DECwindows Motif Version 1.2–3 for OpenVMS Release Notes

January 1995

This manual describes new features, software problems, corrections, and restrictions, as well as documentation changes that pertain to the DECwindows Motif Version 1.2–3 for OpenVMS Alpha and OpenVMS VAX products.

Revision/Update Information:	This manual supersedes the *DECwindows Motif Version 1.2 for OpenVMS Release Notes*.
Operating System:	OpenVMS AXP Version 1.5 VMS Version 5.5–2
Software Version:	DECwindows Motif Version 1.2–3 for OpenVMS Alpha DECwindows Motif Version 1.2–3 for OpenVMS VAX

Digital Equipment Corporation
Maynard, Massachusetts

January 1995

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

Possession, use, or copying of the software described in this publication is authorized only pursuant to a valid written license from Digital or an authorized sublicensor.

The following are trademarks of Digital Equipment Corporation: Alpha, AlphaStation, Bookreader, CDA, DDIF, DEC, DECfonts, DEClinks, DECnet, DECpresent, DECsound, DECterm, DECwindows, DECwrite, DEC Ada, Digital, DTIF, LinkWorks, LN03, OpenVMS, POLYCENTER. ReGIS, TURBOchannel, ULTRIX, VAX, VAX C, VAX DOCUMENT, VAXstation, VMS, VT320, VT330, VT340, XUI, and the DIGITAL logo.

The following are third-party trademarks:

Ada is a registered trademark of the United States government.

Adobe and PostScript are registered trademarks of Adobe Systems Incorporated.

HP, LaserJet, PaintJet, QuietJet, RuggedWriter, and ThinkJet are registered trademarks of Hewlett-Packard Company.

IBM is a registered trademark of International Business Machines Corporation.

Internet is a registered trademark of Internet, Inc.

LaserWriter is a registered trademark of Apple Computer, Inc.

Motif, OSF, OSF/1, and OSF/Motif are registered trademarks and Open Software Foundation is a trademark of the Open Software Foundation, Inc.

National Center for Supercomputer Applications (NCSA) Mosaic and Mosaic are trademarks of the University of Illinois at Urbana–Champaign. This version is based on Enhanced NCSA Mosaic developed by Spyglass, Inc.

POSIX is a registered trademark of the IEEE.

Sun is a registered trademark of Sun Microsystems, Inc.

UNIX is a registered trademark licensed exclusively by X/Open Company Limited.

X Window System is a trademark of the Massachusetts Institute of Technology.

All other trademarks and registered trademarks are the property of their respective holders.

ZK6433

This document is available on CD–ROM.

This document was prepared using VAX DOCUMENT Version 2.1.

Send Us Your Comments

We welcome your comments on this or any other OpenVMS manual. If you have suggestions for improving a particular section or find any errors, please indicate the title, order number, chapter, section, and page number (if available). We also welcome more general comments. Your input is valuable in improving future releases of our documentation.

You can send comments to us in the following ways:

- Internet electronic mail: `openvmsdoc@zko.mts.dec.com`
- Fax: `603-881-0120 Attn: OpenVMS Documentation, ZKO3-4/U08`
- Online form

  Print or edit the online form SYS$HELP:OPENVMSDOC_COMMENTS.TXT. Send the completed online form by electronic mail to our Internet address, or send the completed hardcopy form by fax or through the postal service.

  Please send letters or the form to:

  ```
  Digital Equipment Corporation
  Information Design and Consulting
  OpenVMS Documentation
  110 Spit Brook Road, ZKO3-4/U08
  Nashua, NH  03062-2698
  USA
  ```

Thank you.

How To Order Additional Documentation

Use the following table to order additional documentation or information. If you need help deciding which documentation best meets your needs, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Location	Call	Fax	Write
U.S.A.	DECdirect 800.DIGITAL 800.344.4825	Fax: 800.234.2298	Digital Equipment Corporation P.O. Box CS2008 Nashua, NH 03061
Puerto Rico	809.781.0505	Fax: 809.749.8300	Digital Equipment Carribean, Inc. 3 Digital Plaza, 1st Street, Suite 200 P.O. Box 11038 Metro Office Park San Juan, Puerto Rico 00910−2138
Canada	800.267.6215	Fax: 613.592.1946	Digital Equipment of Canada, Ltd. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6 Attn: DECdirect Sales
International			Local Digital subsidiary or approved distributor
Internal Orders	DTN: 264.3030 603.884.3030	Fax: 603.884.3960	U.S. Software Supply Business Digital Equipment Corporation 10 Cotton Road Nashua, NH 03063−1260

Contents

Preface xi

1 DECwindows Motif Version 1.2–3 for OpenVMS New Features

1.1 OSF/Motif Release 1.2.3 Toolkit 1–1
1.2 POLYCENTER Software Installation Utility Integration 1–1
1.3 DECterm Features 1–2
1.3.1 Scrolling Through the Keyboard 1–2
1.3.2 Reporting the DECterm Window Size 1–2
1.3.3 Resizing the Terminal 1–2
1.3.4 Timeout for Displaying the Copyright Notice 1–2
1.3.5 Cross-Hair and Rubberband-Line Cursor Support 1–2
1.4 DECwindows Mail Enhancements 1–2
1.5 DECnet/OSI Full Names Support 1–3
1.6 Internationalization Enhancements 1–3
1.6.1 Using the CDA Viewer to View Asian-Language Text 1–3
1.6.1.1 Specifying an Options File 1–3
1.6.1.2 Defining Logical Names 1–4
1.6.2 Converting Files That Contain Asian-Language Characters 1–5
1.7 NCSA Mosaic Browser 1–6
1.7.1 Starting Mosaic Software 1–7
1.7.2 Creating a Home Page 1–7
1.7.3 Configuring Mosaic to Process Multimedia Files 1–7
1.7.4 Customizing the Mosaic Browser 1–8
1.7.5 Restrictions for Using Mosaic from Behind a Firewall 1–16
1.7.5.1 ProxyGateway Resource 1–16
1.7.5.2 NoProxy Resource 1–16
1.8 Console Window Application 1–17
1.9 Using Shareable Linkages to Install Images 1–18
1.10 Window Manager Overlay Support 1–19

2 General User Release Notes

2.1 Notes Specific to DECwindows Motif Version 1.2–3 for OpenVMS Software 2–1
2.2 Using Drag and Drop 2–2
2.3 Using Tear-Off Menus 2–2
2.4 Access Control Not Explicitly Enabled by Default 2–2
2.5 Toolkit Warning in the Work in Progress Dialog Box 2–3
2.6 DECwindows Motif Application Notes 2–3
2.6.1 Bookreader 2–3
2.6.1.1 Draft-Quality Printing 2–3
2.6.2 Including Comment Characters in the DECW$BOOKSHELF File . . . 2–3

2.6.3 Calendar . . . 2–3
2.6.3.1 Data-File Format . . . 2–3
2.6.3.2 Additional Problems and Restrictions . . . 2–3
2.6.4 Cardfiler . . . 2–4
2.6.5 CDA Run-Time Services . . . 2–4
2.6.5.1 CDA Dynamic Font Support . . . 2–4
2.6.5.2 CDA WRITE$FONTS Logical Name . . . 2–5
2.6.5.3 CDA Documents . . . 2–5
2.6.5.4 CDA PostScript Documents . . . 2–5
2.6.6 CDA—Additional Applications . . . 2–5
2.6.6.1 Pack and Unpack Applications . . . 2–6
2.6.6.2 Pack Application . . . 2–6
2.6.6.3 Unpack Application . . . 2–7
2.6.6.4 Pack and Unpack Error Messages . . . 2–8
2.6.7 CDA—Hints, Restrictions, and Known Problems . . . 2–8
2.6.7.1 Display PostScript with the CDA Viewer . . . 2–8
2.6.7.2 Message for CDA Viewer . . . 2–8
2.6.7.3 CDA Paper Size Button Renamed . . . 2–9
2.6.7.4 CDA Watch Progress Restriction . . . 2–9
2.6.7.5 CDA Viewer Supports DECfonts Version 1.2 . . . 2–9
2.6.7.6 CDA Viewer—Viewing PostScript Files with Errors . . . 2–9
2.6.8 Clock—DECsound Capability . . . 2–9
2.6.9 DECsound Application . . . 2–9
2.6.9.1 DECsound Prerequisites . . . 2–9
2.6.9.2 DECsound Problems . . . 2–10
2.6.9.3 Escape Sequences . . . 2–10
2.6.10 DECterm—Restrictions and Problems . . . 2–10
2.6.10.1 Changing the Auto Repeat Setting . . . 2–10
2.6.10.2 Positioning DECterm . . . 2–11
2.6.10.3 User Font Selection . . . 2–11
2.6.10.4 Local Echo . . . 2–11
2.6.10.5 Answerback Message . . . 2–12
2.6.10.6 Seven-Bit Printer Support . . . 2–12
2.6.10.7 VT330 and VT340 Terminal Emulation . . . 2–12
2.6.10.8 CREATE/TERMINAL/DETACHED/PROCESS . . . 2–13
2.6.10.9 Printing to an Attached Printer . . . 2–13
2.6.10.10 DECterm Graphics . . . 2–13
2.6.10.11 Initializing DECterm . . . 2–14
2.6.10.12 DECterm Resource Usage . . . 2–15
2.6.10.13 Diagnostic Crash File and Messages . . . 2–15
2.6.10.14 Monitor Screen Width Determines Default Font Size . . . 2–15
2.6.10.15 ReGIS Locator Report . . . 2–15
2.6.11 DECW$CDPLAYER . . . 2–16
2.6.12 LinkWorks . . . 2–16
2.6.12.1 Removing Support for LinkWorks Manager (DEClinks) . . . 2–16
2.6.13 DECwindows Mail . . . 2–16
2.6.13.1 Responses to Keyboard Actions . . . 2–16
2.6.13.2 Using the Color Customizer with DECwindows Mail . . . 2–16
2.6.13.3 DECwindows Mail—Known Problems . . . 2–17

2.6.14 DECwindows Motif Window Manager . . . 2–17
2.6.14.1 Using the Color Customizer with DECwindows Motif Window Manager . . . 2–17
2.6.14.2 Configuration File . . . 2–17
2.6.14.3 DECwindows XUI Applications . . . 2–17
2.6.14.4 Restarting Motif Window Manager . . . 2–18
2.6.15 Motif Window Manager—Known Problems . . . 2–18
2.6.15.1 Accelerators for Icon in Icon Box . . . 2–18
2.6.15.2 Application Started as Icon . . . 2–18
2.6.15.3 Customizing Color-Related Resources for Monochrome Monitors . . . 2–18
2.6.15.4 Customizing Colors on 4-Plane Displays . . . 2–18
2.6.15.5 Customizing Icon Placement in Icon Box . . . 2–18
2.6.15.6 DECwindows XUI Modal Dialog Boxes . . . 2–18
2.6.15.7 Customizing the Icon Box . . . 2–19
2.6.15.8 Moving the Icon Box Off Screen . . . 2–19
2.6.15.9 Truncating the Icon Title . . . 2–19
2.6.15.10 Invoking Motif Window Manager Help . . . 2–19
2.6.15.11 Multihead Systems—Customizing Colors . . . 2–19
2.6.15.12 Multiline Icon Title Not Centered . . . 2–19
2.6.15.13 Truncated Icon Title in Vertical Icon Box . . . 2–19
2.6.16 Notepad . . . 2–19
2.6.16.1 Notepad Is Linked with the OSF/Motif Release 1.1.3 Toolkit . . . 2–19
2.6.17 Paint . . . 2–19
2.6.17.1 Private Colormaps . . . 2–20
2.6.17.2 Enhancing the Performance of Some Paint Operations . . . 2–20
2.6.18 Print Screen . . . 2–20
2.6.18.1 Large Default Icon . . . 2–20
2.6.19 Session Manager and FileView . . . 2–20
2.6.19.1 Removing Support for LinkWorks Manager and LinkWorks . . . 2–20
2.6.19.2 Invoking DECchart from the Session Manager Applications Menu . . . 2–20
2.6.19.3 Eastern European Languages Now Available . . . 2–21
2.6.19.4 Security Options . . . 2–22
2.6.19.5 Using the Color Customizer with Session Manager . . . 2–22
2.6.19.6 Input Focus Change When Starting Private Logo . . . 2–22
2.6.19.7 Session Manager Process—Stopping . . . 2–22
2.6.19.8 Public Profiles for Layered Products . . . 2–23
2.7 Window Dump to Print File (xpr) Utility . . . 2–23
2.8 Keyboard Enhancements for Disabled Users . . . 2–28
2.8.1 Sticky Keys . . . 2–28
2.8.2 Mouse Keys . . . 2–28
2.8.3 Toggle Keys . . . 2–28
2.8.4 Repeat Keys . . . 2–28
2.8.5 Slow Keys . . . 2–28
2.8.6 Bounce Keys . . . 2–29
2.8.7 Time Out . . . 2–29
2.9 Printing from Applications . . . 2–29
2.10 Unsupported Translations by the Motif XmText Widget . . . 2–29

3 System Manager Release Notes

3.1 Notes Specific to DECwindows Motif Version 1.2–3 for OpenVMS Software . . . 3–1
3.2 Required Reboot on OpenVMS Alpha Version 6.2 Systems . . . 3–1
3.3 DECwindows Motif Login Screen—Known Color Problem . . . 3–1
3.4 Color Customizer Example Program . . . 3–2
3.4.1 Supported Displays . . . 3–2
3.4.2 Supported Applications . . . 3–2
3.4.3 Building the Color Customizer on OpenVMS Systems . . . 3–2
3.4.4 Running the Color Customizer . . . 3–3
3.4.5 Modifying the DECW$LOGIN.COM File . . . 3–3
3.4.6 Command Interface Summary . . . 3–3
3.4.7 Changing the Mapping Between Color Resources and Color Cells . . . 3–4
3.4.8 DECterm Windows Not Affected . . . 3–5
3.4.9 Changing the Default Value of the Automatic Shadowing Toggle Button . . . 3–5
3.4.10 Using the Customizer on Multihead Systems . . . 3–6
3.4.11 Using the XSETROOT_CUST.EXE Demonstration Program . . . 3–6
3.5 Define DECW$UTILS Global Symbol When Moving DECW$EXAMPLES Global Symbol . . . 3–6
3.6 FileView Creates Detached Processes by Default . . . 3–7
3.7 Customizing the Login Screen . . . 3–7
3.7.1 Customizing the Digital Logo and Login Screen Colors . . . 3–8
3.7.2 Changing Positions of the Start Session and Set Password Dialog Boxes . . . 3–8
3.7.3 Disabling a Node Name Display in the Start Session Dialog Box . . . 3–9
3.8 Customized Login Logos . . . 3–9
3.9 Version Checking Command Files . . . 3–9
3.10 Implications of the Message, “System Menu Bar: Pseudo Mouse Not Available” . . . 3–10
3.11 DECterm System Management Issues . . . 3–10
3.11.1 DECterm Logical Names . . . 3–10
3.11.2 Automatic Window Positioning . . . 3–11
3.11.3 Hold Screen Response Time . . . 3–11
3.11.4 Using the Debugger . . . 3–11
3.11.5 Virtual Terminal Support . . . 3–12

4 Programmer Release Notes

4.1 Programming Release Notes Specific to DECwindows Motif Version 1.2–3 for OpenVMS Software . . . 4–1
4.2 DECwindows Motif for OpenVMS Toolkit Versions . . . 4–1
4.3 Run-Time and Programming Environments . . . 4–1
4.4 OSF/Motif Toolkit Compatibility . . . 4–2
4.4.1 _Xm Routines . . . 4–3
4.4.2 OSF/Motif Release 1.2.2 and X11 Release 5 Shareable Libraries . . . 4–3
4.5 Language Bindings . . . 4–5
4.5.1 OSF/Motif Release 1.1.3 Ada Bindings . . . 4–5
4.6 OSF/Motif Release 1.1.3 Programming Support and XUI . . . 4–5
4.7 Adding POSIX for OpenVMS Support to DECwindows Transports . . . 4–8
4.7.1 POSIX fork() Routine . . . 4–8
4.7.2 SYS$POSIX_FORK_CONTROL System Service . . . 4–9
4.8 Translated Image Support . . . 4–9
4.8.1 Running Translated Images on OpenVMS Systems . . . 4–9

4.8.2 DECwindows Motif Toolkit Functionality 4–11
4.9 Cross-Development Tools 4–11
4.10 Compiling Applications Written in C 4–12
4.11 Compiling Applications Written in Fortran 4–12
4.12 CDA Application 4–13
4.12.1 CDA—Implementing the Drag-and-Drop Feature 4–13
4.12.2 CDA Run-Time Services Features 4–13
4.12.2.1 Version Number 4–13
4.12.2.2 Changes to the Programming Interface for CDA 4–14
4.12.2.3 Changes to External Reference Processing 4–15
4.12.2.4 Message for Style Guide Fallback 4–15
4.12.2.5 Using Logical Names with CONVERT Commands 4–15
4.12.2.6 Restructuring Shareable Images 4–15
4.13 DECterm Programming 4–16
4.13.1 Page-Movement Escape Sequences 4–16
4.13.2 DECCRA Sequence 4–17
4.13.3 DECLFKC Sequence 4–17
4.13.4 ReGIS Input Cursors 4–17
4.14 DECwindows OSF/Motif Toolkit 4–17
4.14.1 Compilation Error When Including Both CURSES.H and XMP.H Files 4–17
4.14.2 Motif Text Widget Translations 4–18
4.14.3 Upward Compatibility 4–18
4.14.3.1 Restrictions on Mixing Motif and XUI Widgets 4–18
4.14.4 Compile-Time Incompatibilities in Motif Header Files 4–19
4.15 OSF/Motif Example Programs 4–19
4.15.1 Cut and Paste Example Program 4–19
4.15.2 DNDDemo Example Program 4–20
4.15.3 Dogs Example Program 4–21
4.15.3.1 Dog Widget 4–21
4.15.3.2 Square Widget 4–21
4.15.4 Helloint Example Program 4–22
4.15.5 Hellomotif Example Program 4–22
4.15.6 Motifanim Example Program 4–22
4.15.7 Motifgif and Pict Viewing Programs 4–23
4.15.8 Motifshell Example Program 4–23
4.15.9 Periodic Example Program 4–24
4.15.10 Textedit Example Program 4–24
4.15.10.1 Additional Translations 4–25
4.15.11 View Example Program 4–25
4.15.12 Xmpiano Example Program 4–26
4.15.13 Motif Sample Programs 4–27
4.15.14 Xmtravel Example Program 4–28
4.15.15 Resource Files for Example Programs 4–29
4.15.16 UID Files for Example Programs 4–29
4.16 DECwindows Extensions to Motif 4–29
4.16.1 DXmCSText Input Method Support 4–29
4.16.2 SVN—Horizontal Live Scrolling Not Supported 4–30
4.16.3 SVN Horizontal Separator Line 4–30
4.16.4 DXmFormSpaceButtonsEqually Restriction 4–30
4.17 Display Server Extensions 4–30
4.17.1 Display PostScript (DPS) Server Extension—Color Cells 4–30
4.17.2 X Image Extension 4–30
4.17.3 Client Side Extension Library 4–31

4.17.4 Shared Memory Extension Support 4–31
4.17.4.1 How to Use Shared Memory Extension 4–32
4.17.4.2 Using Shared Memory XImages 4–32
4.17.4.3 Using Shared Memory Pixmaps 4–35
4.17.5 Using Extension Include Files 4–36
4.18 Xlib Issues 4–36
4.18.1 xlibint.h Is Now a Public Header File 4–36
4.18.2 Command Procedure Builds .PEN Files 4–36
4.18.3 Parameter/Protocol Datasize Mismatches 4–36
4.18.4 XSelectAsyncEvent and XSelectAsyncInput Routines 4–37
4.18.5 Xlib Internationalization 4–37
4.18.5.1 Vendor Pluggable Layer 4–38
4.18.5.2 Digital Internationalization Xlib Implementation 4–38
4.18.5.3 Locale in OpenVMS Systems 4–39

