memory • 4-3  
 PF4 key  
   in EVE • 5-10  
 PHYSICALPAGES parameter • 6-2  
 Poor man's lockdown • 4-3 to 4-8  
 PostScript Previewer  
   documentation note • 5-1  
 Print symbiont  
   virtual memory deallocation • 6-6  
 Pseudo terminal driver • 3-8

---

**Q**

---

Q-bus devices  
   manually configuring on MicroVAX 3400 Series  
   Systems • 3-4  
 Queue  
   batch • 6-9  
   execution • 6-3  
   generic • 6-3  
   print • 6-9  
 Queue file  
   fragmentation • 6-9  
 Queue manager  
   starting • 6-9

## Index

---

### R

---

REPEAT command • 2-8  
RMS \$TRUNCATE service • 4-1  
RMS (Record Management Services)  
  \$FREE restriction • 6-10  
  problem • 6-6  
RMS Statistics  
  restrictions • 4-1  
RQDX3 microcode revision level  
  correction • 5-22  
RTL (Run-Time Library) • 6-4, 6-10  
RTL LIB\$SYS\_TRNLOG routine  
  description • 5-12  
RTL Parallel Processing (PPL\$) Manual  
  correction • 5-11  
RTL PPL\$ exit handler function  
  correction • 4-2  
RUN <process> command  
  correction • 4-1

---

### S

---

Scroll margin  
  restriction • 2-7  
Server • 4-13  
SET (SCROLLING) built-in procedure • 2-7  
SET ACCOUNTING Facility • 6-4  
SET ACL command  
  /LIKE qualifier • 6-4  
SET CLIPBOARD command • 2-7  
SET ENTRY/REQUEUE command • 6-1  
SET SCROLL MARGINS command • 2-7  
SHOW QUEUE command • 2-1, 6-1  
SPAWN command • 2-7  
SPELL command • 2-7  
Standalone BACKUP  
  diskette requirements • 3-8  
  Problem Building on RX01 floppy diskettes • 3-7  
  problems booting • 6-2, 6-6, 6-9  
STARTNET.COM file • 3-3  
Status line  
  in EVE • 5-10  
SYSGEN  
  PQL\_MPRCLM parameter • 2-4

System dump file  
  size • 5-20  
System Generation Utility (SYSGEN)  
  NPAGEDYN parameter • 6-9  
  PHYSICALPAGES parameter • 6-2  
  WINDOW\_SYSTEM parameter • 6-2, 6-8  
System manager • 3-1 to 3-8  
System page  
  locking in memory • 4-3

---

### T

---

Terminal emulator  
  font • 4-17  
  programming • 4-14  
TU81-Plus tape drives • 6-5  
  data loss problems • 6-6

---

### U

---

UETP (User Environment Test Package)  
  modifications • 3-4  
ULTRIX applications  
  running • 2-10  
UNLOCK\_SYSTEM\_PAGES macro • 4-7  
Upgrade considerations • 1-1  
User Environment Test Package  
  See UETP

---

### V

---

VAX 6200 Series computers • 3-5, 6-2, 6-6, 6-9  
  License Unit Requirement Table (LURT) • 3-5  
VAX Ada  
  CALENDAR function • 6-3  
VAX Ada Run-Time Library • 6-3  
VAX Diagnostic Supervisor  
  See VDS  
VAXINFO REMOTE  
  installation • 6-3  
VAX RMS Journaling  
  detached recovery performance • 2-2  
VAX *Text Processing Utility Manual*  
  omissions • 5-10

VAXTPU  
 See VAX Text Processing Utility  
 DECwindows notes • 4–10

VAX Volume Shadowing • 6–6

VDS (VAX Diagnostic Supervisor) • 3–5

Version 5.0-1 Release Notes • 6–1  
 documentation corrections • 5–20  
 general user • 6–4  
 programmer • 6–10  
 system manager • 6–4

Version 5.0-2 Release Notes • 6–1  
 documentation corrections • 5–20  
 general user • 6–1  
 programmer • 6–3  
 system manager • 6–1

Version 5.0 Release Notes  
 documentation corrections • 5–22

VIRTUALPAGECNT parameter • 5–20

VMB.EXE file • 3–5  
 update after upgrade for 8530, 8550, 8700, 8800 •  
 3–6

VMS Batch/Print facility • 6–3

*VMS Compound Document Architecture Manual*  
 corrections • 5–19

*VMS DECwindows Toolkit Routines Reference Manual*  
 addition • 5–16

*VMS DECwindows User Interface Language Reference Manual*  
 corrections • 5–19

*VMS DECwindows Xlib Routines Reference Manual*  
 corrections • 5–16

VMS Documentation Base Kit • 5–21

VMS Documentation Extended Kit • 5–21

VMS Documentation Release Notes Kit • 5–21

VMSINSTAL • 5–20  
 CHECK\_VMS\_VERSION callback • 6–11

VMS Mail Utility • 2–1  
 folder name parameter • 2–1  
 sending DDIF files • 2–1

*VMS RTL Library (LIB\$) Manual*  
 documentation error • 5–11

VMS Version 5.1  
 documentation note • 5–1  
 new documentation • 5–1

*VMS Version 5.1 New Features Manual*  
 correction • 5–3

---

## W

---

Window Manager • 2–2  
 WINDOW\_SYSTEM parameter • 6–2, 6–8  
 WPS Ruler key • 5–10

---

## X

---

X\$DELETE\_MODIFIERMAP\_ENTRY routine • 5–16

Xlib  
 programming • 4–12

XUI Toolkit  
 programming • 4–11

---

## Z

---

Zone analysis  
 problem • 4–2



## 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

<b>Your Location</b>	<b>Call</b>	<b>Contact</b>
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 <sup>1</sup>	_____	SDC Order Processing - WMO/E15 <i>or</i> Software Distribution Center Digital Equipment Corporation Westminster, Massachusetts 01473

---

<sup>1</sup>For internal orders, you must submit an Internal Software Order Form (EN-01740-07).



# Reader's Comments

VMS  
Version 5.1 Release Notes  
AA-MG16A-TE

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

Thank you for your assistance.

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

I would like to see more/less \_\_\_\_\_  
\_\_\_\_\_

What I like best about this manual is \_\_\_\_\_  
\_\_\_\_\_

What I like least about this manual is \_\_\_\_\_  
\_\_\_\_\_

I found the following errors in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

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

Name/Title \_\_\_\_\_ Dept. \_\_\_\_\_  
Company \_\_\_\_\_ Date \_\_\_\_\_  
Mailing Address \_\_\_\_\_  
\_\_\_\_\_ Phone \_\_\_\_\_



- Do Not Tear - Fold Here and Tape -

**digital**™

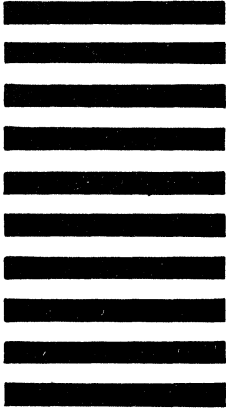


No Postage  
Necessary  
if Mailed  
in the  
United States

**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

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



- Do Not Tear - Fold Here -

Cut Along Dotted Line

# Reader's Comments

VMS  
Version 5.1 Release Notes  
AA-MG16A-TE

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

Thank you for your assistance.

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

I would like to see more/less \_\_\_\_\_

What I like best about this manual is \_\_\_\_\_

What I like least about this manual is \_\_\_\_\_

I found the following errors in this manual:

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

Additional comments or suggestions to improve this manual:  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

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

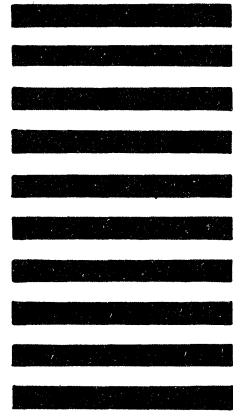
Name/Title \_\_\_\_\_ Dept. \_\_\_\_\_  
Company \_\_\_\_\_ Date \_\_\_\_\_  
Mailing Address \_\_\_\_\_  
Phone \_\_\_\_\_

- Do Not Tear - Fold Here and Tape -

**digital**™



No Postage  
Necessary  
if Mailed  
in the  
United States



**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

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



- Do Not Tear - Fold Here -

Cut Along Dotted Line

digital