5 Documentation Release Notes

5.1 Documentation Release Notes Specific to the DECwindows Motif Version 1.2–3 for OpenVMS Product 5–1
5.2 DECwindows Motif Version 1.2–3 for OpenVMS Documentation 5–1
5.3 DECwindows Motif Version 1.2–3 for OpenVMS Documentation Corrections 5–2
5.3.1 *DECwindows Motif for OpenVMS Applications Guide* 5–2
5.3.1.1 Enhancing Information About the Finish Printing Option 5–3
5.3.2 *Using DECwindows Motif for OpenVMS* 5–3
5.3.2.1 Using the Drag-and-Drop Feature 5–3
5.3.2.2 Tear-Off Menus 5–3
5.3.2.3 Adding Target Screen Options to Application Menu Items 5–4
5.3.2.4 Changing the Startup Environment 5–4
5.3.2.5 Enhancing Startup Performance 5–4
5.3.2.6 Enhancing Hold-Screen Response Time 5–5
5.3.3 *Managing DECwindows Motif for OpenVMS Systems* 5–5
5.3.3.1 Global Symbols 5–5
5.3.3.2 Security Options 5–5
5.3.3.3 Displaying a Customized Logo 5–6
5.3.3.4 Enabling and Disabling Access Control 5–6
5.3.4 *DECwindows Motif for OpenVMS Guide to Non-C Bindings* 5–6
5.3.4.1 GET_CHAR_STRUCT Function 5–6
5.3.5 *VMS DECwindows Guide to Xlib (Release 4) Programming: VAX Binding* 5–7
5.3.6 *DECwindows Extensions to Motif* 5–7
5.3.6.1 DXmNlayoutDirection Resource Constants 5–7
5.3.7 *DECwindows Motif Guide to Application Programming* 5–8
5.3.7.1 UIL Source Code for the OpenVMS DECburger Application 5–8
5.3.7.2 Help Widget Implementation Code 5–8
5.3.7.3 Help Widget Implementation—Callbacks 5–8
5.3.7.4 Using UIL to Create the Help Widget 5–8
5.3.8 *Overview of DECwindows Motif for OpenVMS Documentation* 5–8
5.4 OSF/Motif List of Known Problems 5–8

A OSF/Motif Release 1.2 Release Notes

A.1 Performance Improvements A–1
A.2 Backward Compatibility A–2
A.2.1 Visual and Behavioral Compatibility A–2
A.3 Changes and New Features for OSF/Motif Release 1.2 . . . . A–2
A.3.1 General Toolkit Changes A–2
A.3.1.1 Include File Changes A–2
A.3.2 Change in XT Translations A–3
A.3.3 ANSI C Compliance A–3
A.3.4 Display and Screen Specific Data A–3
A.3.5 Drag and Drop A–3
A.3.6 Tear-Off Menus A–4
A.3.7 Insensitive Visuals A–4
A.3.8 Other Visual Changes A–4
A.3.9 Titles for Frames A–5
A.3.10 Audible Warning A–5
A.3.11 Color Enhancements A–5
A.3.12 Baseline Alignment A–5
A.3.13 Expanded Traversal Set A–5
A.3.14 Two-Dimensional Menu Traversal A–5
A.3.15 Input Focus A–5
A.3.16 Traversal Access Functions A–6
A.3.17 Virtual Keys A–6
A.3.18 Resource Management A–7
A.3.19 Changes for CUA and Windows Compliance A–7
A.4 Changes and Enhancements to Specific Widgets A–7
A.4.1 XmClipboard A–7
A.4.2 XmCommand A–8
A.4.3 XmList A–8
A.4.4 XmMessageBox A–8
A.4.5 XmRowColumn and Menus A–9
A.4.6 XmScrollBar A–9
A.4.7 XmScrolledWindow A–9
A.4.8 XmSelectionBox, XmFileSelectionBox A–9
A.4.9 XmText A–9
A.4.10 XmTextField A–10
A.4.11 XmToggleButton, XmToggleButtonGadget A–10
A.5 Motif Window Manager Enhancements A–10
A.5.1 Changes to MWM A–10
A.5.2 New and Enhanced MWM Resources A–11
A.5.3 New and Enhanced MWM Functions A–11
A.5.4 New MWM Action A–12
A.6 Changes to the User Interface Language A–12

Index

Tables

1–1 Asian Language Codes for Options Files 1–4
1–2 Logical Names for Specifying Text Encoding 1–5
1–3 Languages and Associated Basic Fonts 1–6
1–4 Functional Resources 1–9
1–5 Visual Resources 1–13
1–6 Font Resources 1–14
1–7 Proxy Gateway Resources 1–15
2–1 Window Dump to Print File Options 2–24
3–1 Moving the Digital Logo and Changing Login Screen Colors 3–8
3–2 Changing Position of the Start Session and Set Password Dialog Boxes 3–8
3–3 Logical Names Supported by DECterm 3–10
4–1 Names of Shareable Libraries Based on R5 4–4
4–2 Names of Shareable Libraries Based on OSF/Motif Release 1.2.2 4–4
4–3 Directories for Previous XUI or Motif Programming Environment . . . 4–7
4–4 Drag-and-Drop Widgets 4–13
4–5 New Header File Names 4–14
4–6 Names of Shareable Images 4–16
4–7 ReGIS Input Cursors—Cursor styles and Values 4–17
4–8 View Menu Options 4–24
4–9 Textedit Source Files 4–24
4–10 Motif Sample Programs 4–27
4–11 Routine Names and Arguments Sent as 16-Bit Values 4–36
5–1 DECwindows Motif for OpenVMS Documentation 5–2

Preface

The DECwindows Motif Version 1.2–3 for OpenVMS product incorporates the OSF/Motif Release 1.2.3 Toolkit and is based on Release 5 of the X Window System Intrinsics. As with DECwindows Motif Version 1.2 for OpenVMS, DECwindows Motif Version 1.2–3 for OpenVMS continues to provide run-time support for the XUI user interface and Toolkit. Application development is supported for the OSF/Motif Release 1.2.3 Toolkit.

In this manual, DECwindows XUI (X User Interface) refers to the DECwindows product prior to DECwindows Motif Version 1.0.

These release notes describe new features, software problems, corrections, and restrictions, as well as documentation changes that pertain specifically to DECwindows Motif Version 1.2–3 for OpenVMS software. They also describe software problems, corrections, restrictions, and documentation changes from previous DECwindows Motif releases that have not been updated in the existing documentation for those products. Margin notes indicate the versions of the DECwindows Motif products that apply to each release note.

Intended Audience

This manual contains the following information:

- Chapter 1 describes the new features introduced with the DECwindows Motif Version 1.2–3 for OpenVMS product.
- Chapter 2 contains general user release notes for all users.
- Chapter 3 contains release notes intended for system managers.
- Chapter 4 contains release notes intended for application and system programmers.
- Chapter 5 describes additions and corrections to the documentation set.
- Appendix A contains a subset of the release notes for OSF/Motif Release 1.2.3.

Conventions

In this manual, every use of OpenVMS Alpha refers to the OpenVMS Alpha operating system; every use of OpenVMS VAX refers to the OpenVMS VAX operating system; and every use of OpenVMS refers to both the OpenVMS Alpha and VAX operating systems.

In this manual, every use of DECwindows and DECwindows Motif refers to DECwindows Motif for OpenVMS software.

The following conventions are used to identify information specific to OpenVMS Alpha or to OpenVMS VAX systems:

Alpha

The Alpha icon denotes the beginning of information specific to OpenVMS Alpha.

VAX

The VAX icon denotes the beginning of information specific to OpenVMS VAX.

♦

The diamond symbol denotes the end of a section of information specific to OpenVMS Alpha or to OpenVMS VAX.

The following conventions are also used in this manual:

Ctrl/*x*

A sequence such as Ctrl/*x* indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.

PF1 *x*

A sequence such as PF1 *x* indicates that you must first press and release the key labeled PF1 and then press and release another key or a pointing device button.

GOLD *x*

A sequence such as GOLD *x* indicates that you must first press and release the key defined as GOLD and then press and release another key. GOLD key sequences can also have a slash (/), dash (–), or underscore (_) as a delimiter in EVE commands.

The GOLD key definition is often mapped to the PF1 key on the keypad.

Return

In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)

. . .

Horizontal ellipsis points in examples indicate one of the following possibilities:

- Additional optional arguments in a statement have been omitted.
- The preceding item or items can be repeated one or more times.
- Additional parameters, values, or other information can be entered.

.
.
.

Vertical ellipsis points indicate 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 command format descriptions, parentheses indicate that, if you choose more than one option, you must enclose the choices in parentheses.

[ ]

In command format descriptions, brackets indicate optional elements. You can choose one, none, or all of the options. (Brackets are not optional, however, in the syntax of a directory name in an OpenVMS file specification or in the syntax of a substring specification in an assignment statement.)

{ }

In command format descriptions, braces surround a required choice of options; you must choose one of the options listed.

boldface text

Boldface text represents the introduction of a new term or the name of an argument, an attribute, or a reason (user action that triggers a callback).

Boldface text is also used to show user input in Bookreader versions of the manual.

italic text

Italic text emphasizes important information and indicates complete titles of manuals and variables. Variables include information that varies in system messages (Internal error *number*), in command lines (/PRODUCER=*name*), and in command parameters in text (where *device-name* contains up to five alphanumeric characters).

UPPERCASE TEXT

Uppercase text indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.

`struct`

Monospace type in text identifies the following C programming language elements: keywords, the names of independently compiled external functions and files, syntax summaries, and references to variables or identifiers introduced in an example.

-

A hyphen in code examples indicates that additional arguments to the request are provided on the line that follows.

numbers

All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated.

mouse

The term *mouse* refers 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 user can redefine the buttons.)

PB1, PB2, PB3, PB4

PB1, PB2, PB3, and PB4 indicate buttons on the puck.

SB1, SB2, SB3

SB1, SB2, and SB3 indicate buttons on the stylus.

1 DECwindows Motif Version 1.2–3 for OpenVMS New Features

This chapter describes features available with the DECwindows Motif Version 1.2–3 for OpenVMS software:

- OSF/Motif Release 1.2.3 Toolkit
- POLYCENTER Software Installation utility
- DECterm features
- DECwindows Mail enhancements
- DECnet/OSI full names support on OpenVMS systems
- Internationalization enhancements for the CDA Viewer
- NCSA Mosaic browser
- Console Window application
- Installing shareable images by default on OpenVMS Alpha systems
- Window Manager overlay support

1.1 OSF/Motif Release 1.2.3 Toolkit

V1.2–3 The OSF/Motif Release 1.2.3 Toolkit, which is integrated into the DECwindows Motif Version 1.2–3 for OpenVMS software, resolves many problems from previous releases of the Toolkit. New features are not included in Release 1.2.3 of the Toolkit. See Appendix A for Release 1.2.2 and Release 1.2.3 information.

1.2 POLYCENTER Software Installation Utility Integration

V1.2–3 The POLYCENTER Software Installation utility enables users to install software products quickly and with a single command. You can also remove software products that have been installed using the POLYCENTER Software Installation utility. The installation utility provides information about the DECwindows Motif Version 1.2–3 for OpenVMS product. Refer to the *DECwindows Motif Version 1.2–3 for OpenVMS Installation Guide* for POLYCENTER Software installation information.

Note

You must use the POLYCENTER Software Installation utility to install your DECwindows Motif Version 1.2–3 for OpenVMS software on OpenVMS Version 6.2 (or higher) systems. You can use either the POLYCENTER Software Installation utility or the VMSINSTAL procedure to install your DECwindows Motif Version 1.2–3 for OpenVMS software on OpenVMS Version 6.1 systems. You must use the VMSINSTAL procedure to install DECwindows Motif Version 1.2–3 software on Version 5.5–2 through Version 6.0 OpenVMS systems.

1.3 DECterm Features

V1.2–3 This section describes DECterm features that are available in DECwindows Motif Version 1.2–3 for OpenVMS software. Refer to the remaining chapters in this document for specific user, system manager, programmer, and documentation information.

1.3.1 Scrolling Through the Keyboard

V1.2–3 You can scroll through the keyboard by pressing the Ctrl key and arrow keys or by pressing the Prev or Next key on the editing keypad.

1.3.2 Reporting the DECterm Window Size

V1.2–3 When you enable the Show Feedback option in the Workspace Options dialog box of the Window Manager and you resize a DECterm window, the size of the window is reported in characters instead of pixels.

1.3.3 Resizing the Terminal

V1.2–3 If you maximize the DECterm window when you enable the Auto Resize Terminal window option, the window is expanded to its nearest cell boundary; the window does not necessarily occupy the entire screen. The Restore options continue to function appropriately.

1.3.4 Timeout for Displaying the Copyright Notice

V1.2–3 If keyboard or mouse activity does not take place for 10 seconds after the copyright notice is displayed on the terminal, the notice disappears from the DECterm screen.

1.3.5 Cross-Hair and Rubberband-Line Cursor Support

V1.2–3 DECterm supports cross-hair and rubberband-line input cursors, in addition to diamond and rubberband-rectangle ReGIS input cursors. Refer to Section 2.6.9.3 and Section 4.13.4 for additional information.

1.4 DECwindows Mail Enhancements

V1.2–3 The DECwindows Mail application integrates the OSF/Motif Release 1.2.3 Toolkit into the DECwindows Motif Version 1.2–3 for OpenVMS release. Specifically, DECwindows Mail includes the following enhancements:

- Support for tear-off menus
- Enhanced performance

1.5 DECnet/OSI Full Names Support

V1.2–3 DECwindows Motif Version 1.2 for OpenVMS supports DECnet/OSI full names on OpenVMS Alpha Version 6.2 and OpenVMS VAX Version 6.1 (and higher) systems. DECnet/OSI supports the use of names that have a hierarchical structure and are longer than those currently supported by DECnet Phase IV. The following components support DECnet/OSI full names:

- Bookreader
- DECnet transport
- DECterm
- DECwindows Mail
- FileView
- Session Manager
- [DCL command] SET DISPLAY

Refer to the *OpenVMS VAX Version 6.1 New Features Manual* and the *OpenVMS Version 6.2 New Features Manual* manuals for additional information about full names support.

1.6 Internationalization Enhancements

V1.2–3 Additional internationalization support enables users to view and convert files that contain Asian-language characters.

This section provides the following information about internationalization support for the CDA Viewer application:

- Section 1.6.1, Using the CDA Viewer to View Asian-Language Text
- Section 1.6.2, Converting Files That Contain Asian-Language Characters

1.6.1 Using the CDA Viewer to View Asian-Language Text

V1.2–3 You can use the CDA Viewer in two ways to view text files that contain Asian characters:

- Specify an options file to the CDA Viewer application.
- Define logical names at the DCL command level or in a LOGIN.COM file.

Refer to the *DECwindows Motif for OpenVMS Applications Guide* for information about using the CDA Viewer.

1.6.1.1 Specifying an Options File

V1.2–3 Specify an options file by including a one-line entry in the file in the following format:

```
TEXT TEXT_ENCODING text_encoding_value
```

- TEXT is the format.
- TEXT_ENCODING is the option you specify to CDA.
- *text_encoding_value* is the value of the codeset. (See Table 1–1 for a list of values.)

Table 1–1 shows the languages, codesets, and text-encoding values.

Table 1–1 Asian Language Codes for Options Files

Language	Codeset	Text Encoding Value
Japanese	DEC Kanji	DEC_KANJI
Japanese	Super DEC Kanji	SDECKANJI
Traditional Chinese	DEC Hanyu	DEC_HANYU
Simplified Chinese	DEC Hanzi	DEC_HANZI
Korean	DEC Korean	DEC_HANGUL

The following table shows examples of one-line entries.

Options File	One-Line Entry
HANYU.CDA$OPTIONS	TEXT TEXT_ENCODING DEC_HANYU
HANZI.CDA$OPTIONS	TEXT TEXT_ENCODING DEC_HANZI
HANGUL.CDA$OPTIONS	TEXT TEXT_ENCODING DEC_HANGUL

To view the EXAMPLES_CUSTOMERS.TXT file that contains Japanese text in DEC Kanji, use your editor to create an options file called KANJI.CDA$OPTIONS. Add the following one-line entry to the file:

```
TEXT TEXT_ENCODING DEC_KANJI
```

When you access the file through the Options File dialog box with the CDA Viewer, the EXAMPLES_CUSTOMERS.TXT file is viewable in the DEC Kanji codeset (Japanese language).

1.6.1.2 Defining Logical Names

V1.2–3 The second option to enable viewing files in Asian languages is to specify the text file and encoding value by defining two logical names:

- DDIF$READ_TEXT_GL
- DDIF$READ_TEXT_GR

Table 1–2 shows the logical names and associated encoding values.

Table 1–2 Logical Names for Specifying Text Encoding

DDIF$READ_TEXT_GL	DDIF$READ_TEXT_GR	Encoding Value
LATIN1	MCS	MCS
LATIN1	LATIN1	ISO Latin–1
LATIN1	KATAKANA	ASCII–Kana
LATIN1	KANJI	DEC Kanji
ROMAN	MCS	Roman–MCS
ROMAN	LATIN1	Roman
ROMAN	KANJI	Roman–Kanji
ROMAN	KATAKANA	Roman–Kana
LATIN1	HANZI	DEC Hanzi
LATIN1	HANGUL	DEC Hangul
LATIN1	HANYU	DEC Hanyu

You can define the logical names on the DCL command line or in your LOGIN.COM file. For example:

```
$ DEFINE DDIF$READ_TEXT_GL LATIN1
$ DEFINE DDIF$READ_TEXT_GR KANJI
```

Note that this example defines the text encoding for DEC Kanji (see Table 1–2).

1.6.2 Converting Files That Contain Asian-Language Characters

V1.2–3

You can convert an Asian-language text file to another format by specifying an options file or by defining the logical names DDIF$READ_TEXT_GL and DDIF$READ_TEXT_GR as discussed in Section 1.6.1.1 and Section 1.6.1.2.

The format for converting a document from TEXT to another format is as follows:

```
$ CONVERT/DOCUMENT/OPTION=language.CDA$OPTIONS filename.TXT/FORMAT=TEXT -
_$ filename.output_extension/FORMAT=output_format
```

For example, to convert a traditional Chinese language text file to a DDIF file, enter the following command line:

```
$ CONVERT/DOCUMENT/OPTION=HANYU.CDA$OPTIONS -
_$ GUIDELINES_PERSONNEL.TXT/FORMAT=TEXT  GUIDELINES_PERSONNEL.DDIF
```

Note that this command line does not include the /FORMAT=DDIF qualifier; DDIF is the default.

The output file, GUIDELINES_PERSONNEL.DDIF, contains language data.

You can also create Asian language PostScript files from a DDIF, DTIF, or text (ASCII) file. For example, to convert a DDIF file to PostScript (.PS) format, enter the following command:

```
$ CONVERT/DOCUMENT filename.DDIF filename.PS/FORMAT=PS
```

Note

Convert only DDIF and DTIF files that contain language data to Asian language PostScript format.

When you print an Asian language PostScript file on a PostScript printer, ensure that the required language fonts are available on the printer. Otherwise, the PostScript file defaults to a basic set of fonts. If these fonts do not exist, the PostScript file defaults to Courier fonts. Table 1–3 shows the languages and their associated basic fonts.

Table 1–3 Languages and Associated Basic Fonts

Language	Basic Fonts
Japanese	Ryumin-Light-EUC-H or Ryumin-Light-Hankaku
Hanyu	Sung-Light-CNS11643, Sung-Light-DTSCS
Hangul	Munjo
Hanzi	XiSong-GB2312-80

Note

Vertical writing is not supported by the CDA converters. All vertical text is printed horizontally.

1.7 NCSA Mosaic Browser

V1.2–3

DECwindows Motif Version 1.2–3 for OpenVMS provides the National Center for Supercomputing Applications (NCSA) Mosaic Version 2.4 browser to let you navigate and access information on the Internet.

The NCSA Mosaic browser is provided as sample, "as is" software to introduce you to the Internet and help you get started using a World Wide Web viewer. Because the NCSA Mosaic browser that ships with this release is only a sample implementation, we have placed Mosaic with the other sample software in the DECW$UTILS directory. We plan to replace this sample NCSA Mosaic browser implementation in a future release.

The Mosaic browser enables OpenVMS users to search, retrieve, display, store, and forward data using the Internet. Mosaic functions in conjunction with the World Wide Web environment, which is a collection of information servers.

Note that the graphical user interface for Mosaic is not compliant with the Motif style guide; therefore, the Mosaic browser does not look the same as the DECwindows Motif out-of-the-box (OOTB) applications.

The following information about Mosaic is included in this section:

- Section 1.7.1, Starting Mosaic Software
- Section 1.7.2, Creating a Home Page
- Section 1.7.3, Configuring Mosaic to Process Multimedia Files
- Section 1.7.4, Customizing the Mosaic Browser
- Section 1.7.5, Restrictions for Using Mosaic from Behind a Firewall

1.7.1 Starting Mosaic Software

V1.2–3

The Mosaic browser, which is located in the DECW$UTILS directory, is installed as part of the product installation for DECwindows Motif Version 1.2–3 for OpenVMS systems.

To access the verb and start Mosaic, select Menus... from the Options Menu in FileView. Mosaic is listed in the Item Names list. Add the verb Mosaic to a menu, such as the Applications Menu, so that you can invoke Mosaic and display the DECwindows Motif for OpenVMS home page. Refer to *Using DECwindows Motif for OpenVMS* for information about adding verbs to menus.

Note

You can display the sample DECwindows Motif for OpenVMS home page without a TCP/IP connection to the network. However, to access information contained in the topics in the Help menu requires that TCP/IP Services for OpenVMS be installed.

1.7.2 Creating a Home Page

V1.2–3

A home page lets you make information about yourself, your company, your products, and so on available on the Internet. Click on the *On HTML...* topic in the Help menu for information about using HTML (Hypertext Markup Language) to create a home page. HTML uses markup tags to tell Mosaic how to format text. *A Beginner's Guide to HTML* is displayed, which provides information about markup tags, examples, formatting, creating links, troubleshooting, and so on. Note that you can access topics in the Help menu only if your system is connected to the TCP/IP network using TCP/IP Services for OpenVMS.

As an example of an HTML source, see the DECwindows Motif sample home page. Click on *View Source...* from the File menu to see the HTML source.

1.7.3 Configuring Mosaic to Process Multimedia Files

The Mosaic browser that is included with DECwindows Motif Version 1.2–3 for OpenVMS provides support for recognizing and processing the following types of files:

- GIF
- Audio
- MPEG
- MIME
- PostScript

If Mosaic encounters a hyperlink (or an anchor) to any of these file types, it attempts to start an external program to display the image or to play the sound. If Mosaic cannot locate an appropriate external viewer, it prompts the user to enter a file name where the data file can be stored in case the data is needed outside of Mosaic.

To determine what external viewer will display the image or play the sound, Mosaic does the following:

1. Determines the MIME type of the incoming file according to either the file extension or the way the file is specified by the document server.

If the file extension is used to determine the external viewer, Mosaic uses either a built-in default list or an extension map file that can be configured by the user.

2. Matches the MIME file type of the incoming file to an external viewer according to either the built-in default list or a setup that is configured by the user and specified in the file DECW$UTILS:XMOSAIC-SETUP.COM.

For example, GIF files are mapped to the xv viewer. The xv viewer is shareware software that is not provided with DECwindows Motif. However, the xv viewer is available at the following Uniform Resource Locator (URL) address:

```
http://gatekeeper.dec.com/pub/net/infosys/pw/pw-fetch.html
```

The xv mapping is defined in the file DECW$UTILS:XMOSAIC-SETUP.COM, a self-documented command procedure, as follows:

```
    $ XV   :== $DECW$EXAMPLES:PICT.EXE
```

The executable PICT.EXE is used to display the GIF file. Note that PICT.EXE cannot be used to display a JPEG file.

Other mappings include the following:

Image Type	Display/Sound Logical Name
audio/*	→ showaudio
image/xwd	→ xwud
image/x-xwd	→ xwud
image/x-xwindowdump	→ xwud
image/*	→ xv
application/postscript	→ ghostview

Mappings in the previous list that are followed by an asterisk (*) are available in some versions of Mosaic (for example, Version 2.4) that contain native HDF/netCDF support.

Additional examples of external view setup in the file DECW$UTILS:XMOSAIC-SETUP.COM can include the following definitions:

```
  $ APLAY         :== mcr decsound -volume 70 -speaker -play
  $ SHOWAUDIO     :== mcr decsound -volume 70 -speaker -play
  $ GHOSTVIEW     :== view/format=ps/interface=decwindows
  $ XWUD          :== $decw$utils:xwud.exe
```

1.7.4 Customizing the Mosaic Browser

You can customize the Mosaic browser by setting any of the NCSA Mosaic X resources in the file DECW$USER_DEFAULTS:MOSAIC.DAT. The current list of resources is available at the following Internet location:

```
http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/resource
```

The class name for Mosaic for X is Mosaic. The information that follows shows the names and descriptions of the X resources. Note that class and resource names are the same except that the first character in the resource name is capitalized.

The following resources are listed:

- Functional resources

- Visual resources
- Font resources
- Proxy gateway resources

Functional Resources

Table 1–4 lists the functional resources and descriptions.

Table 1–4 Functional Resources

Resource	Description
annotationsOnTop: Boolean	Indicates whether inlined document annotation hyperlinks are prepended to the document (if true) or postpended (if false). (Default is false).
autoPlaceWindows: Boolean	If set to false, the new Document View windows are not automatically positioned by the program itself; your Window Manager determines the placement. (Default is true.)
catchPriorAndNext: Boolean	Controls whether using the keys Prior and Next (Page Up and Page Down on most keyboards) are explicitly caught in the Document View window and passed to the vertical scrollbar, or whether the normal Motif scrolled window mechanism is relied upon to automatically take this action. (Default is false.) If the Page Up and Page Down key sequences are not functioning appropriately for your environment, set this resource to true. (Refer to the information about hot keys.)
confirmDeleteAnnotation: Boolean	Confirms deleted annotations through a confirmation dialog box. (Default is true.)
confirmExit: Boolean	Determines whether the NCSA Mosaic browser pops up a dialog box to confirm exiting from the program when the Exit Program menu option is selected. (Default is true.)
defaultAuthorName: string	Identifies your full name (for example, John Q. Public). Your full name is normally stored in the system password file, and Mosaic extracts the name from that file if this resource is NULL; otherwise, the value of this resource is used. (Default is NULL.)
defaultHeight: integer	Specifies the default height in pixels for a Document View window. (Default is 680.)
defaultHotlistFile: string	Specifies a file name to store the default hotlist (or a list of bookmarks). (Default is ".mosaic-hotlist-default". The value of the environment variable $HOME is prepended to this string.)
defaultWidth: integer	Specifies the default width in pixels for a Document View window. (Default is 620.)

(continued on next page)

Table 1–4 (Cont.) Functional Resources

Resource	Description
delayImageLoads: Boolean	Selects delayed image loading, which is for users with slow network connections. If set to true, Mosaic substitutes a small icon for inlined-images and an arrow icon for inlined-images that also act as hyperlinks. Clicking on an icon tells Mosaic to load that inlined-image; clicking on the arrow is equivalent to following the associated hyperlink. Selecting Load Images in Current from the Mosaic Options menu loads and displays all the inlined-images in the current document. Delayed image loading can also be selected with the -dil command-line flag or on a per-window basis from the Options menu in Mosaic. (Default is false.)
displayURLsNotTitles: Boolean	Displays URLs wherever document titles are usually displayed, if set to true. (Default is false.)
DocsDirectory: string	Overrides the location of the help documents. (Default is NULL.)
documentsMenuSpecfile: string	Specifies the name of the file that holds an optional “Documents Menu Specfile” to allow sites to add their own Documents menu to the Mosaic menu bar. For more information, see the documentation about creating a configurable Documents menu. (Default is "/usr/local/lib/mosaic/documents.menu".)
fancySelections: Boolean	Cuts and pastes from the Document View window with fancy selections enabled in the HTML widget, which results in underlined headers, bulleted lists, and so on, if set to true. Uses normal selection behavior if set to false. Note that selections can be for each window at any time. (Default is false.)
fullHostname: string	Allows explicit setting of the fully qualified host name for systems that fail on a call to gethostbyname( ). Use the fullHostname resource only if necessary, in conjunction with gethostbynameIsEvil.
gethostbynameIsEvil: Boolean	Set to true if you are running your system on a Sun set up so that when the system attempts to find its own full host name through gethostbyname( ), the system coredumps. (Default is false.)
globalExtensionMap: string	Set to the location of the system-wide extension map config file of your choice. (Default is "/usr/local/lib/mosaic/mime.types".)
globalHistoryFile: string	Specifies the name of the file to store the global history of all documents accessed from session to session. (Default is .mosaic-global-history. The value of the environment variable $HOME will be prepended to this string.)
globalTypeMap: string	Specifies the location of the system-wide mailcap (type map config) file of your choice. (Default is "/usr/local/lib/mosaic/mailcap".)

(continued on next page)

Table 1–4 (Cont.) Functional Resources

Resource	Description
gunzipCommand: string	Specifies the command to uncompress gzip files (that is, files with an extension of .z.) You can obtain gunzip from prep.ai.mit.edu in /pub/gnu. (Default is gunzip -n -f, which requires the 1.2.4 version of gzip.)
hdfMaxImageDimension: integer	Specifies the maximum height and/or width of an HDF inlined image in pixels. (Default is 400.)
hdfMaxDisplayedDatasets: integer	Specifies the maximum number of displayed data sets while browsing HDF files. If more than the maximum number exist in an HDF file, Mosaic displays the HDF file in "brief mode."" (Default is 15.)
hdfMaxDisplayedAttributes: integer	Specifies the maximum number of displayed attributes while browsing HDF files. If more than the maximum number exist in an HDF file, Mosaic displays the HDF file in "brief mode." (Default is 10.)
hdfPowerUser: Boolean	Set to true if most of the supporting text in an HDF file will be removed. (Default is false.)
homeDocument: string	Specifies the document to access at program startup. See the information about on specifying your home document. (Default is http://www.ncsa.uiuc.edu/SDG /Software/Mosaic/NCSAMosaicHome.html)
imageCacheSize: integer	Specifies the size of the inlined-image cache in kilobytes. The -ics command-line flag can also be used. All the images in a given document are (temporarily) cached while on that page, regardless of the imageCacheSize value. Cache flushes the least recently viewed. (Default is 2048.)
initialWindowIconic: Boolean	Set to true if the first Document View window opened (when the program starts) will be iconified. (Default is false.)
maxWaisResponses: integer	Controls the maximum number of matches Mosaic accesses from a WAIS server. (Default is 200.)
personalAnnotationDirectory: string	Specifies the name of the directory in which to store personal annotations (as well as the log file of all personal annotations currently in existence). If the named directory does not exist, it is created. However, intermediate directories, if any are named, are not created. (Default is .mosaic-personal-annotations. The value of the environment variable $HOME will be prepended to this string.)
personalExtensionMap: string	Specifies the location of the personal extension map configuration file of your choice. The value of the environment variable HOME is prepended. (Default is .mime.types)
personalTypeMap: string	Specifies the location of the personal mailcap file of your choice. The value of the environment variable HOME is prepended. (Default is .mailcap)

(continued on next page)

Table 1–4 (Cont.) Functional Resources

Resource	Description
printCommand: string	Specifies the name of the default command for the Print menu option. This command accepts a single argument: the name of the file to print. Note that this location can be changed at any time through the Print option's dialog box. (Default is lpr)
recordCommandLocation: string	Specifies the location (full path name) of the command used to record audio annotations (on Sun, SGI, and HP platforms). If the named command does not exist when Mosaic is started, audio annotations are disabled (the menu entry becomes insensitive). (Default is "/usr /sbin/recordaiff" on SGI; "/usr/demo/SOUND/record" on Sun; "/usr/audio/bin/srecorder" on HP.)
recordCommand: string	Specifies the command used to record audio annotations; this is the full command, using the command named in recordCommandLocation as the first word. (Default is "recordaiff -n 1 -s 8 -r 8000" on SGI; "record" on Sun; "srecorder -au" on HP. The named command accepts a single additional argument: the file name of the new audio file. The command also correctly terminates recording when a SIGINT is sent.)
reloadReloadsImages: Boolean	Clears out the Reload option cached image data for the current document before reloading the document if set to true; thus causing the images to also be reloaded. This is useful for HTML authors who are using Mosaic to view their work. (Default is false.)
sendmailCommand: string	Specifies a pointer to the sendmail binary. It assumes that this program accepts command-line arguments that specify addresses to which messages should be mailed and accepts other headers and message text from stdin. (Default is "/usr/lib/sendmail".)
simpleInterface: Boolean	Allows the Mosaic menu bar and bottom button configuration to be minimized considerably. (Default is false).
tmpDirectory: string	Specifies the name of the directory in which to store temporary files generated by Mosaic. This directory should contain at least 10 or 20 megabytes, in case you to pull down a very large data file. Setting this resource is equivalent to using the command-line flag, -tmpdir, or setting the environment variable TMPDIR.
trackFullURLs: Boolean	Enables prototypical smart display of "where you're going" during mouse tracking, including format definition, if set to false. (Default is true.)
trackPointerMotion: Boolean	Sets the mouse pointer so that it will not be tracked so that URLs are visible on the lower status line, if set to false. (Default is true.)
trackVisitedAnchors: Boolean	Displays visited anchors (that is, anchors that point to documents that have previously been viewed) with a different style than anchors that are not yet visited, if set to true. (On slow systems, changing this resource to false may substantially improve the performance of Mosaic.) (Default is true.)

(continued on next page)

Table 1–4 (Cont.) Functional Resources

Resource	Description
tweakGopherTypes: Boolean	Mosaic interprets Gopher documents strictly by the Gopher typing system, if set to false. When set to true, Mosaic uses its own mechanism to determine the file types of Gopher documents. (Default is true.)
twirlIncrement: integer	Indicates the minimum number of bytes that are are transferred between updates of the twirling transfer icon for normal (FTP, Gopher, HTTP) transfers. (Default is 4096.)
twirlingTransferIcon: Boolean	Set to false to turn off the twirling NCSA logo during document transfers. Clicking the icon still interrupts a transfer. (Default is true.)
uncompressCommand: string	Specifies the command to uncompress compressed files (that is, files with an extension of .Z). (Default is uncompress).
useDefaultExtensionMap: Boolean	Set to false to keep any default file-extension-to-MIME-type mappings from Mosaic. Keeping this resource setting to true and overriding the default as necessary is strongly recommended. (Default is true.)
useDefaultTypeMap: Boolean	Set to false to keep any default MIME-type-to-external-viewer mappings from Mosaic. It is strongly recommended that you leave this resource set to true; override the setting as necessary. (Default is true.)
useGlobalHistory: Boolean	Automatically stores a global history of everywhere you visit, if set to true. (This history is used to shade anchors in different colors based on whether you visited the corresponding documents, if the resource trackVisitedAnchors is set to true.) (Default is true.)
xtermCommand: string	Specifies the name of the command used to start a terminal window for a telnet session. (Default is xterm on most platforms; the default is aixterm -v on IBM AIX 3.x systems.)

Visual Resources

Table 1–5 lists the visual resources that are associated with the HTML widget that is used in the Document View window. Specify these resources in the file DECW$USER_DEFAULTS:MOSAIC.DAT in the following format:

```
Mosaic*anchorColor: cyan
```

Table 1–5 Visual Resources

Resource	Description
anchorColor: color	Specifies the color to shade anchors whose corresponding documents have not previously been visited. (Default is blue3.)
visitedAnchorColor: color	Specifies the color to shade anchors whose corresponding documents have previously been visited. (Default is violetred4.)

(continued on next page)

Table 1–5 (Cont.) Visual Resources

Resource	Description
activeAnchorFG: color	Specifies the color to shade anchors that are in the process of being activated. (Default is red.)
activeAnchorBG: color	Specifies the color to shade the background of the anchors that are in the process of being activated. (Default is grey80, which is the same color as the background in the browser.)
anchorUnderlines: integer	Specifies the number of lines to draw underneath the unvisited anchors (Valid values: 0, 1, 2, 3) (Default is 1.)
visitedAnchorUnderlines: integer	Specifies the number of lines to draw underneath the visited anchors. Valid values: 0, 1, 2, 3 (Default is 1.)
dashedAnchorUnderlines: Boolean	Sets drawn lines under the unvisited anchors to dashes, if set to true. (Default is false.)
dashedVisitedAnchorUnderlines: Boolean	Sets drawn lines under unvisited anchors to dashes, if set to true. (Default is false.)
colorsPerInlinedImage: integer	Specifies the number of unique colors to allocate for each inlined-image. (Note that this setting does not affect the images that are displayed externally to the NCSA Mosaic browser (for example, GIF images passed to an external viewer).) (Default is 50.)
percentVerticalSpace: integer	Specifies the height of the vertical space between paragraphs, headers and paragraphs, and so on, as expressed as a percentage of the normal line height. (Default is 90.)
reverseInlinedBitmapColors: Boolean	Specifies to reverse foreground and background colors in inlined XBM bitmaps, if set to true. (Default is false.)
verticalScrollOnRight: Boolean	Specifies that the Document View window places its scroll bar on the left side, if set to false. (Default is true.)

Font Resources

Font resources, which are also associated with the HTML widget, can be changed dynamically from the Options menu. Table 1–6 lists the font resource names and descriptions.

Table 1–6 Font Resources

Resource	Description
font: font	Used in normal formatted text. (Default is "-adobe-times-medium-r-normal-*-17-*-*-*-*-*-iso8859-1".)
italicFont: font	Used in italic formatted text. (Default is "-adobe-times-medium-i-normal-*-17-*-*-*-*-*-iso8859-1".)
boldFont: font	Used in bold formatted text. (Default is "-adobe-times-bold-r-normal-*-17-*-*-*-*-*-iso8859-1".)
fixedFont: font	Used in fixed (typewriter style) formatted text. (Default is "-adobe-courier-medium-r-normal-*-17-*-*-*-*-*-iso8859-1".)

(continued on next page)

Table 1–6 (Cont.) Font Resources

Resource	Description
header1Font: font	Used in level 1 headers. (Default is "-adobe-times-bold-r-normal-*-24-*-*-*-*-*-iso8859-1".)
header2Font: font	Used in level 2 headers. (Default is "-adobe-times-bold-r-normal-*-18-*-*-*-*-*-iso8859-1".)
header3Font: font	Used in level 3 headers. (Default is "-adobe-times-bold-r-normal-*-17-*-*-*-*-*-iso8859-1".)
header4Font: font	Used in level 4 headers. (Default is "-adobe-times-bold-r-normal-*-14-*-*-*-*-*-iso8859-1".)
header5Font: font	Used in level 5 headers. (Default is "-adobe-times-bold-r-normal-*-12-*-*-*-*-*-iso8859-1".)
header6Font: font	Used in level 6 headers. (Default is "-adobe-times-bold-r-normal-*-10-*-*-*-*-*-iso8859-1".)
addressFont: font	Used in addresses. (Default is "-adobe-times-medium-i-normal-*-17-*-*-*-*-*-iso8859-1".)
plainFont: font	Used in plaintext regions or preformatted documents. (Default is "-adobe-courier-medium-r-normal-*-14-*-*-*-*-*-iso8859-1".)
listingFont: font	Used in listing regions. (Default is "-adobe-courier-medium-r-normal-*-12-*-*-*-*-*-iso8859-1".)
fixedboldFont: font	Used for bold in fixed (typewriter style) formatted text.
fixeditalicFont: font	Used for italics in fixed (typewriter style) formatted text.
plainboldFont: font	Used for bold in plaintext regions or preformatted documents.
plainitalicFont: font	Used for italic in plaintext regions or preformatted documents.

Proxy Gateway Resources

Support for proxy gateways is still experimental. The resources listed at the following URL may be pointers to proxy gateways:

```
   Mosaic*httpProxy: http://www.cern.ch:911/
```

Table 1–7 provides additional details about proxy gateway support in the NCSA Mosaic Version 2.4 browser.

Table 1–7 Proxy Gateway Resources

Resource	Description
ftpProxy: String	Used for ftp URLs.
gopherProxy: String	Used for gopher URLs.
httpProxy: String	Used for http URLs.
newsProxy: String	Used for news URLs.
waisProxy: String	Used for wais URLs.
ProxyGateway: String	Used for all access methods.

1.7.5 Restrictions for Using Mosaic from Behind a Firewall

V1.2–3

The following sections present information about using the ProxyGateway and NoProxy resources with Mosaic to access information on the Internet when security restrictions (that is, a firewall) exist between internal and external network access at your site.

1.7.5.1 ProxyGateway Resource

NCSA Mosaic Version 2.4 provided in DECwindows Motif Version 1.2–3 for OpenVMS supports the proxy gateway (ProxyGateway) resource. A **proxy gateway** is a trusted agent that enables Mosaic to access network information that is either within or outside the firewall, but not both.

When the Mosaic browser is used from behind a firewall, Mosaic can use the proxy gateway to pass network requests to the Internet in a URL formatted address. The proxy gateway returns the results to the Mosaic browser. The process of passing and returning network requests is transparent to the user. You can view documents on the Internet without difficulty.

Note, however, that if the proxy gateway is located outside of your firewall, then information behind the firewall (such as home pages, World Wide Web documents, and Notes files) cannot be accessed by the proxy gateway.

1.7.5.2 NoProxy Resource

If your site requires the ability to access Internet information both within and outside the firewall, then a version of Mosaic that supports the NoProxy resource is required. This resource allows you to specify a list of domains that should **not** be consulted by the proxy gateway (that is, access to information in these domains does not go through the proxy gateway).

NCSA Mosaic Version 2.4a for OpenVMS supports the NoProxy resource and enables access to a proxy gateway. If your site operates in a firewall environment and you need to access information from both within and outside of the firewall, you can copy this pre-release version of Mosaic from the following URL:

```
ftp://gatekeeper.dec.com/pub/DEC/Mosaic/
```

As previously stated, the NoProxy resource restricts access for the proxy gateway to a set of domains that are not consulted by the proxy gateway. You can specify the NoProxy resource, provided that you have NCSA Mosaic Version 2.4a for OpenVMS, in the file DECW$USER_DEFAULTS:MOSAIC.DAT according to the following format:

```
Mosaic*ProxyGateway:    http://www-proxy.site.org:8080/
Mosaic*NoProxy:         localhost, site.org
```

Where:	Description
www-proxy.site.org	Name of the host that runs the proxy server
8080	Port number through which Mosaic communicates to the proxy server through the firewall
local host, site.org	List of site names, which are separated by commas, excluded from going through the proxy server

Refer to the following location (URL) for additional information about proxy gateways:

```
http://info.cern.ch/hypertext/WWW/Daemon/user/Guide.html
```

1.8 Console Window Application

V1.2–3 DECwindows Motif Version 1.2–3 for OpenVMS introduces the feature of displaying console messages in the Console Window application. Console messages are not displayed in the console window by default in the DECwindows Motif Version 1.2–3 for OpenVMS product. Previous versions of DECwindows Motif displayed the console window by default.

Note

The new default for displaying console messages in the DECwindows Motif Version 1.2–3 for OpenVMS release is DISABLE. The default in previous versions of DECwindows Motif was ENABLE. These values are discussed in greater detail later in this section. If the user selects the Alternate Console port for console communications, the DECwindows Console Window is disabled and the console broadcasts are enabled. Refer to the owner's guide for your workstation for information about selecting the Alternate Console port.

Specify how to display messages by defining the global symbol DECW$CONSOLE_SELECTION in the customized startup file SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM. Enter one of the following values: WINDOW, DISABLE, or ENABLE.

- WINDOW

  Displays console messages in the Console Window application. This is a new application available with the DECwindows Motif Version 1.2–3 for OpenVMS software. If you specify the WINDOW value, the Console Window is displayed in the lower right corner of the login screen by default and continues to be displayed after the user logs in to the system.

  The Console Window application shares the same executable file and looks similar to the Message Window. However, a menu bar is not displayed in the Console Window; it reads its resources from the DECW$CONSOLE.DAT file instead of from the DECW$MESSAGEPANEL.DAT file. Internally, the Console Window is invoked by running the DECW$MESSAGEPANEL.EXE executable with the command line option *-console*.

  To control the initial position of the Console Window and the classes of OPCOM output that are enabled, you can the define the following global symbol in the file SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM:

  DECW$CONSOLE_GEOMETRY

  This symbol specifies the value of the *-geometry* option in the DECW$MESSAGEPANEL.EXE command line; this command is used to start the Console Window application. The default value is "-0-0", which specifies the location of the window in the lower right corner of the screen.

  To position the window at the lower left corner of the screen, for example, add the following line to the command file SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM:

  ```
  $ DECW$CONSOLE_GEOMETRY == "+0-0"
  ```

- DISABLE (default)

Disables broadcasts to the OPA0: device. Console messages are not displayed.

- ENABLE

  Displays console messages in the console window. The console window is a six-line display area at the top of the workstation screen.

Note

Although ENABLE was the default value in previous releases of DECwindows Motif, it is recommended that you do not use this option in the DECwindows Motif Version 1.2–3 for OpenVMS product. Displaying console messages by default in the console window can corrupt the contents of the workstation display.

Refer to *Managing DECwindows Motif for OpenVMS Systems* for information about defining global symbols in the file SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM.

1.9 Using Shareable Linkages to Install Images

V1.2–3

Alpha

On OpenVMS Alpha systems, using shareable linkages to install images on DECwindows Motif Version 1.2–3 software offers the following features:

- Enhanced startup performance by decreasing image-activation time
- Conserved memory usage by decreasing the total of memory pages used by the image

By default, the following images are installed using shareable linkages on DECwindows Motif Version 1.2–3 for OpenVMS Alpha systems:

- DECW$XLIBSHR.EXE (MIT Release 5 Xlib)
- DECW$XTLIBSHRR5.EXE (MIT Release 5 Intrinstics)
- DECW$XMLIBSHR12.EXE (OSF/Motif Release 1.2.3 Toolkit)
- DECW$MRMLIBSHR12.EXE (OSF/Motif Release 1.2.3 Motif Resource Manager)
- DECW$DXMLIBSHR12.EXE (Digital Extensions to the OSF/Motif Toolkit)

These images are installed using the /SHARE=ADDRESS_DATA option. Note that if you accept the default, these images cannot be replaced during a restart of DECwindows Motif software. As a result, when you restart DECwindows Motif, the images are not replaced and the following message is displayed:

```
Shared linkage sections are in use on this system and no images will be
reinstalled. If you are restarting DECwindows to reinstall images then you
must reboot the system.
```

To replace new images installed with this option, reboot the system. Note that you can disable shared-linkage sections by defining the logical name DECW$IGNORE_SHARED_ADDRESS in the SYSTARTUP_VMS.COM command procedure.♦

1.10 Window Manager Overlay Support

V1.2–3

The latest version of the Window Manager (MWM) is modified to support overlays and utilize additional planes of memory, which are available on several 3D graphics accelerators: ZLX-M1, ZLX-M2, ZLX- L1, ZLX-E2, ZLX-E3, ZLXp-E2, and ZLXp-E3. The Window Manager places borders and banners for all the windows into these extra planes of memory and thereby reduces the number of expose events for your applications that use overlays.

Modify your existing applications that use overlays to avoid potential problems with the colormap. Set up your system to share the overlay colormap with the Window Manager, as the hardware supports only one colormap for the overlay planes.

To modify your applications to share the overlay colormap with the Window Manager, query the server property name SERVER_OVERLAY_COLORMAPS. When you make the query, the server returns the 32-bit value for the overlay Colormap ID.

To set up your system to share the overlay colormap with the Window Manager, edit the files SYS$COMMON:[VUE$LIBRARY.SYSTEM]VUE$MWM.COM and SYS$COMMON:[SYSMGR]DECW$MWM.COM. Change the following line in each file:

```
$ mwm -multiscreen
```

Edit this line by adding the -Overlay command-line option as follows:

```
$ mwm -multiscreen "-Overlay"
```

Note that if you create and install your own colormap, the following problems can result:

- Colors flash on the screen when the colormap is changed.
- Border and banner colors also change when you change the colors of your colormap.

Restrictions

The following restrictions apply when you enable the Window Manager to use overlays:

- The Window Manager supports only single-screen systems and does not function correctly with multiple graphics devices (multihead).
- If you select a Matte Size value other than “None” from the Window Manager options list, the Matte color is not correct; that is, the color does not match the selection and is occasionally transparent.
- If you select “Show feedback when moving or resizing windows” from the Workspace Options menu, the window with the feedback information causes expose events.
- When you move windows by showing the outline of the window, the outline appears to go below the window borders and banners.
- Window borders are occasionally and randomly displayed in clear or black. If this problem occurs, select the restart option from the Workspace menu to restart the Window Manager.

- The ZLXp-E2 and ZLXp-E3 3D graphics accelerators are not currently supported when you run the Window Manager with the -Overlay command-line option.

2

General User Release Notes

This chapter contains information about DECwindows Motif for general users.

2.1 Notes Specific to DECwindows Motif Version 1.2–3 for OpenVMS Software

V1.2–3 The release notes in this chapter are cumulative from VMS DECwindows Motif Version 1.0 and still apply to DECwindows Motif Version 1.2–3 for OpenVMS software. The following sections contain general user release notes that pertain specifically to the DECwindows Motif Version 1.2–3 for OpenVMS release:

- Section 2.3, Using Tear-Off Menus
- Section 2.5, Toolkit Warning in the Work in Progress Dialog Box
- Section 2.6.1.1, Draft-Quality Printing
- Section 2.6.2, Including Comment Characters in the DECW$BOOKSHELF File
- Section 2.6.5, CDA Run-Time Services
- Section 2.6.9.1, DECsound Prerequisites
- Section 2.6.9.3, Escape Sequences
- Section 2.6.10.1, Changing the Auto Repeat Setting
- Section 2.6.10.2, Positioning DECterm
- Section 2.6.10.11, Initializing DECterm
- Section 2.6.12.1, Removing Support for LinkWorks Manager (DEClinks)
- Section 2.6.13, DECwindows Mail
- Section 2.6.13.1, Responses to Keyboard Actions
- Section 2.6.16.1, Notepad Is Linked with the OSF/Motif Release 1.1.3 Toolkit
- Section 2.6.19.1, Removing Support for LinkWorks Manager and LinkWorks
- Section 2.6.19.2, Invoking DECchart from the Session Manager Applications Menu
- Section 2.10, Unsupported Translations by the Motif XmText Widget

2.2 Using Drag and Drop

V1.2 The DECwindows Motif for OpenVMS applications support the drag-and-drop feature with the exception of Notepad. DECwindows Mail supports the drag-and-drop feature in all windows except the main message area, where DECwindows Mail has its own drag-and-drop; you can use MB2 to move messages around with the SVN interface.

The drag-and-drop feature lets you move or copy screen objects; this feature is provided primarily for programmers who choose to incorporate drag-and-drop into their applications. For example, you can move text from buttons and paste it elsewhere.

To drag and drop text into a new location:

1. Select the text to be copied or moved with MB1.
2. To move the text, press and hold MB2; to copy the text, press and hold Ctrl/MB2.

   A move or copy icon is displayed.
3. Drag the icon to the location where you want to drop the text and release MB2.

   If the object is highlighted as you drag the icon across it, you can drop the text into that location.

2.3 Using Tear-Off Menus

V1.2–3 DECwindows Mail supports tear-off menus in the DECwindows Motif Version 1.2–3 for OpenVMS release.

V1.2 The following applications do not support tear-off menus:

- CDA Viewer
- Notepad
- Print Screen

Refer to Section 5.3.2.2 for instructions on tearing off a menu and closing a tear-off menu.

2.4 Access Control Not Explicitly Enabled by Default

V1.0 DECwindows Motif does not enable access control by default. Instead, the product uses access control set by the server. The DECwindows X11 display server enables access control at startup time.

To force the DECwindows Session Manager to enable or disable access control explicitly at login time, you can define one of the following logical names:

```
$ DEFINE/SYSTEM/EXECUTIVE DECW$LOGIN_ACCESS_CONTROL ENABLE
```

```
$ DEFINE/SYSTEM/EXECUTIVE DECW$LOGIN_ACCESS_CONTROL DISABLE
```

If the logical name is not defined or if it is defined to some other value, such as “SERVER”, DECwindows login neither enables nor disables access control.

In most cases, it should not be necessary to define the logical name.

2.5 Toolkit Warning in the Work in Progress Dialog Box

V1.2–3 When you run DECwindows Motif Version 1.2–3 for OpenVMS software on OpenVMS Version 6.2 systems, the Work in Progress dialog box displays the following error message in several applications:

```
X Toolkit Warning: locale not supported by C Library, locale unchanged
```

You can safely ignore this error message; operations of the applications are not affected.

2.6 DECwindows Motif Application Notes

This section contains information about DECwindows Motif applications.

2.6.1 Bookreader

This section includes problems, restrictions, and enhancements in the Bookreader application.

2.6.1.1 Draft-Quality Printing

V1.2–3 The Bookreader application allows only draft-quality printing for books or topics.

2.6.2 Including Comment Characters in the DECW$BOOKSHELF File

V1.2–3 If you include a comment character (! or #) in the last line of the DECW$BOOKSHELF file, Bookreader fails with a reserved operand fault. As a workaround, ensure that you do not add comment characters to the last line of the file.

This restriction will be removed in a future version of the product.

2.6.3 Calendar

This section describes problems and restrictions with the Calendar application.

2.6.3.1 Data-File Format

V1.2 The Calendar data-file format in DECwindows Motif for OpenVMS is incompatible with previous Calendar formats. When you run the Calendar application in DECwindows Motif Version 1.2 for OpenVMS for the first time, a dialog box in the Calendar application is displayed that asks whether to convert the Calendar data file to the new format. If you answer “Yes”, the data files are converted. Once the data file is converted to the new format, the file is no longer usable by previous (that is, pre-Version 1.2) versions of Calendar. If you answer “No”, the data files are not converted and the Calendar application cannot be used.

To prevent this problem, ensure that all the nodes in a cluster are running the same version of DECwindows Motif.

2.6.3.2 Additional Problems and Restrictions

V1.2 The following are known problems with the Calendar application:

- When you initially run the Calendar application after you install DECwindows Motif for OpenVMS software, Calendar displays a message dialog box stating that the Calendar database file needs to be upgraded to the new format. If the Motif Window Manager is listed before Calendar in the Session Manager Automatic Startup list, then the message dialog box will be small and unreadable. As a possible solution, move Calendar before the Motif Window Manager in the Automatic Startup list.

- If the Window Manager buttons are customized so that an action is taken when the pointer is in the window and you press MB1, double clicking on the Month display is disabled.

2.6.4 Cardfiler

V1.0 You cannot modify or save DECwindows Motif card files from a DECwindows X User Interface (XUI) version of the Cardfiler.

The DECwindows XUI version of Cardfiler can be used to read DECwindows Motif card files. Information in the old card file format is compatible with the DECwindows Motif Version 1.0 Cardfiler. DECwindows XUI card files are changed to Motif format when they are read in and saved using the DECwindows Motif Cardfiler.

2.6.5 CDA Run-Time Services

V1.2–3 The DECwindows CDA Viewer supports both Versions 1.2 and 1.3 of the DECfonts product.

V1.2 This section describes features in the CDA Run-time Services component:

- Pack and Unpack applications are provided. See Section 2.6.6.1
- Performance enhancements for viewing CDA and PostScript documents.
- Dynamic font support is implemented. Future upgrades to installed fonts are usable by the CDA Viewer and the PostScript back end.
- The DECwindows CDA Viewer supports DECfonts Version 1.2.
- CDA support is improved for documents created by DECwrite that contain links to other documents.

You can find additional information about CDA in the following manuals:

- *Digital CDA Base Services Reference Manual*

  Provides reference material for using the CDA data structures and access routines to create compound document applications, converters, and viewers.
- *Technical Overview of the Digital CDA Base Services*

  Provides an introduction to CDA, compound document processing concepts, and using the DEC CDA Base Services.

2.6.5.1 CDA Dynamic Font Support

V1.2 As well as supporting a static-table for the fonts supported by the DECfonts Typeface Collection Version 1.2, this version of CDA Run-Time Services includes support for dynamic font lookup. This enables the CDA Viewer and the PostScript back-end converter to use new fonts as they are installed on the system.

Dynamic font support is implemented using the WRITE$FONTS.INI file, which you can maintain using the Font utility provided with either DECwrite or DECpresent. If a document contains a font not found in the static tables, the CDA Viewer or the PostScript back-end converter tries to open the WRITE$FONTS.INI file and search for the font. If the font is not found or if the system does not contain a WRITE$FONTS.INI file, the viewer uses a fallback font.

For further information on the Font utility, see the following documents:

- *DECwrite Installation Guide for VMS Systems*
- *DECpresent Installation Guide for VMS Systems*

- *DECfonts Typeface Collection and User's Guide for OpenVMS*

2.6.5.2 CDA WRITE$FONTS Logical Name

V1.2 The default location for the WRITE$FONTS.INI file is SYS$LIBRARY, but, if the logical name WRITE$FONTS is defined, the CDA Viewer or the PostScript back end uses the logical name definition to search for the WRITE$FONTS.INI file.

Full path support is included, so any of the following definitions are valid:

WRITE$FONTS Logical Name	Resulting File
Undefined	SYS$LIBRARY:WRITE$FONTS.INI
DISK:[DIRECTORY]	DISK:[DIRECTORY]WRITE$FONTS.INI
SYS$LOGIN:	SYS$LOGIN:WRITE$FONTS.INI
.TMP	SYS$LIBRARY:WRITE$FONTS.TMP
DISK:[DIRECTORY]FILE	DISK:[DIRECTORY]FILE.INI

2.6.5.3 CDA Documents

V1.2 The current version of CDA Run-Time Services includes a performance enhancement that decreases the time it takes to display the first page of a CDA document. Other applications that use the CDA Viewer to view documents (for example, DECwindows Mail) also benefit from this enhancement.

The CDA Viewer enables this performance enhancement feature by default. You can disable the feature as follows:

```
$ DEFINE CDA_QUICK_FIRST_PAGE FALSE
```

The CDA Viewer might not display some documents correctly when this feature is enabled. If you encounter such a problem, disable the feature and invoke the CDA Viewer again. If the problem persists, please submit a Software Performance Report along with a copy of the document (and any associated documents). The CDA Pack application (see Section 2.6.6.1) allows you to create a single file that includes a .DDIF or .DTIF document and its associated files.

2.6.5.4 CDA PostScript Documents

V1.2 The PostScript viewing feature of the CDA Viewer no longer reads the entire PostScript document before displaying the first page. Instead, it processes one page at a time. In the case of large documents, this significantly reduces the time needed to display the first page.

Note that since the PostScript file is no longer read in all at once, the total number of pages in the PostScript document is not known until the last page is read. Therefore, the current page indicator does not show the number of pages in the document until the entire PostScript document has been read by the CDA Viewer.

2.6.6 CDA—Additional Applications

V1.2 This section contains information about additional CDA applications.

2.6.6.1 Pack and Unpack Applications

V1.2

CDA Run-Time Services includes two standalone applications that can be used for transferring CDA documents across a network. The CDA Pack application packages a CDA document along with all of its externally referenced files into a single file that can be copied between systems or mailed to other users. The CDA Unpack application reads a file that is packaged by the CDA Pack application and creates a copy of the original document file and all its externally referenced files.

These applications allow you to copy CDA documents between systems without copying externally referenced files separately or correcting external file reference information after copying documents.

To use these applications, add the following lines to your LOGIN.COM file (or add the lines to the SYS$MANAGER:SYLOGIN.COM file):

```
$ PACK == "$SYS$SYSTEM:CDA$PACK.EXE"
$ UNPACK == "$SYS$SYSTEM:CDA$UNPACK.EXE"
```

These lines enable you to use the symbols PACK and UNPACK to invoke the Pack and Unpack applications, respectively.

2.6.6.2 Pack Application

V1.2

The CDA Pack application creates a single output file that contains the contents of a .DDIF or .DTIF input file. The single output file also includes the files that are referenced by the .DDIF or .DTIF input file.

The format of the PACK command is as follows:

```
$ PACK input-file-spec output-file-spec
```

The following sections explain the format of the PACK command.

input-file-spec
Specifies the name of the primary .DDIF or .DTIF input file.

output-file-spec
Specifies the name of the output file that is created by the PACK application. If you do not specify a device or directory, the output file is created in the current default directory.

Qualifiers:

/[NO]SKIP_MISSING
Controls whether the Pack application continues processing if it cannot find one or more of the files that are listed as external references in the input file. The names of any missing files are sent to SYS$ERROR when the Pack application is completed. If you specify /NOSKIP_MISSING, the Pack application does not create an output file if any of the externally referenced files are missing. The default is /SKIP_MISSING.

/[NO]CONTROLLED_COPY
Controls whether the output file includes only those external references that specify COPY_REFERENCE as the value of the ERF_CONTROL item in the input file. If you specify /NOCONTROLLED_COPY, the Pack application includes all referenced files, regardless of the value of the ERF_CONTROL item. The default is /NOCONTROLLED_COPY.

/ALWAYS_ENCODE

Controls whether an output file is created when there are no external references in the input file, or if none of the externally referenced files are found. The default is not to create an output file in these cases. If an output file is not created for these reasons, the Pack application returns the CDA_W_NOOUTFIL status code.

For example:

```
$ PACK MYFILE.DDIF TEST.PACK
```

2.6.6.3 Unpack Application

V1.2 The CDA Unpack application unpacks an input file created by the Pack application. The output files are the .DDIF or .DTIF file that is packed by the Pack application, as well as a file for each external reference in the .DDIF or .DTIF file. The Unpack application sends a list of created files to SYS$ERROR.

The format of the UNPACK command is as follows:

```
$ UNPACK input-file-spec
```

The following sections explain the format of the UNPACK command.

input-file-spec

Specifies the name of the input file that is created by the Pack application.

Qualifier:

/OUTPUT=output-file-spec

Specifies the file name and location of the files created by the Unpack application.

If you specify an output file name without a directory name, the Unpack application creates the main .DDIF or .DTIF file with the file name you specify in the current default directory. It also creates all externally referenced files in the current default directory.

If you specify a directory name without a file name, the Unpack application creates the main .DDIF or .DTIF file and all externally referenced files in the specified directory. The main .DDIF or .DTIF file has the same name as the file packed by the Pack application.

If you specify a directory name and a file name, the Unpack application creates the main .DDIF or .DTIF file and all the externally referenced files, in the specified directory. The main .DDIF or .DTIF file has the file name you specify.

For example:

```
$ UNPACK TEST.PACK
Output file DISK$:[SMITH]MYFILE.DDIF created.
Output file DISK$:[SMITH]FIGURE_1.DDIF created.

$ UNPACK TEST.PACK/OUTPUT=[SMITH.UNPACK]
Output file DISK$:[SMITH.UNPACK]MYFILE.DDIF created.
Output file DISK$:[SMITH.UNPACK]FIGURE_1.DDIF created.

$ UNPACK TEST.PACK/OUTPUT=[SMITH.UNPACK]NEW_FILE.DDIF
Output file DISK$:[SMITH.UNPACK]MYFILE.DDIF renamed NEW_FILE.DDIF
Output file DISK$:[SMITH.UNPACK]FIGURE_1.DDIF created.
```

2.6.6.4 Pack and Unpack Error Messages

V1.2 This section describes messages associated with the CDA Pack and Unpack applications.

FILESPEC, Missing filespec: *file-name*

Severity: Informational

Explanation: The Pack application cannot locate an external file included as an external reference in the .DDIF or .DTIF file or in one of the files referenced in the .DDIF or .DTIF file.

NOOUTFIL, No output file was created.

Severity: Warning

Explanation: The Pack application cannot find external references in the .DDIF or .DTIF document to be packed, and you did not specify the /ALWAYS_ENCODE qualifier.

OUTFILE, Output file created: *file-name*

Severity: Warning

Source: CDA_UNPACK

Explanation: The Unpack application created the specified file while unpacking a file created by the Pack application.

2.6.7 CDA—Hints, Restrictions, and Known Problems

This section contains helpful hints, known restrictions, and known problems with the CDA Viewer.

2.6.7.1 Display PostScript with the CDA Viewer

V1.2 Previously, the CDA Viewer had a compatibility problem with the previous version of the DECwindows Display PostScript extensions.

This problem sometimes caused parts of the data displayed on a page to be rotated, shifted, or incorrectly scaled—sometimes to an extent that the data was no longer visible on the screen. The first page of some documents appeared blank but became visible after displaying the second page.

The problem is resolved in the latest version of the CDA Viewer; however, if a previous version of the CDA Viewer is run on one system (the client) with display set to another system (the server) and the server has the new Display PostScript engine, then the problem still exists. The solution is to upgrade the version of CDA on the client system.

2.6.7.2 Message for CDA Viewer

V1.2 The CDA Viewer issues the following message if it is unable to create the application context:

DRMCTXFAIL, DVR could not create application context, aborting

Level: Error

Explanation: The CDA Viewer ends because an attempt to create the application context using the Resource Manager failed, which is usually caused by insufficient memory.

User Action: Reduce the system load and start the application again.

2.6.7.3 CDA Paper Size Button Renamed

V1.2 In the CDA Viewer, the name of the Paper Size button in the Open dialog box is changed to Display Options. However, the function invoked by the button is not changed.

2.6.7.4 CDA Watch Progress Restriction

V1.2 If you use the CDA Viewer to view a PostScript document and you enable the Watch Progress feature, then manipulating CDA Viewer buttons (other than the Cancel button) or scroll bars while the wait cursor is active can cause corruption of the display.

2.6.7.5 CDA Viewer Supports DECfonts Version 1.2

V1.1 The CDA Viewer supports DECfonts Versions 1.2 and 1.3.

The Linotext fonts were renamed following DECfonts Version 1.1 and before Version 1.2. If a .DDIF file contains DECfonts Version 1.1 Linotext text, the CDA Viewer does not find it and defaults to Courier 12 point for this text.

To correct this problem, modify the font using DECwrite or DECpresent in conjunction with DECfonts Version 1.2.

2.6.7.6 CDA Viewer—Viewing PostScript Files with Errors

V1.0 If you use the CDA Viewer to view a PostScript file that contains syntax errors, the CDA Viewer can stop processing without displaying an error message until you click on the Cancel button. This problem also occurs if you specify a .PS formatted file other than PostScript to the viewer. This problem can occur in the CDA Viewer application or in any application that uses the callable CDA Viewer interface in DDIF$VIEWSHR.

Specify only valid PostScript files to the CDA Viewer when specifying the .PS format.

2.6.8 Clock—DECsound Capability

V1.2 The Clock application includes an alarm feature that can be used if your system is a TURBOchanel system that supports DECsound. On systems without sound capabilities, you can select only the keyboard bell. When you choose Alarm from the Options menu, a pop-up window appears. This pop-up window allows you to set the alarm time, choose the sound to be played, and indicate an alarm message. To see if your system supports this feature, invoke one of the sounds located in the DECW$EXAMPLES directory (for example, BELLS.AUD).

2.6.9 DECsound Application

This section contains information about DECsound.

2.6.9.1 DECsound Prerequisites

V1.2–3 DECsound is supported only on TURBOchannel systems. TURBOchannel systems that support DECsound include the following:

- DEC 3000 Model 400 series
- DEC 3000 Model 500 series
- DEC 3000 Model 700 series
- DEC 3000 Model 900 series
- VAXstation 4000 Model VLC, 60 and 90

DECsound is not supported on the following systems:

- DEC 2000 Model 300 series
- AlphaStation 200
- AlphaStation 400

2.6.9.2 DECsound Problems

V1.1 The following are known problems with DECsound:

- Links to audio files created using DECwrite Version 2.0 and DECpresent Version 1.0 cannot be played back using the CDA Viewer because these applications currently store links to audio files as private data that cannot be accessed by other CDA applications.

  To correct this problem, use DECwrite or DECpresent to view documents with links to audio recordings.
- The Record operation does not work when you select part of the waveform.

  To correct this problem, do not select any part of the waveform before you click on the Record button.

2.6.9.3 Escape Sequences

V1.2–3 The DECterm application supports all ReGIS input cursors:

- Crosshair
- Diamond
- Rubber-band line
- Rubber-band rectangle

For a shape other than the diamond cursor when *n* is equal to 1, define the logical name DECW$TERM_REGIS_CURSOR as one of the numbers defined in the SYS$LIBRARY:DECW$CURSOR file.

V1.2 The following escape sequences are supported by DECterm:

- All page movement sequences (NP, PP, PPA, PPB, and PPR).
- One rectangular area operation sequence (DECCRA).
- The DECLFKC sequence.
- The ReGIS command S(C(In)) supports the rubber-band rectangle cursor and the diamond cursor.

See Section 4.13.4, ReGIS Input Cursors for additional information about escape sequences in DECwindows Motif software. See Chapter 4 for details and restrictions on the use of these sequences.

2.6.10 DECterm—Restrictions and Problems

This section discusses restrictions and problems related to the DECterm application.

2.6.10.1 Changing the Auto Repeat Setting

V1.2–3 You cannot change the Auto Repeat setting in the DECterm Keyboard Options dialog box. Instead, change the setting in the Keyboard dialog box in the Session Manager.

2.6.10.2 Positioning DECterm

V1.2–3

If the resource Mwm*clientAutoPlace is set to True, DECterm windows are not displayed where the DECW$TERMINAL.x and DECW$TERMINAL.y resources are specified. As a workaround, set this resource to False and restart MWM.

2.6.10.3 User Font Selection

V1.2

DECterm allows you to select available fonts from the Options/Window dialog box. A different font can be selected to replace the default Big Font, default Little Font, or default German Standard Font.

In the Options/Window dialog box, you can select the default font or enter the name of a font into a text field and select that font. When the Big Font, Little Font, or German Standard Font is selected and the Other toggle button for the font is selected, DECterm attempts to locate and use the font name supplied in the adjacent text field. For information on font naming conventions or fonts available on your system, consult your system manager.

Note

Not all fonts work properly with DECterm. DECterm emulates a character-cell terminal, and, therefore, expects the fonts to be monospaced (each glyph occupies the same number of pixels). Proportional fonts can be chosen, but they produce unpredictable results.

Furthermore, DECterm fonts are supplied in families of 26 related fonts, including variations for bold, double-width, double-width/double-height, normal, and condensed characters. Also, DECterm fonts include special characters, such as the Line Drawing and the DEC Technical character sets. A DECterm font family is identified by the font naming convention.

Use of fonts that do not have all the related font family variations, fonts that do not have the needed special characters, or font families that do not adhere to the DECterm font family naming conventions, may not result in an optimal display or otherwise perform as expected.

On OpenVMS systems, Version 6.1 or higher, two additional fonts are available: the VT330 font and the VT font. The VT330 font is based on the 10x20 font used in the VT330 and VT340 terminals. The VT font is a 9x18 font derived from the VT330 font. Use these fonts on a system where they are available by entering the following font-name strings in the font-name text field:

```
-DEC-VT330-*-*-*--20-*-*-*-c-*-*-*
    -DEC-VT-*-*-*--18-*-*-*-c-*-*-*
```

2.6.10.4 Local Echo

V1.2

DECterm supports a local echo mode. In the Options/General dialog box, select Local Echo, which causes all character sequences generated locally to be echoed on the display and passed to the remote host. This feature is useful when connected to a host that does not echo typed characters.

2.6.10.5 Answerback Message

V1.2 A user interface is available to enter answerback messages. This answerback message field is for compatibility with Digital terminals.

The answerback field is a buffer that contains up to thirty characters. The answerback field in earlier Digital terminals contained a message used to identify itself to the host system. For DECterm windows, the answerback field can be used to store a sequence of characters that you can use for any repetitive purpose.

A field is provided in the Options/General dialog box to enter answerback text. Click on the answerback field and enter your text. To enter control characters, encode the control character as a two-digit hex ASCII code, preceded by a number sign (#).

For example, when you enter #0D in the answerback field, DECterm responds with a carriage return.

If two consecutive number sign characters are entered (##), a single number sign is transmitted.

If anything other than a valid two-digit hex code or another number sign is detected after an initial number sign, the number sign is treated as a normal text character. Refer to any ASCII table for a complete list of characters.

The answerback text can also be concealed. When the Conceal Answerback button is enabled, the answerback message is concealed. To deselect the Conceal Answerback button, click on the answerback text field, which erases the previous answerback message.

2.6.10.6 Seven-Bit Printer Support

V1.2 When the 7-Bit Printer button is selected in the Options/Printer dialog box, DECterm modifies printed text to be compatible with printers that do not support 8-bit characters. This includes modifying control sequence introducer (CSI) strings to use the format Escape-Left Bracket rather than the single 8-bit CSI character.

When the 8-Bit Printer button is selected, DECterm allows the use of 8-bit characters when printing. This mode can cause problems for older printers if they can not interpret 8-bit characters.

The default is 8-Bit.

2.6.10.7 VT330 and VT340 Terminal Emulation

V1.0 DECterm incorporates some of the features of the VT330- and VT340-series video terminals, such as ReGIS and Sixel graphics. DECterm does not provide complete VT330 and VT340 terminal emulation.

The following restrictions apply to DECterm:

- User-loadable characters (DRCS), local mode, and control representation mode (CRM) are not implemented.
- The checkerboard character (character 97 in the DEC Special Graphic character set) is used as an error character in place of the reverse question mark.
- DECterm uses replace mode as the default for Sixel drawing on servers with eight planes or less. On servers with more than eight planes, DECterm uses overlay mode; replace mode is not functional on those servers.

2.6.10.8 CREATE/TERMINAL/DETACHED/PROCESS

V1.0

The /PROCESS=*procnam* qualifier does not work when used in conjunction with the /DETACHED qualifier, unless there is already a process running on the system where its process name is equal to the user name.

To work around this problem, use the following command procedure:

```
$! CREATE_TERM_PROC.COM
$!
$! Invoke as SPAWN/NOWAIT @CREATE_TERM_PROC procname
$!
$ SET NOON
$!
$! Set Process name to username
$!
$ X = F$CONTEXT("PROCESS", PID, "PRCNAM", "’’F$PROCESS()’","EQL")
$ NAME = F$EDIT(F$GETJPI(X,"USERNAME"),"COLLAPSE")
$ SET PROCESS/NAME="’’NAME’"
$ CREATE/TERMINAL/DETACHED/PROCESS="’’P1’"
$!
$! Allow new process to RUN LOGINOUT before exiting subprocess
$ WAIT 00:00:10
```

If the /PROCESS=*procnam* qualifier specifies a process name that is already in use, the DECterm is created but creation of the process inside the DECterm fails. In this case, the DCL command CREATE/TERMINAL returns the following error message:

```
Duplicate process name
```

2.6.10.9 Printing to an Attached Printer

V1.1

Printing to a port device requires you to have read and write privileges on that port. You cannot print by allocating the device, since the controller requires access to the device. Instead, set the device to WORLD:RW.

For example, to use the printer port on a VAX 3100 system, enter the following command from a privileged account or include the command in the system startup file:

```
$ SET PROTECTION=WORLD:RW TTA3:/DEVICE
```

2.6.10.10 DECterm Graphics

V1.1

The following information is specific to DECterm graphics:

- In some cases, a private colormap is created in DECterm. This private colormap is created when ReGIS or sixel graphics are displayed in the window and a sufficient number of colors from the default colormap cannot be allocated. The result is that when the DECterm window has input focus, the colormap changes for the entire workstation. The default colormap is four colors on a four-plane or monochrome system and 16 colors on color systems with more than four planes.

  To restore a DECterm window to the default colormap, select Clear Display from the Commands menu to clear the window. Then select Reset Terminal from the Commands menu to reset the terminal.

- Only graphics, not text, are written to the graphics backing store. When part of a window has to be redrawn in DECterm, the graphics portion of the window is drawn first, then the text is overlaid. As a result, the redrawn window might not look the same as the original picture.

- ReGIS addresses the entire window, not just 24 rows and 80 columns, so the aspect ratio between text and graphics might not always be the same as on the VT330 or VT340 terminal.
- The following ReGIS features are not implemented:
  - Command Display mode
  - Scrolling
  - Output cursors

2.6.10.11 Initializing DECterm

V1.2–3 The workaround that is discussed in the Version 1.1 release note that follows is no longer necessary on OpenVMS Alpha Version 6.1 and OpenVMS VAX Version 6.2 (or higher) systems.

V1.1 To ensure that your DECterm windows do not shrink unexpectedly to the default size of 80 characters by 24 lines, systemwide and user login command procedures (SYLOGIN.COM and LOGIN.COM) should not execute the SET TERMINAL/INQUIRE command procedure on DECterm windows. Executing the SET TERMINAL/INQUIRE command on mailbox devices that are created from the Session Manager also prevents the Session Manager from starting applications such as DECterm.

You do not need to use the SET TERMINAL/INQUIRE command; the DECterm controller provides OpenVMS systems with the proper characteristics and size of DECterm windows.

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

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

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

2.6.10.12 DECterm Resource Usage

V1.1

You cannot create more terminal windows than your system resources and quotas allow. If you have insufficient resources, a dialog box is displayed with a message indicating that no additional DECterm windows can be created.

To reduce the memory requirement of each DECterm window and create additional terminal windows, decrease the number of Record Lines Off Top in the Display dialog box and decrease the number of columns for each DECterm window.

Once the resource limit is reached, log out of all DECterm windows that are running on the host system before you increase the number of terminal windows.

2.6.10.13 Diagnostic Crash File and Messages

V1.1

DECterm produces a diagnostic file when a status code of fatal is returned. The DECTERM_ERROR.LOG file is produced in the login directory when the DECterm application exits abnormally. If you have a problem with the DECterm application and submit a Software Performance Report (SPR), include a copy of the log file. Under certain circumstances, a log file is generated even when no problem is encountered. Therefore, the appearance of a log file as an isolated event should not be cause for a problem report.

You can enable additional levels of diagnostic messages by defining either a logical name or a symbol named DECTERM_DIAG. When defined, enhanced diagnostics are displayed by the DECterm images. Use this mode only for diagnosing problems; it causes Session Manager message windows to be generated for each new DECterm created from the Session Manager.

DECterm diagnostics can be captured in a file by defining the logical name DECW$TERMINAL_OUTPUT to point to a file.

2.6.10.14 Monitor Screen Width Determines Default Font Size

V1.0

Three font sets are available for 100 dots-per-inch monitors: 100, 140, and 180 decipoint fonts. When a 100-dpi monitor with a physical screen width of less than 325 millimeters is used (approximate screen width of a 15-inch monitor), the 140 and 100 decipoint fonts are selected, respectively, for the Big and Little font sets. If the physical screen width is larger than 325 millimeters, then the 180 and 140 decipoint fonts are used.

To use the smaller font sets on a large 100-dpi monitor, place the following lines in the DECW$TERMINAL_DEFAULTS.DAT file:

```
DECW$TERMINAL.main.terminal.littleFontSetName: -*-terminal-*-*-*--*-100-*-*-*-*-*-*

DECW$TERMINAL.main.terminal.bigFontSetName: -*-terminal-*-*-*--*-140-*-*-*-*-*-*
```

Alternatively, you can enter these font strings in the text fields for the “Other” fonts in the Options/Window dialog box.

2.6.10.15 ReGIS Locator Report

V1.0

When DECterm sends a ReGIS locator report in response to the R(P(I)) command, or in multiple input mode and the locator position is outside the addressable area, DECterm sends a locator report with the coordinates omitted. For example, press the A key to generate the report:

```
A[]<CR>, where <CR> is a carriage return (ASCII code 13).
```

2.6.11 DECW$CDPLAYER

V1.1 The DECW$CDPLAYER application in the DECW$EXAMPLES directory requires PHY_IO and DIAGNOSE privileges to operate the compact-disc player hardware. Either your process or the image must have these privileges.

2.6.12 LinkWorks

This section contains information about LinkWorks (DEClinks).

2.6.12.1 Removing Support for LinkWorks Manager (DEClinks)

V1.2–3 Support for DEClinks, formerly known as LinkWorks, is removed from the Session Manager and from the desktop applications menus and from the out-of-the-box (OOTB) desktop applications.

The verbs “LinkWorks Manager” and “LinkWorks Setup” are removed from the system profile file, VUE$SYSTEM_PROFILE.VUE$DAT. If users modified or moved these verbs, they are stored in their private profile file, VUE$PROFILE.VUE$DAT. Delete any customizations that reference these two verbs, either from autostart or as a private verb; these verbs are no longer supported. To eliminate these customizations, delete the verbs from Options Automatic Startup and from Options Menus dialogs.

2.6.13 DECwindows Mail

This section contains information about the DECwindows Mail application.

V1.2–3 Since the DECwindows Motif Version 1.2 release, performance of the DECwindows Mail application is enhanced.

2.6.13.1 Responses to Keyboard Actions

V1.2–3 Enhancements in the DECwindows Mail application to comply more closely with OSF/Motif style conventions cause some changes with application responses to keyboard actions. Note the following changes.

If you use the Tab key to advance through text entry boxes in the Create/Send window, the current field is no longer highlighted. To select a field, use one of the standard Motif actions, such as double or triple clicking MB1, or use the Shift+Alt → key sequence.

In several dialog boxes, press either the Select key or the space bar to activate a pushbutton through the keyboard. Note that the Return key and the Enter key are bound to other widgets in the dialog box. Alternatively, you can continue to click MB1 to activate a pushbutton.

2.6.13.2 Using the Color Customizer with DECwindows Mail

V1.2 If you use the color customizer sample program provided in the directory DECW$EXAMPLES to control DECwindows Mail colors, the DECwindows Mail color customization dialog boxes used to modify those colors may not reflect the correct current color values. This is normal behavior; use the color customizer instead of the DECwindows Mail color customization dialog boxes to change these values. Alternatively, exit from the color customizer and restart DECwindows Mail. See Section 3.4 for more information about the color customizer.

2.6.13.3 DECwindows Mail—Known Problems

V1.0

The following problems exist with the DECwindows Mail application:

- The XUI Window Manager does not automatically set input focus to windows when they are mapped. You must click on each window to get input focus.
- The next and previous arrow buttons in the Read window do not dim at the end or beginning of the folder, respectively. They are ineffective if selected.
- Use of the Message button in the Read window to scroll through a large message can be very slow.
- Performance of the text widget in the Create-Send window can be degraded if the word wrap option is turned on. To enhance performance, turn off the word wrap option.

2.6.14 DECwindows Motif Window Manager

This section contains information about the DECwindows Motif Window Manager.

2.6.14.1 Using the Color Customizer with DECwindows Motif Window Manager

V1.2

If you are using the color customizer provided in the DECW$EXAMPLES directory to control Motif Window Manager colors, the Motif Window Manager customization dialog boxes used to modify those colors may not reflect the correct current color values. This is normal behavior; use the color customizer instead of the Motif Window Manager color customization dialog boxes to change these values. Alternatively, exit the color customizer and then restart Motif Window Manager. See Section 3.4 for more information about the color customizer.

2.6.14.2 Configuration File

V1.0

The configuration file DECW$MWM_RC.DAT defines how the Window Manager uses the function keys. Most of the accelerators use the form Alt key (or Compose Character key) and function key, for example Alt+F7.

If any application needs to use these keys, you must either comment them out by placing an exclamation point (!) at the beginning of the line, or create new keyboard bindings. Then change Mwm*keyBindings:DefaultKeyBindings in the MWM resource file to point to the new bindings.

With the Motif binding, you can no longer use the Alt+spacebar or the Compose Character+spacebar to bring up the Window menu because it interferes with Compose Character sequences in DECterm. Use Shift+F11 to bring up the Window menu.

To reenable Alt+space, select the appropriate option in the Workspace Options dialog box and apply the current settings. You can also remove the comment for the default button bindings for Alt+space in the DECW$MWM_RC.DAT file.

2.6.14.3 DECwindows XUI Applications

V1.0

The resource Mwm*useDECMode allows previous versions of DECwindows XUI applications to behave correctly with the Motif Window Manager. In particular, this resource is used to control focus, window placement, multiline icons, and the window's initial state (normal or minimized).

2.6.14.4 Restarting Motif Window Manager

V1.0

The file SYS$MANAGER:DECW$MWM.COM is used for information on how to restart the Window Manager. By default, it is always restarted on all the screens that are available. However, if you are not starting the Window Manager from the Session Manager, then the Window Manager might not have been initially started on all the available screens. You can modify this file to change the way the window is restarted for your system.

2.6.15 Motif Window Manager—Known Problems

This section contains information about known problems with the Motif Window Manager.

2.6.15.1 Accelerators for Icon in Icon Box

V1.0

When using an icon box, the accelerators on an icon's Window menu do not work unless you display that menu. Instead, accelerators always apply to the icon box.

2.6.15.2 Application Started as Icon

V1.0

If an application is started as an icon before the Window Manager is started, the application's Main window is not minimized when the Window Manager is started. This can occur when you change X properties relating to icons before the Window Manager is started or when the Window Manager is starting. It can also occur if you start an application as an icon when the Window Manager is restarted after it has been previously started and is inactive.

2.6.15.3 Customizing Color-Related Resources for Monochrome Monitors

V1.0

The Motif Window Manager does not support full customization of color-related resources for monochrome monitors in the Options dialog box. In order to change the colors, you might need to modify the pixmap resources by directly editing the DECW$MWM_BW.DAT resource file. For example, to change the color of the active window's title background, you must change the Mwm*activeBackgroundPixmap resource. Some values include 25_foreground, 50_foreground, 75_foreground, and unspecified pixmap.

In addition, by default, the title text is created with a white background. To use the same color as the rest of the title, set the Mwm*cleanText resource to FALSE.

2.6.15.4 Customizing Colors on 4-Plane Displays

V1.1

A 4-plane display has a limited number of colors. You might not be able to modify colors with the color mix widget if the system usurps the color entries.

To correct this problem, distribute as many of the colors as possible between the screen, window, border, icons, and pointer.

2.6.15.5 Customizing Icon Placement in Icon Box

V1.0

If you customize the icon placement of applications by specifying iconX and iconY resources in the application's resource file, the placement is ignored if you are using an icon box.

2.6.15.6 DECwindows XUI Modal Dialog Boxes

V1.0

Two problems exist with DECwindows XUI modal dialog boxes.

If an application displays a dialog box before the main window is visible, the modal dialog box might be hidden behind the main window. It then appears that the application is hung. In this case, you must terminate the application.

If you dismiss a modal dialog box, the focus might not automatically revert to its parent.

2.6.15.7 Customizing the Icon Box

V1.0 To customize the position and size of an icon box, move and resize the icon box and then select Apply Current Settings from the Workspace Options menu.

2.6.15.8 Moving the Icon Box Off Screen

V1.0 If you move the icon box to the edge of the screen and then resize it using the keyboard, you can move it off the screen. To retrieve the icon box, press Alt+Tab until you reach that window and then press Shift Escape (F11) to bring up the Window menu for that window. You can then move the window back onto the screen.

2.6.15.9 Truncating the Icon Title

V1.0 The text for inactive icons is truncated to the size of the icons. Currently, there are two methods to view the complete text for an icon. You can make the icon active or you can customize your icons by choosing the Icon Options menu item and changing the width of the icons.

2.6.15.10 Invoking Motif Window Manager Help

V1.2 Invoking Motif Window Manager Help through the workspace Help menu causes the Motif Window Manager to stop.

2.6.15.11 Multihead Systems—Customizing Colors

V1.0 If you have a multihead system with different monitor types (color, monochrome, or gray-scale), you can customize the colors only by using the Options dialog box on the monitors that match the type of your main monitor (screen 0). To customize the other monitors, you must either log in to a system with that monitor type or directly edit the resource files.

2.6.15.12 Multiline Icon Title Not Centered

V1.0 The Window Manager does not center all the lines of a multiline icon title.

2.6.15.13 Truncated Icon Title in Vertical Icon Box

V1.0 If the icon box is displayed vertically with only one column, then when an icon is selected, the active icon label is truncated on the right side.

2.6.16 Notepad

This section contains information about the Notepad application.

2.6.16.1 Notepad Is Linked with the OSF/Motif Release 1.1.3 Toolkit

V1.2–3 The Notepad application is linked with the OSF/Motif Release 1.1.3 Toolkit. Notepad is not modified to link with the OSF/Motif Release 1.2.3 Toolkit, which is provided with the DECwindows Motif Version 1.2–3 for OpenVMS product. The following restrictions apply:

- OSF/Motif Release 1.2 drag-and-drop functionality is not supported. As a workaround, use the standard clipboard operations (Cut, Copy, and Paste) to transfer text into Notepad.
- OSF/Motif Release 1.2 tear-off menus are not supported.

2.6.17 Paint

This section contains information about the Paint application.

2.6.17.1 Private Colormaps

V1.1

If your workstation does not have sufficient colormap entries to view or edit a color image, Paint creates a private colormap. When this happens, the Paint image retains its colors, but the colors on the rest of the workstation are modified. To restore the colors to their original values, give another window input focus by clicking on it.

2.6.17.2 Enhancing the Performance of Some Paint Operations

V1.0

On GPX systems, Paint might appear slow even when performing basic operations such as drawing a brush stroke. This is because the pixmap is being swapped into the pixmap memory in order to paint the object. If Paint performance is slow, click on the Pencil tool and draw a point in the image area. This should improve performance following the initial Pencil click.

When editing images (especially color images), you can resize the image area using the Picture Size... entry from the Options menu. Resize to the least possible image area to significantly reduce the amount of required pixmap memory.

2.6.18 Print Screen

This section contains information about the Print Screen application.

2.6.18.1 Large Default Icon

V1.0

The Print Screen application does not have its own distinctive icon and does not support icon sizes other than large.

As a result, the icon is the default Motif icon of four squares. If you attempt to customize the icon size, only a portion of this icon is viewable.

2.6.19 Session Manager and FileView

This section contains information about the Session Manager and FileView applications.

2.6.19.1 Removing Support for LinkWorks Manager and LinkWorks

V1.2–3

Delete the verbs “LinkWorks Manager” and “LinkWorks Setup” from your Applications menu; they are no longer supported. The verbs are deleted from the system profile file, VUE$SYSTEM_PROFILE.VUE$DAT. If users modified or moved these verbs, the verbs are stored in the private profile file VUE$PROFILE$DAT.

Remove any customizations that refer to these two verbs either from autostartup or as a private verb by deleting the verbs from the Options Automatic Startup and the Options Menus dialogs.

2.6.19.2 Invoking DECchart from the Session Manager Applications Menu

V1.2–3

If you invoke the DECchart application from the Session Manager applications menu, your session can fail. As a workaround to this problem, edit the file VUE$LIBRARY:DECCHART$VUE.COM to incorporate the following text:

```
$!
$! Copyright (c) 1989, 1991 Digital Equipment Corporation.
$! All rights reserved.
$!
$! Command procedure to run DECchart from the User Executive
$! in DECwindows
$!
$  vue$suppress_output_popup
$!
$! See if we should skip the dialog box
$!
$    vue$get_symbol vue$show_hidden_dialogs
$    vue$read show_hidden_dialogs
$
$    if show_hidden_dialogs then goto select_qualifiers
$       vue$get_qualifiers
$       goto do_chart
$
$select_qualifiers:
$       vue$popup_qualifiers
$
$
$do_chart:
$
$ vue$popup_progress_box 8
$ vue$read vue$command
$
$ if "’’vue$command’" .eqs. "DETACHED_APPLY" then goto
$ select_qualifiers
$
$ ’vue$command
$
$ decchart :== $sys$system:decchart$motif.exe
$ vue$get_next_selection
$ vue$read selection
$ decchart ’selection
$
$ vue$check_verb_loop
$ vue$read loop
$ if "’’loop’" .eqs. "TRUE" then goto select_qualifiers
```

The only change is to relocate the following line from the beginning of the VUE$LIBRARY:DECCHART$VUE.COM file:

```
$ vue$popup_progress_box 8
```

This statement causes the Progress Box to pop up after the qualifiers are processed.

2.6.19.3 Eastern European Languages Now Available

V1.2

In addition to the previously available languages, you can select the following languages from the Session Manager Language Options dialog box:

- Czech
- Greek
- Hungarian
- Polish
- Russian
- Slovak

- Turkish

Note

To select these language options, the language variant kits must be installed on your system, along with the base English DECwindows Motif Version 1.2 for OpenVMS product.

2.6.19.4 Security Options

V1.2 In the Session Manager Security Options dialog box, place the node name within quotation marks if the name contains any of the following:

- Reserved characters: space, tab, comma (,) or double quotation mark (")
- Double colon (::)
- A colon (:) as the final character in the node name

Session Manager automatically adds quotation marks to the node name if they are needed, unless the node name begins with a double quotation mark. If the node name begins with a double quotation mark, Session Manager assumes that the user has already quoted the node name and does not change it.

Within a quoted string, a double quotation mark should be replaced by two double quotation marks (""). For example, the quoted string `"DEC:.zko."my node""` should be changed to the following:

```
("DEC:.zko.""my node""").
```

2.6.19.5 Using the Color Customizer with Session Manager

V1.2 If you are using the color customizer example program provided in the directory DECW$EXAMPLES to control Session Manager colors, the Session Manager color customization dialog boxes used to modify those colors may not reflect the correct current color values. This is normal behavior; use the color customizer instead of the Session Manager color customization dialog boxes to change these values. Alternatively, exit the color customizer and restart your session. See Section 3.4 for more information about the color customizer.

2.6.19.6 Input Focus Change When Starting Private Logo

V1.0 If you are logging into DECwindows and using a private logo command file, input focus might revert unexpectedly to the Username field when the private logo starts up.

2.6.19.7 Session Manager Process—Stopping

V1.0 Stopping the Session Manager process can have serious consequences for nonprivileged workstation users. DECwindows must be restarted to avoid the following problems:

- A nonprivileged user cannot start a new Session Manager or create a new login box.
- If the session is paused, the Pause cover window is deleted and unauthorized users can access windows on that workstation.
- The workstation can also become unusable if the Session Manager process is terminated by the job controller (for example, when the user’s access hours, which might be from 8 a.m. to 5 p.m., are exceeded).

As a possible solution, restart DECwindows on each workstation in a batch job that runs during off-hours.

If you stop the Session Manager process, restart DECwindows (if you have system manager privileges) with the following command:

```
$ @SYS$MANAGER:DECW$STARTUP RESTART
```

2.6.19.8 Public Profiles for Layered Products

V1.0

Layered products that provide a DECwindows interface can use the Create Public Profile File command to create a profile file that ships with the layered product and is installed when the system manager installs the layered product. The file can be placed in the SYS$COMMON:[VUE$LIBRARY.USER] directory. Product-specific profile files should be named by appending the product prefix to the name PROFILE.VUE$DAT.

For example, the file CMS$PROFILE.VUE$DAT names a profile containing CMS definitions. Any command files or other related files can also be placed in SYS$COMMON:[VUE$LIBRARY.USER] and should be referenced as VUE$LIBRARY in the profile.

For more information on using the Create Public Profile File command, refer to *Using DECwindows Motif for OpenVMS*.

2.7 Window Dump to Print File (xpr) Utility

V1.2

The Window Dump to Print File utility prints an X Window dump using the xpr program.

The xpr program receives as input a window dump file produced by the Window Dump utility (xwd) and formats it for output on the following printers:

- PostScript
- Digital LN03 or LA100
- IBM PP3812 page printer
- HP LaserJet (or other PCL printers)
- HP PaintJet

You must specify an input file. The xpr program prints the largest possible representation of the window on the output page. Options allow the user to add headers and trailers, specify margins, adjust the scale and orientation, and append multiple window dumps to a single output file.

Use the following command format:

```
$ xpr input_file [options...]
```

Options include:

```
-append filename  -noff  -output filename
-compact
-device {ln03 | la100 | ps | lw | pp | ljet | pjet | pjetxl}
-dump
-gamma correction
-gray {2 | 3 | 4}
-height inches  -width inches
-header string  -trailer string
-landscape  -portrait
-left inches  -top inches
-noposition
-nosixopt
-plane n
-psfig
-render type
-report
-rv
-scale scale
-slide
-split n-pages
```

Table 2–1 defines the available options.

Table 2–1 Window Dump to Print File Options

Option	Description
-device *devtype*	Specifies the device on which the file is printed. Currently supported devices: la100 — Digital LA100. ln03 — Digital LN03. ljet — HP LaserJet series and other monochrome PCL devices such as ThinkJet, QuietJet, RuggedWriter, HP series, and HP-series printers. pjet — HP PaintJet (color mode). pjetxl — HP PaintJet XL Color Graphics Printer (color mode). pp — IBM PP3812. ps — PostScript printer. lw — LaserWriter is equivalent to -device ps and is provided only for backwards compatibility. The default is PostScript.
-scale *scale*	Affects the size of the window on the page. The PostScript, LN03, and HP printers can translate each bit in a window pixel map into a grid of a specified size. For example, each bit might translate into a 3x3 grid. This would be specified by -scale 3. By default, a window is printed with the largest scale that will fit onto the page for the specified orientation.
-height *inches*	Specifies the maximum height of the page.
-width *inches*	Specifies the maximum width of the page.
-left *inches*	Specifies the left margin in inches. Fractions are allowed. By default the window is centered in the page.

(continued on next page)

Table 2–1 (Cont.) Window Dump to Print File Options

Option	Description
-top *inches*	Specifies the top margin for the picture in inches. Fractions are allowed.
-header *string*	Specifies a header string to be printed above the window.
-trailer *string*	Specifies a trailer string to be printed below the window.
-landscape	Forces the window to be printed in landscape mode. By default, a window is printed so that its longest side follows the long side of the paper.
-portrait	Forces the window to be printed in portrait mode. By default a window is printed so that its longest side follows the long side of the paper.
-plane *number*	Specifies which bit plane to use in an image. The default is to use the entire image and map values into black and white based on color intensities.
-gray	Uses a 2x2, 3x3, or 4x4 gray scale conversion on a color image, rather than mapping to strictly black and white. This doubles, triples, or quadruples the effective width and height of the image.
-rv	Forces the window to print in reverse video.
-compact	Uses run-length encoding for compact representation of windows with white pixels.
-output *filename*	Specifies an output file name.
-append *filename*	Specifies a file name previously produced by xpr to which the window is to be appended.
-noff	When specified in conjunction with -append, the window appears on the same page as the previous window.
-split *n-pages*	Allows the user to split a window onto several pages. This might be necessary for very large windows that would otherwise cause the printer to overload and print the page in an obscure manner.
-psfig	Suppresses translation of the PostScript picture to the center of the page.
-density *dpi*	Indicates dot-per-inch density tobe used by the HP printer.
-cutoff *level*	Changes the intensity level where colors are mapped to either black or white for monochrome output on a LaserJet printer. The level is expressed as percentage of full brightness. Fractions are allowed.
-noposition	Causes header, trailer, and image positioning command generation to be bypassed for LaserJet, PaintJet and PaintJet XL printers.
-gamma *correction*	Changes the intensity of the colors printed by the PaintJet XL printer. The correction is a floating-point value in the range 0.00 to 3.00. Consult the operator's manual to determine the correct value for the specific printer.
-render *algorithm*	Allows the PaintJet XL printer to render the image with the best quality versus performance tradeoff. Consult the operator's manual to determine the available algorithms.

(continued on next page)

Table 2–1 (Cont.) Window Dump to Print File Options

Option	Description
-slide *filename*	Allows overhead transparencies to be printed using the PaintJet and PaintJet XL printers.

The program contains the following limitations:

- Support for PostScript output currently cannot use the -append, -noff, or -split options.
- The -compact option is only supported for PostScript output. It compresses white space but not black space, so it is not useful for reverse-video windows.
- For color images, map directly to PostScript image support.

Program limitations with an LN03 printer:

- The current version of xpr can print most X Windows that are not larger than two-thirds of the screen.

  For example, the LN03 prints a large Emacs window, but fails when trying to print the entire screen.
- The LN03 has memory limitations that cause it to incorrectly print large or complex windows. The two most common errors encountered are “band too complex” and “page memory exceeded” and are described as follows:
  - “band too complex”

    A window may have a particular six pixel row that contains too many changes (from black to white to black). This causes the printer to drop part of the line and possibly drop parts of the page. The printer flashes the number “1” on its front panel when this problem occurs. A possible solution to this problem is to increase the scale of the picture or to split the picture onto two or more pages.
  - “page memory exceeded”

    This occurs if the picture contains too much black space, or if the picture contains complex half-tones, such as the background color of a display. When this problem occurs, the printer automatically splits the picture onto two or more pages. The number “5” may flash on its front panel. As a possible solution to the problem, it might be necessary to either cut and paste or to rework the application to produce a less complex picture.

Program limitations with a LA100 printer:

- The picture is always printed in portrait mode.
- The scale is ignored.
- The scale factor will be different in the horizontal and vertical directions.

Program limitations with an HP printer:

- If the -density option is not specified, 300 dots-per-inch (dpi) is assumed for the ljet device and 90-dpi for the pjet device. The LaserJet printer supports 300-, 150-, 100-, and 75-dpi. Consult the operator’s manual to determine the densities supported by other printers.
- If the -scale option is not specified, the image is expanded to fit the printable page area.

- The default printable page area is 8x10.5 inches. Other paper sizes can be accommodated using the -height and -width options.
- Note that a 1024x768 image fits the default printable area when processed at 100-dpi with scale=1; the same image can also be printed using 300-dpi with scale=3, but it requires more data to be transferred to the printer.
- The xpr program may be tailored for use with monochrome PCL printers other than the LaserJet. To print on a ThinkJet (HP 2225A) printer, invoke xpr as follows:

  ```
  xpr -density 96 -width 6.667 filename
  ```

  To print black-and-white output on a PaintJet printer, invoke xpr as follows:

  ```
  xpr -density 180 filename
  ```

- The monochrome intensity of a pixel is computed as 0.30*R + 0.59*G + 0.11*B. If the computed intensity of a pixel is less than the -cutoff *level*, it prints white. This maps light-on-dark display images to black-on-white hard copy. The default cutoff intensity is 50% of full brightness. For example, specifying -cutoff 87.5 means that a pixel will be displayed as black if the computed intensity is less than 85% of full brightness.
- A LaserJet printer must be configured with sufficient memory to print the image. To print a full page at 300-dpi, approximately 2 MB of printer memory is required.
- Color images are produced on the PaintJet printer at 90-dpi. The PaintJet is limited to 16 colors from its 330 color palette on each horizontal print line. The xpr program issues a warning message if more than 16 colors are encountered on a line. Xpr programs the PaintJet for the first 16 colors encountered on each line and uses the nearest matching programmed value for other colors on the line.
- Specifying the -rv option on the PaintJet printer causes black and white to be interchanged on the output image. No other colors are changed.
- Multiplane images must be recorded by xwd in ZPixmap format. Single-plane (monochrome) images may be in either XYPixmap or ZPixmap format.
- Some PCL printers do not recognize image positioning commands. Output for these printers is not centered on the page, and header and trailer strings may not appear where expected.
- The -gamma and -render options are supported only on the PaintJet XL printers.
- The -slide option is not supported on LaserJet printers.
- The -split option is not supported on HP printers.
- The -gray option is not supported on HP or IBM printers.

2.8 Keyboard Enhancements for Disabled Users

V1.2

Alpha

On OpenVMS Alpha systems, the AccessX extension provides features to help disabled users interact with workstations running DECwindows Motif Version 1.2 software. These features make it easier to use the keyboard and mouse. You can interact with workstations by entering commands and manipulating menus and dialog boxes. However, with AccessX features, performing these input operations is even easier.

A client application is provided to enable and customize the AccessX features. To run this application, enter the following commands:

```
$ SET DEFAULT DECW$EXAMPLES
$ RUN AccessX
```

Online help is available by selecting the Help menu option.

AccessX offers the features described in the following sections.

2.8.1 Sticky Keys

The Sticky Keys feature allows you to perform multikey operations with one hand, one finger, or a mouth stick. You can use this feature to enter uppercase letters or punctuation characters without having to hold down the Shift key while pressing the character key. This feature also makes it easier to enter control characters such as Ctrl/C.

2.8.2 Mouse Keys

The Mouse Keys feature lets you map actions that you would perform with a mouse to keys on the numeric keyboard or other keys that you specify. With this feature, you can use one finger or a mouth stick to move the cursor to different areas of the screen, manipulate menus, and select, cut, and paste text.

2.8.3 Toggle Keys

The Toggle Keys feature provides audio feedback when the Shift Lock (Caps Lock) key is pressed. This feature helps users who might have difficulty seeing the keyboard light indicator for the Shift Lock key or users who are using a keyboard that does not provide light indicators for any keyboard settings.

2.8.4 Repeat Keys

The Repeat Keys feature allows you to adjust the auto-repeat keyboard mechanism speed or to turn it off entirely. With this feature turned on, you can set your keyboard so that holding down a key for a longer than average time does not cause a repeat entry of that character.

2.8.5 Slow Keys

The Slow Keys feature makes the keys less likely to respond when brushed accidentally. With this feature turned on, the computer accepts only keystrokes that are held for a certain length of time. The computer ignores light keystrokes that are held only for a moment.

2.8.6 Bounce Keys

The Bounce Keys feature eliminates the problem of pressing a key and then accidentally pressing it again before moving to another key. You can set this feature to tell the computer not to process a second pressing of a key unless a certain length of time elapses between each pressing.

2.8.7 Time Out

The Time Out feature shuts off the AccessX features on a workstation after a specified period of time. If you are sharing a workstation and have set AccessX features, the settings are automatically turned off before the next use. To retain the AccessX settings at all times, you can turn off the Time Out feature.♦

2.9 Printing from Applications

V1.2 Applications that are linked against OSF/Motif Release 1.1.3 may end abruptly when you attempt to print on systems that do not have print queues. Any layered products that linked against the OSF/Motif Release 1.1.3 libraries and use the standard DECwindows print dialog ("print widget") are also affected.

As a possible solution, either avoid displaying the DECwindows print dialog, or define a print queue on your system. The print queue does not have to be connected to a printer to accept print jobs. Assign a name to the the print queue that indicates the print queue is not connected to a printer, for example, NULL_PRINTER.

2.10 Unsupported Translations by the Motif XmText Widget

V1.2–3 By default, the Motif XmText widget does not support the following translations for Versions 1.1, 1.2, and 1.2–3 of the DECwindows Motif product:

```
F12:            beginning-of-line()
F13:            delete-previous-word()
Ctrl e:         end-of-line()
Ctrl j:         delete-previous-word()
Ctrl h:         beginning-of-line()
Ctrl r:         redraw-display()
Ctrl u:         delete-to-start-of-line()
~Ctrl ~Meta ~Shift Alt<Key>space: self-insert()
```

These translations are preferred by OpenVMS users to ensure consistency with the DCL command line interface.

The DECwindows Motif for OpenVMS Session Manager defines translations so that they can be used by applications that are displayed to an OpenVMS display server. If you display your applications on a different display server (for example, on a PC or DEC OSF/1 workstation), these translations are not functional.

To enable these translations, add the following lines to your applications default file (for example, DECW$USER_DEFAULTS:⟨*app_name*⟩.DAT or to the file DECW$USER_DEFAULTS:XDEFAULTS.DAT):

```
*XmText.translations:   #override \n\
     <Key>F12:     beginning-of-line()\n\
     <Key>F13:     delete-previous-word()\n\
     Ctrl<Key>e:   end-of-line()\n\
     Ctrl<Key>j:   delete-previous-word()\n\
     Ctrl<Key>h:   beginning-of-line()\n\
     Ctrl<Key>r:   redraw-display()\n\
     Ctrl<Key>u:   delete-to-start-of-line()\n\
     ~Ctrl ~Meta ~Shift Alt<Key>space: self-insert()\n
```

You can copy the text for these translations from the file DECW$SYSTEM_DEFAULTS:DECW$LOGIN.DAT.

Enabling these translations does not affect the standard Motif translations, such as Alt-right for end-of-line or Ctrl-right for end-of-word.

Note

If you add lines to the file DECW$USER_DEFAULTS:XDEFAULTS.DAT file, startup performance for all applications is slightly degraded.

3
System Manager Release Notes

This chapter contains system manager release notes.

3.1 Notes Specific to DECwindows Motif Version 1.2–3 for OpenVMS Software

V1.2–3

The release notes in this chapter are cumulative from DECwindows Motif Version 1.0 and still apply to DECwindows Motif Version 1.2–3 for OpenVMS software. The following sections contain system management release notes that pertain specifically to the DECwindows Motif Version 1.2–3 for OpenVMS software:

- Section 3.2, Required Reboot on OpenVMS Alpha Version 6.2 Systems
- Section 3.10, Implications of the Message, System Menu Bar: Pseudo Mouse Not Available

3.2 Required Reboot on OpenVMS Alpha Version 6.2 Systems

V1.2–3

Alpha

Although the installation dialog indicates that a system reboot is necessary to complete the installation, it may not be needed on OpenVMS VAX systems (including Version 6.2) and on OpenVMS Alpha systems prior to Version 6.2 if your system parameters are correct.

Note

You must reboot OpenVMS Alpha Version 6.2 systems after the installation of DECwindows Motif Version 1.2–3 for OpenVMS software.♦

3.3 DECwindows Motif Login Screen—Known Color Problem

V1.2

A problem may occur on systems that have a customized DECW$LOGIN.DAT file. The Start Session dialog box is the color blue instead of tan. If this condition exists, look for a customized DECW$LOGIN.DAT file in the directory SYS$COMMON:[DECW$DEFAULTS.USER] and move it to SYS$MANAGER. A DECW$LOGIN.DAT file in SYS$COMMON:[DECW$DEFAULTS.USER] prevents the "*background:" resource from being defined; thus, it will default to the color blue.

Digital provides a copy of the DECW$LOGIN.DAT file in the SYS$COMMON:[DECW$DEFAULTS.SYSTEM] directory. Any customized versions of this file should reside only in SYS$MANAGER.

3.4 Color Customizer Example Program

V1.2 The color customizer example program allows you to dynamically control the colors of your workstation environment. Window, icon, and window manager colors can be changed individually or as part of a palette switch. You can control mapping between resources and color cells, as well as the size and contents of the palette set. Also, automatic shadowing with the standard Motif shadowing algorithms is supported.

3.4.1 Supported Displays

V1.2 The color customizer supports any display using pseudocolor or grayscale visuals. This includes most 4- and 8-plane workstation displays.

3.4.2 Supported Applications

V1.2 The color customizer can affect the colors of any applications that use the current release of the Digital X Toolkit Library. Applications from other vendors and previous versions of the Digital X Toolkit Library are unaffected.

Note

If the color customizer is used to control the colors of applications that have their own color customization dialog boxes (like the Session Manager, Window Manager, and DECwindows Mail), those application-specific color customization dialog boxes may not reflect the correct current color values while the customizer is running the application. This is normal; use the customizer instead of the application-specific dialog box to change these color values.

3.4.3 Building the Color Customizer on OpenVMS Systems

V1.2 To build the color customizer on OpenVMS systems, perform the following steps:

1. Copy the files to a private directory. For example:

```
$ SET DEFAULT SYS$LOGIN
$ CREATE/DIRECTORY [.CUSTOMIZER]
$ SET DEFAULT [.CUSTOMIZER]
$ DECW$EXAMPLES:CUSTOM.C []
$ COPY DECW$EXAMPLES:CUSTOM.UIL []
$ COPY DECW$EXAMPLES:CUSTOMIMAGE.DAT []
$ COPY DECW$EXAMPLES:XSETROOT_CUST.C []
$ COPY DECW$EXAMPLES:BUILD_CUSTOMIZER.COM []
```

2. Build the customizer using the following command:

```
$ @BUILD_CUSTOMIZER.COM
```

This command procedure creates the following output files:

CUSTOM.UID
CUSTOM.EXE
XSETROOT_CUST.EXE

3.4.4 Running the Color Customizer

V1.2

To run the color customizer, perform the following steps:

1. Copy the files CUSTOM.UID and CUSTOM.EXE, which were created during the customizer build, to the directory where the customizer will be run. A typical location is the directory SYS$LOGIN or the directory DECW$USER_DEFAULTS.
2. Copy the files CUSTOM.DAT and DXMDEFAULTS.DAT from the directory DECW$EXAMPLES to the same location as you copied the files in step 1. The same typical locations apply.
3. Run the executable file CUSTOM.EXE as follows:

```
$ RUN CUSTOM
```

Note

Only the colors of applications invoked after the customizer starts will be affected. For this reason, start the customizer as the first X application during the login process.

3.4.5 Modifying the DECW$LOGIN.COM File

V1.2

As noted in Section 3.4.4, the color customizer should be the first X application started during the login process. Do this by starting it as a subprocess from within the DECW$LOGIN.COM file. Add a command to wait approximately 10 seconds between customizer startup and the startup of other applications.

For example, add the following lines to the DECW$LOGIN.COM file:

```
$! Starting the color customizer
$ DISPLAY = F$LOGICAL("DECW$DISPLAY")
$ SPAWN/NOWAIT/OUTPUT='DISPLAY' RUN SYS$LOGIN:CUSTOM.EXE
$ WAIT 0:0:10
```

See *Using DECwindows Motif for OpenVMS* and *Managing DECwindows Motif for OpenVMS Systems* for more information on the file DECW$LOGIN.COM.

3.4.6 Command Interface Summary

V1.2

A box containing a list of available palettes is in the leftmost section of the Color Customizer window. Click on the desired palette to see the colors take affect.

Below the palettes are two arrays of colored buttons, representing the dynamically allocated color cells for normal and shadow colors. To find out what resources are affected by a color cell, click and hold the arrow button next to the color cell.

Hint

As a shortcut, you can click on the screen facsimile in the rightmost corner of the dialog box. If the portion you click on is colored by one of the resource values controlled by the customizer, the pop-up window for the appropriate color button is displayed.

To modify a single color cell, click on the corresponding color button. A colormix widget pops up; as you modify the color, these modifications are reflected in your workstation environment. Use the colormix widget reset button to return to the starting color at any time. You can also change the color cell you are modifying by clicking on a different color button while the colormix widget is displayed.

The automatic shadowing option causes shadow and select colors to be automatically updated when their corresponding background colors are changed. The standard Motif shadowing algorithms are used for these calculations.

Use the File menu to modify, add, and delete color palettes as follows:

- To modify an existing palette, select the palette, change the colors, and choose Save Palette from the File menu.
- To add a new palette, select an existing palette, modify the colors as necessary, and choose Save Palette As... from the File menu. A message box prompts you for the name of the new palette.
- To delete a palette, select the palette and choose Delete Palette from the File menu.

Changes made through the File menu automatically update the CUSTOM.DAT file, which contains the resource defaults.

The File menu Exit button causes the customizer application to exit. A warning dialog is displayed first. Note that the color cells allocated by the customizer (and used by the currently running applications) will be deallocated. After the customizer exits, if the colors of the currently running applications are not correct, the applications should be restarted to restore normal colors. Usually, there is no need to exit the color customizer; it is typically kept running at all times, like the Session Manager.

3.4.7 Changing the Mapping Between Color Resources and Color Cells

V1.2

The file DXMDEFAULTS.DAT allows you to control how many dynamic color cells are allocated and what resources are affected. This file contains resource specifications like the following:

```
*background:       DXmDynamicWindowBackground
*foreground:       DXmDynamicWindowForeground
*topShadowColor:   DXmDynamicWindowTopShadow
```

When the customizer is started, the file DXMDEFAULTS.DAT is written to a property on the root window. Any application that is subsequently run and that uses the correct X Toolkit Library merges these resources with its normal resource database. Resource specifications in this file take precedence over specifications with equivalent resource names in other resource default files.

The resource values within the file DXMDEFAULTS.DAT have a special format. For each unique color value in this file that begins with the string "DXmDynamic", a color button is created in the color customizer. If the string "Shadow" is encountered in a name, the color button is placed in the shadow button box rather than the normal color button box. If a color value string ends with the suffix "Background", it is linked to any color buttons with identical prefixes and suffixes of "TopShadow", "BottomShadow", or "SelectColor" for purposes of automatic shadowing. If a color value named "DXmDynamicScreenBackground" is encountered, the color cell allocated is used by the customizer to set the root window background color.

You can edit the file DXMDEFAULTS.DAT and define resources to use the same color cells. You can have separate dynamic color cells, for scrollbar widgets or for your DECwindows Mail application, for example, by adding the following lines to the file DXMDEFAULTS.DAT:

```
Mail*background:         DXmDynamicMyMailBackground
Mail*foreground:         DXmDynamicMyMailForeground
Mail*topShadowColor:     DXmDynamicMyMailTopShadow
Mail*bottomShadowColor:  DXmDynamicMyMailBottomShadow
```

Adding the previous lines to the file DXMDEFAULTS.DAT and restarting the customizer causes four new color cells to be allocated and four new color buttons to be added to the customizer interface. These buttons are assigned default color values (usually black or white) for each palette. These defaults can then be modified for each palette through the customizer interface.

Note

The text of the DXMDEFAULTS.DAT file is read and parsed by the color customizer. The parsing algorithm does not allow comments, incorrect spacing, or incorrect resource specifications. If this file or the CUSTOM.DAT resource file become corrupt, the customizer cannot start correctly. To resolve the problem, copy the versions of CUSTOM.DAT and DXMDEFAULTS.DAT from the DECW$EXAMPLES directory into your login directory.

3.4.8 DECterm Windows Not Affected

V1.2

The color customizer does not affect the colors of DECterm windows. To change the colors of DECterm windows, copy the DECterm resource specifications from the file DXMDEFAULTS.DAT and add them to the DECterm resource defaults file DECW$USER_DEFAULTS:DECW$TERMINAL_DEFAULT.DAT. For example, add the following lines to the DECterm resource defaults file:

```
   .
   .
   .
 DECW$TERMINAL.main.terminal.background: DXmDynamicTerminalBackground
 DECW$TERMINAL.main.terminal.foreground: DXmDynamicTerminalForeground
```

This allows the DECterm window colors to be customized with the color customizer.

3.4.9 Changing the Default Value of the Automatic Shadowing Toggle Button

V1.2

The default value of the automatic shadowing toggle button is set using the Custom.autoShadow resource in the CUSTOM.DAT file as follows:

```
Custom.autoShadowing: False
```

The default value is True.

3.4.10 Using the Customizer on Multihead Systems

V1.2

The color customizer affects only applications started on the same screen as the customizer. On multihead systems, you can start a different color customizer for each screen and have a different palette in effect on each screen.

The color customizer can be configured so that it is invoked once and affects all applications regardless of where they are started. This mode is invoked by modifying the Custom.multiScreen resource in the CUSTOM.DAT file as follows:

```
Custom.multiScreen: True
```

The default value is False.

3.4.11 Using the XSETROOT_CUST.EXE Demonstration Program

V1.2

The XSETROOT_CUST.EXE demonstration program, created during the customizer build, is a modified version of the MIT utility program xsetroot that is used to set a bitmap on the root window. The XSETROOT_CUST.EXE program uses DXmDynamicScreenBackground and DXmDynamicScreenForeground as the background and foreground colors of the specified bitmap. If your DXMDEFAULTS.DAT file contains entries for these two dynamic colors, then use the customizer to dynamically modify the colors of your bitmap.

For example:

```
$ XSETROOT_CUST :== "$SYS$LOGIN:XSETROOT_CUST.EXE"
$ XSETROOT_CUST -BITMAP your_xbm_file.XBM
```

3.5 Define DECW$UTILS Global Symbol When Moving DECW$EXAMPLES Global Symbol

V1.2

By default, DECW$UTILS points to a subdirectory of DECW$EXAMPLES. If you define a DECW$EXAMPLES global symbol in the DECW$PRIVATE_APPS_SETUP.COM command procedure to change the directory for DECwindows example programs, you must also define DECW$UTILS to change the directory for utilities.

For example, to redefine both DECW$EXAMPLES and DECW$UTILS, add the following lines to the SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM procedure:

```
$ DECW$EXAMPLES == "SYS$SYSROOT:[DECWEXAMPLES]
$ DECW$UTILS == "SYS$SYSROOT:[DECWEXAMPLES.UTILS]
```

Note

If the SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM file does not exist, create it from the SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.TEMPLATE file.

Then, restart DECwindows Motif with the following command:

```
$@SYS$MANAGER:DECW$STARTUP RESTART
```

The DECW$UTILS global symbol is new in DECwindows Motif Version 1.2 for OpenVMS.

3.6 FileView Creates Detached Processes by Default

V1.1 Applications created by FileView and Session Manager are detached processes.

The implication is that during application startup, SYS$MANAGER:SYLOGIN.COM and SYS$LOGIN:LOGIN.COM command procedures are executed. Any command executed by these command procedures which reads from SYS$INPUT reads data intended to be used by FileView or Session Manager for the application startup. This prevents the application from starting. Examples of such commands are INQUIRE, READ/PROMPT, and SET TERMINAL/INQUIRE.

Extensive SYLOGIN.COM or LOGIN.COM command procedures slow down application startup. Many of the operations performed in a SYLOGIN.COM or LOGIN.COM are meaningless for DECwindows application startup. Therefore, the SYLOGIN.COM and LOGIN.COM files should be conditionalized for DECwindows application startup performance. When starting a DECwindows application, only a minimum of SYLOGIN.COM and LOGIN.COM commands should be executed. Typically, the commands that should be executed are the redefinition of DECW$USER_DEFAULTS (if present), and other logical name definitions if the user will be referencing them from within the context of a DECwindows application. The following code segment can be inserted into SYLOGIN.COM and LOGIN.COM immediately following the commands necessary for DECwindows:

```
$ mode = f$mode()
$ tt_devname = f$trnlnm("TT")
$ session_mgr_login = (mode .eqs. "INTERACTIVE") .and.  -
     (f$locate("WSA",tt_devname) .ne. f$len(tt_devname))
$ session_detached_process = (mode .eqs. "INTERACTIVE") .and. -
     (f$locate("MBA",tt_devname) .ne. f$len(tt_devname))
$ if session_mgr_login .or. session_detached_process then exit
```

Applications continue to run even if these lines are not added to the SYLOGIN.COM and LOGIN.COM files.

3.7 Customizing the Login Screen

V1.2 To customize the login screen, create a file named DECW$LOGIN.DAT in the SYS$MANAGER directory that contains your resource definitions. The custom resource definitions from SYS$MANAGER:DECW$LOGIN.DAT are merged with the resource definitions supplied by Digital in SYS$COMMON:[DECW$DEFAULTS.SYSTEM]DECW$LOGIN.DAT to form the new login screen.

Keep customized versions of the DECW$LOGIN.DAT resource file in the SYS$MANAGER directory, and **not** in DECW$SYSTEM_DEFAULTS, to prevent your customized file from being overwritten when upgraded to a newer version of DECwindows Motif for OpenVMS software. In addition, storing the file in the SYS$MANAGER directory prevents the custom file from superseding the file that is supplied by Digital.

3.7.1 Customizing the Digital Logo and Login Screen Colors

V1.2 You can define the resources in Table 3–1 to control the position and colors of the Digital logo and the color of the screen background in the Start Session screen.

Table 3–1 Moving the Digital Logo and Changing Login Screen Colors

Resource	Description
rootColor	Color of the screen background.
logoColor	Color of the Digital logo (default is burgundy).
logoX	x position of the Digital logo (default is 0).
logoY	y position of the Digital logo (default is 75).
centerLogoX	Boolean; if true (default), the Digital logo is centered horizontally on the screen.

For example, to position the Digital logo at x=100, y=600, add the following resource definitions to the SYS$MANAGER:DECW$LOGIN.DAT file:

```
decw$login.logoX: 100
decw$login.logoY: 600
decw$login.centerLogoX: false
```

3.7.2 Changing Positions of the Start Session and Set Password Dialog Boxes

V1.2 You can define the resources in Table 3–2 to control the position of the Start Session and Set Password dialog boxes.

Table 3–2 Changing Position of the Start Session and Set Password Dialog Boxes

Resource	Description
centerStartSessionX	Boolean; if true (default), the Start Session dialog box is centered horizontally.
centerStartSessionY	Boolean; if true (default), the Start Session dialog box is centered vertically.
centerSetPasswordX	Boolean; if true (default), the Set Password dialog box for expired passwords is centered horizontally.
centerSetPasswordY	Boolean; if true (default), the Set Password dialog box is centered vertically.

For example, to position the Start Session dialog box at x=100, y=600, add the following resource definitions to the SYS$MANAGER:DECW$LOGIN.DAT file:

```
decw$login.centerStartSessionX: false
decw$login.centerStartSessionY: false
decw$login.HiddenShell.x: 100
decw$login.HiddenShell.y: 600
```

To position the Set Password dialog box at x=30, y=100, add the following resource definitions to the SYS$MANAGER:DECW$LOGIN.DAT file:

```
decw$login.centerSetPasswordX: false
decw$login.centerSetPasswordY: false
decw$login.SetPasswordShell.x: 30
decw$login.SetPasswordShell.y: 100
```

3.7.3 Disabling a Node Name Display in the Start Session Dialog Box

V1.2 To prevent a node name from being displayed in the Start Session dialog box, add the following resource definition to the SYS$MANAGER:DECW$LOGIN.DAT file:

```
decw$login.displayNodeName: false
```

3.8 Customized Login Logos

V1.1 By default, if there is no DECwindows Motif license registered for the SYSTEM account, DECwindows does not display customized login logos. This is a problem on systems with DECwindows Motif personal-use licenses that do not include SYSTEM on the list of authorized DECwindows users.

To display a customized logo without a DECwindows Motif license for SYSTEM, add the following definition to the SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM file:

```
$ DECW$LOGINLOGOSUB == "TRUE"
```

Note

If the file does not exist, copy it from the file SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.TEMPLATE.

After editing the setup file, restart DECwindows Motif using the following command:

```
$ @SYS$MANAGER:DECW$STARTUP RESTART
```

DECwindows Motif login starts the logo process as a subprocess instead of as a detached process. The license check sees that the logo process is a child of the login process and that the X connection is opened.

3.9 Version Checking Command Files

V1.0 The DECwindows Motif for OpenVMS kit contains version-checking command procedures that layered products can use during their installation procedure. The following three files are placed in the SYS$UPDATE directory during the installation of VMS DECwindows Motif Version 1.0:

- DECW$GET_IMAGE_VERSION.COM

  A command procedure that extracts the image identification string from an image and places it into a user-defined symbol.

- DECW$COMPARE_VERSIONS.COM

  A command procedure that compares two image identification strings and assigns a value to a user-defined symbol with these possible results:

  - Facility codes do not match.
  - Identifiers are the same.
  - Second identifier is older than the first.
  - Second identifier is newer then the first.

- DECW$VERSIONS.COM

A command procedure used to display the versions of several components of the DECwindows Motif layered product and the X11 display server. The DECW$VERSIONS.COM procedure uses the DECW$GET_IMAGE_VERSION.COM command procedure to obtain the image idents of each file. Use the following command to display the versions on sys$output:

```
$ @SYS$UPDATE:DECW$VERSIONS *
```

Component	Description
DECwindows ident	MIT Xlib shareable image
DECwindows server	Server DIX file
DECwindows transport	Transport common
DECwindows Xlib	MIT Xlib shareable image
DECwindows OSF/Motif Toolkit	OSF/Motif Xm Toolkit
DECwindows applications	DECwindows FileView
DECwindows programming	OSF/Motif UIL compiler

The output from the command procedure shows DW, the version number, and the date the image is created.

For example:

```
     DW  V1.23-940924, is version 1.2-3. and was created on September 9, 1994
```

3.10 Implications of the Message, “System Menu Bar: Pseudo Mouse Not Available”

V1.2–3

“System Menu Bar: Pseudo Mouse not available” is an informational message that is included in the user’s DECW$USER_DEFAULTS:DECW$SM.LOG file when you run a session. It is not an error message. The message occurs when the OpenVMS Session Manager is run remotely to a non-OpenVMS server. The OpenVMS server provides pseudomouse mode, a mode that allows you to use arrow keys to move the mouse cursor.

3.11 DECterm System Management Issues

This section contains information about DECterm system management issues.

3.11.1 DECterm Logical Names

V1.1

Table 3–3 describes the logical names that are supported by DECterm. If you do not define these logical names in your LOGIN.COM file, the controller uses the default files.

Table 3–3 Logical Names Supported by DECterm

Logical Name	Description
DECTERM_DIAG	Enables diagnostic messages.
DECTERM_SHOW_PARSING	Shows characters as they are parsed.
DECTERM_CHECK_MEMORY	Enables strict memory checking.

(continued on next page)

Table 3–3 (Cont.) Logical Names Supported by DECterm

Logical Name	Description
FAKE_VM_REAL_FREE_OFF	Must be set to 1 if DECTERM_CHECK_MEMORY is defined.
DECW$DECTERM_ERROR	Name of error log file; default is DECTERM_ERROR.LOG.
DECW$DECTERM_OUTPUT	Name of diagnostic output file; default is SYS$OUTPUT.
DECW$DECTERM_REGIS_CURSOR	Specifies which cursor to use for ReGIS.
DECW$TERMINAL_NODENAME	Node name used by controller if it cannot find another name.
DECW$DECTERM_CTRL_SSRWAIT	Sets the SSRWAIT flag for the controller.
DECW$DECTERM_CTRL_PSWAPM	Sets the PSWAPM quota for the controller.
DECW$DECTERM_CTRL_WSEXTENT	Sets the WSEXTENT quota for the controller.
DECW$DECTERM_CTRL_WSQUOTA	Sets the WSQUOTA quota for the controller.
DECW$DECTERM_DISABLE_QUOTA_CHECKING	Turns off quota checking.
DECW$DECTERM_MEM_DIAG	Shows controller quota calculations.

3.11.2 Automatic Window Positioning

V1.1

A resource has been defined to manage repositioning a DECterm window when a resize operation forces part of the window off the screen. If a DECterm window is enlarged by using the Options/Window dialog box or by entering a SET TERMINAL/PAGE=*nn* or SET TERMINAL/WIDTH=*nn* command, the controller moves the newly resized DECterm window so that it can be viewed in its entirety. If you prefer DECterm not to move, add the following line to your DECW$TERMINAL_DEFAULT.DAT file:

```
DECW$TERMINAL.main.terminal.autoAdjustPosition: off
```

3.11.3 Hold Screen Response Time

V1.1

If the hold screen key response time is too slow, add the following lines to your DECW$TERMINAL_DEFAULT.DAT file:

```
DECW$TERMINAL.main.terminal.syncFrequency: 1
DECW$TERMINAL.main.terminal.batchScrollCount: 1
```

Using this resource can affect the performance of the DECterm window. The actual impact on performance varies from site to site. You can trade off scrolling speed to hold-screen response time. A faster hold-screen response results in a slower scrolling speed. The default values for these resources are 10 and 0, respectively.

3.11.4 Using the Debugger

V1.0

To redirect the output from the debugger to a DECterm window, enter the following command:

```
$ CREATE/TERMINAL/NOPROCESS/DEFINE=xxx
```

This command creates a DECterm without an associated process but with a logical name of *xxx* that points to the terminal. This procedure enables you to direct output to a DECterm window other than the window where the application is currently running. To redirect the output, enter the following commands:

```
$ DEFINE /USER DBG$INPUT xxx:
$ DEFINE /USER DBG$OUTPUT xxx:
$ RUN /DEBUG application.EXE
```

3.11.5 Virtual Terminal Support

V1.0

To create a process that uses a virtual terminal, enter the following command:

```
$ CREATE/TERMINAL/NOPROCESS
```

Then establish focus to the newly created DECterm, press Return, and log in. Note that the /DEFINE qualifier is not required.

4

Programmer Release Notes

This chapter contains programmer release notes.

4.1 Programming Release Notes Specific to DECwindows Motif Version 1.2–3 for OpenVMS Software

The release notes in this chapter are cumulative from DECwindows Motif Version 1.0 and still apply to the DECwindows Motif Version 1.2–3 for OpenVMS software. The following sections contain programmer release notes that pertain specifically to the DECwindows Motif Version 1.2–3 for OpenVMS release:

- Section 4.2, DECwindows Motif for OpenVMS Toolkit Versions
- Section 4.3, Run-Time and Programming Environments
- Section 4.4, OSF/Motif Toolkit Compatibility
- Section 4.5, Language Bindings
- Section 4.6, OSF/Motif Release 1.1.3 Programming Support and XUI
- Section 4.8.1, Running Translated Images on OpenVMS Systems
- Section 4.12.2.1, Version Number
- Section 4.13.4, ReGIS Input Cursors
- Section 4.14.4, Compile-Time Incompatibilities in Motif Header Files

4.2 DECwindows Motif for OpenVMS Toolkit Versions

V1.2–3 The DECwindows Motif Version 1.2–3 for OpenVMS release is based on the OSF/Motif Release 1.2.3 Toolkit and MIT X11 Release 5 (R5).

V1.2 The DECwindows Motif Version 1.2 for OpenVMS release is based on the OSF/Motif Release 1.2.2 Toolkit and MIT X11 Release 5 (R5).

V1.1 The VMS DECwindows Motif Version 1.1 release was based on the OSF/Motif Release 1.1.3 Toolkit and MIT X11 Release 4 (R4).

V1.0 The VMS DECwindows Motif Version 1.0 release was based on the OSF/Motif Release 1.1.1 Toolkit and MIT X11 Release 4 (R4).

4.3 Run-Time and Programming Environments

V1.2–3 The following run-time and programming environments are provided with the DECwindows Motif Version 1.2–3 for OpenVMS software:

- Run-time support is provided for the OSF/Motif Release 1.2.3 Toolkit, OSF/Motif Release 1.1.3 Toolkit, and the X User Interface (XUI) Toolkits.
- Development support is provided for the OSF/Motif Release 1.2.3 Toolkit only.

You can, however, choose during installation to save the OSF/Motif Release 1.1.3 programming files that existed on your system prior to the installation of the DECwindows Motif Version 1.2–3 for OpenVMS product. Refer to the *DECwindows Motif Version 1.2–3 for OpenVMS Installation Guide* for details about saving the OSF/Motif Release 1.1.3 programming environment.

- If you install the software using the VMSINSTAL procedure and choose to save these programming files, the files are moved to subdirectories and can be accessed for programming.
- If you install the software using the POLYCENTER Software Installation utility and if programming support for the OSF/Motif Release 1.1.3 Toolkit is present, then you can choose to save the header files and UIL compiler that were used to develop OSF/Motif Release 1.1.3 applications. If selected, the preinstallation procedure creates a subdirectory called [.DECW$113], and the previous programming files are moved into the new subdirectory.

See the *DECwindows Motif Version 1.2–3 for OpenVMS Installation Guide* for additional information about saving the Release 1.1.3 programming environment.

- Application development with the DECwindows Motif Version 1.2–3 for OpenVMS product is supported for DEC C++ as well as for updated Motif language bindings for Ada, Pascal, Fortran, and C. Language bindings for Ada are available in the DEC Ada Version 3.0A for OpenVMS (VAX and Alpha) layered products.

4.4 OSF/Motif Toolkit Compatibility

V1.2–3 You cannot code and compile an application with DECwindows Motif Versions 1.2 or 1.2-3 include files, transport the resulting object files to a system running DECwindows Motif Version 1.1 software, link the object files against DECwindows Motif Version 1.1, and expect the executable image (.EXE) to run. Differences in structures defined in the .h files cause the offsets to be misaligned and thereby prevent you from creating a meaningful executable.

V1.2 The OSF/Motif Release 1.1.3 and Release 1.2.2 Toolkits, on which the DECwindows Motif Versions 1.1 and 1.2 layered products are based, are not compatible. However, applications built against the OSF/Motif Release 1.1.3 Toolkit will continue to run without modification with DECwindows Motif Version 1.2 for OpenVMS; the OSF/Motif Release 1.1.3 Toolkit shareable libraries will continue to be supplied with the DECwindows Motif Version 1.2 for OpenVMS layered product and will be available to existing DECwindows applications on the system.

Although DECwindows Motif Version 1.1 applications will continue to run without modification with DECwindows Motif Version 1.2 for OpenVMS, it is important to note that only those applications that have been built against the OSF/Motif Release 1.2.2 Toolkit can take advantage of new Version 1.2 features, such as drag-and-drop functionality and tear-off menus.

In most cases, simple applications (for example, applications that do not contain private toolkit routines or user-written widgets) can be recompiled and relinked against the OSF/Motif Release 1.2.2 Toolkit to include the new Version 1.2 features. However, previously unidentified problems in a DECwindows Motif Version 1.1 application might be exposed by relinking against the new OSF/Motif

Release 1.2.2 Toolkit, and behavior and widget layout changes in the Release 1.2.2 Toolkit might require other modifications to your application as well.

More complex applications will probably require more substantial modifications with the new DECwindows Motif Version 1.2 for OpenVMS Toolkit to use the macros and functions provided.

If the code generates errors because of references to display structure fields, define the constant XLIB_ILLEGAL_ACCESS for X11 R5 compatible structure access.

For example:

```
$ CC/DEFINE=XLIB_ILLEGAL_ACCESS DECBURGER.C
```

4.4.1 _Xm Routines

V1.2

The OSF/Motif Toolkit libraries contain many undocumented routines, which are prefixed with _Xm. These routines are intended to be used only by the standard Motif widgets. OSF reserves the right to modify the API or functionality of these routines, or to delete them altogether in future releases.

Caution

Digital supplies access to the _Xm routines by copying them into the shareable image transfer vector for the OSF/Motif Release 1.2.2 Toolkit. Digital does not document or support these routines or guarantee their continued existence in future releases. Application developers who use these routines do so at their own risk.

4.4.2 OSF/Motif Release 1.2.2 and X11 Release 5 Shareable Libraries

V1.2

Because the OSF/Motif Release 1.2.2 Toolkit and the OSF/Motif Release 1.1.3 Toolkit are not binary compatible, applications must link with one toolkit or the other. Applications based on OSF/Motif Release 1.2.2 can only link against OSF/Motif 1.2.2-based and X11 R5-based shareable libraries. Applications based on OSF/Motif Release 1.1.3 can only link against OSF/Motif 1.1.3-based and X11 R4-based shareable libraries.

To provide both OSF/Motif Release 1.2.2-based and OSF/Motif Release 1.1.3-based shareable libraries, the Release 1.1.3-based libraries have the same file names as in DECwindows Motif Version 1.1, and the Release 1.2.2-based libraries contain a suffix of either "R5" or "12".

Shareable libraries that work with either the OSF/Motif Release 1.1.3 Toolkit or the OSF/Motif Release 1.2.2 Toolkit do not have a suffix. These libraries are as follows:

- CDA$ACCESS.EXE
- DECW$D2DXLIBSHR.EXE
- DECW$XEXTLIBSHR.EXE
- DECW$XLIBSHR.EXE
- XDPS$DPSBINDINGSSSHR.EXE
- XDPS$DPSCLIENTSHR.EXE
- XDPS$DPSLIBSHR.EXE

- XIE$SHRLIB.EXE

Shareable libraries that are linked with Release 5 (R5) of the Xt Toolkit have a suffix of "R5". Libraries based on the XUI Toolkit have no R5 equivalent libraries and should not be included in a linker options file based on X11 R5 or OSF/Motif Release 1.2.2. Table 4–1 lists these file names.

Note

The DECW$XLIBSHR.EXE file is the MIT Release 5 version, not the MIT Release 4 version of Xlib.

Table 4–1 Names of Shareable Libraries Based on R5

Names of Files Based on R4	Names of Files Based on R5
DECW$MAILSHR.EXE	DECW$MAILSHRR5.EXE
DECW$DWTLIBSHR.EXE	(None)
DECW$DWTSHR.EXE	(None)
DECW$XMULIBSHR.EXE	DECW$XMULIBSHRR5.EXE
DECW$XTRAPLIBSHR.EXE	DECW$XTRAPLIBSHRR5.EXE
DECW$XTSHR.EXE	DECW$XTLIBSHRR5.EXE

Release 5 does not provide an equivalent file for DECW$DWTLIBSHR.EXE or DECW$DWTSHR.EXE. Applications that are built for Release 5 cannot link against these files.

Shareable libraries that are linked with OSF/Motif Release 1.2.2 have a suffix of "12". They should be linked only with libraries compatible with R5 and OSF/Motif Release 1.2.2. Table 4–2 lists these file names.

Table 4–2 Names of Shareable Libraries Based on OSF/Motif Release 1.2.2

Names of Files Based on Release 1.1.3	Names of Files Based on Release 1.2.2
DDIF$VIEWSHR.EXE	DDIF$VIEWSHR12.EXE
DECW$BKRSHR.EXE	DECW$BKRSHR12.EXE
DECW$DXMLIBSHR.EXE	DECW$DXMLIBSHR12.EXE
DECW$MAILSHR12.EXE	(None)
(None)	DECW$MRMLIBSHR12.EXE
DECW$PRINTWGTSHR.EXE	(None)
DECW$TERMINALSHR.EXE	DECW$TERMINALSHR12.EXE
DECW$XMLIBSHR.EXE	DECW$XMLIBSHR12.EXE
DGIT$LIBSHR.EXE	DGIT$LIBSHR12.EXE
IMG$SHRLIB.EXE	IMG$SHRLIB12.EXE
LWK$DXMSHR.EXE	LWK$DXMSHR12.EXE
XNL$SHR.EXE	XNL$SHR12.EXE

There is no DECW$PRINTWGTSHR12.EXE file; the Print Widget is part of the DECW$DXMLIBSHR12.EXE file.

The DECW$MRMLIBSHR12.EXE file is a new image that includes Motif Resource Manager (Mrm) routines that were formerly part of the DECW$XMLIBSHR.EXE file. Any program based on OSF/Motif Release 1.2.2 that calls Mrm routines to access .UID files should link with this library.

For example, a typical linker options file for a program based on OSF/Motif Release 1.1.3 might be as follows:

```
SYS$SHARE:DECW$XLIBSHR/SHARE
SYS$SHARE:DECW$XTSHR/SHARE
SYS$SHARE:DECW$DWTLIBSHR/SHARE
SYS$SHARE:DECW$XMLIBSHR/SHARE
SYS$SHARE:DECW$DXMLIBSHR/SHARE
```

To link this program with OSF/Motif Release 1.2.2, the linker options file should be changed to:

```
SYS$SHARE:DECW$XLIBSHR/SHARE
SYS$SHARE:DECW$XTLIBSHRR5/SHARE
SYS$SHARE:DECW$XMLIBSHR12/SHARE
SYS$SHARE:DECW$MRMLIBSHR12/SHARE
SYS$SHARE:DECW$DXMLIBSHR12/SHARE
```

These changes eliminate the reference to the XUI Toolkit (DECW$DWTLIBSHR.EXE) and links with the Motif Resource Manager (DECW$MRMLIBSHR12.EXE).

4.5 Language Bindings

V1.2–3 Fortran, Pascal, C, and C++ language bindings are provided to support software development for the following DECwindows components:

- Xlib (Release 5)
- Xt (Release 5)
- Xm (Release 1.2.3)
- Mrm (Release 1.2.3)
- DXm

4.5.1 OSF/Motif Release 1.1.3 Ada Bindings

V1.2 This release of DECwindows Motif does not include Ada bindings for either the OSF/Motif Release 1.1.3 Toolkit or the OSF/Motif Release 1.2.2 Toolkit. Strongly typed bindings for both toolkits are part of the DEC Ada Version 3.0A language product.

If DECwindows Motif is installed on OpenVMS DECwindows Motif Version 1.1, then the existing OSF/Motif Release 1.1.3 Ada bindings, located in the DECW$INCLUDE directory, are not deleted or moved.

4.6 OSF/Motif Release 1.1.3 Programming Support and XUI

V1.2–3 **Restrictions**

- Problem:

  You will encounter problems if you save the previous (Release 1.1.3) programming environment and attempt to run an OSF/Motif Release 1.1.3 (Xlib Release 4) program that is created on a DECwindows Motif Version 1.2 or Version 1.2–3 system by running the executable on a DECwindows Motif Version 1.1 target system. That is, if you build an application on a

DECwindows Motif Version 1.2 or Version 1.2–3 for OpenVMS system, the image is linked with Xlib Release 5, by default. If you attempt to run that executable on a system where OSF/Motif Release 1.1.3 is installed, the result is an ident mismatch fatal error.

In addition, the OSF/Motif Release 1.1.3 and the Release 1.2.2 files produce different output; they have different structures and are therefore not compatible. Use the appropriate UIL compiler to produce the correct UID file.

- Resolution:

  1. Rename the DECwindows Motif Version Version 1.1 executable compiler, DECW$MOTIF.EXE, to DECW$MOTIF113.EXE and keep the file in the SYS$COMMON directory. By doing this, the name of the new Version 1.2–3 compiler, DECW$MOTIF.EXE, does not conflict with the previous compiler.
  2. Copy the DECW$XLIBSHR.EXE file (Xlib Release 4) to the SYS$COMMON:[SYSEXE.DECW$113] directory.

     The Xlib Release 5 version of Xlib has the same name; however, a logical name is defined to use the Release 4 version when the programs are linked. The changes in minor ident of DECW$XLIBSHR.EXE between DECwindows Motif Version 1.1 and DECwindows Motif Version 1.2 for OpenVMS caused a loss of backward compatibility with DECwindows Motif Version 1.1 systems.
  3. If you save the previous programming environment during installation, create the following two files and copy them to the SYS$COMMON:[SYSEXE.DECW$113] directory:

     - DECW$UILCOMPILER113.CLD

       This file enables UIL/OSF Motif and UIL/XUI to function for either XUI or OSF/Motif Release 1.1.3 if you select DECW$MOTIF113.EXE for OSF/Motif Release 1.1.3 or DECW$UILCOMPILER.EXE for XUI.
     - DECW$DEFINE113_LOGICALS.COM

       This file points to the saved header files, which in turn point to the Xlib Release 4 version of the DECW$XLIBSHR.EXE file.
  4. An application must be run on the same version (or higher) of the operating system as the version where it is linked. For example, an application that is linked on a VMS Version 5.4–3 system must be run on a VMS Version 5.4–3 (or higher) system.

     DECwindows Motif Version 1.1 applications that are compiled for OpenVMS Version 6.1 systems will run only on OpenVMS Version 6.1 systems and not on systems prior to Version 6.1.

V1.2 The X Window and OSF/Motif libraries that are shipped with DECwindows Motif Version 1.2 for OpenVMS software are incompatible with those shipped with previous versions. Run-time compatibility has been preserved, but the programming environment is not compatible.

Programming in the XUI or Motif Release 1.1.3 environment that is provided in previous versions of DECwindows Motif is no longer supported in DECwindows Motif Version 1.2 for OpenVMS. However, the installation procedure gives you the option of saving the programming files that already exist on your system. If you choose to save these programming files, they are moved to subdirectories where you can access them for programming. Specifically, the installation creates a subdirectory called [.DECW$113] in each of the directories listed in Table 4–3 and moves the previous files into the new subdirectory.

For more information about saving the programming files, see the *DECwindows Motif Version 1.2 for OpenVMS Installation Guide*.

Table 4–3 Directories for Previous XUI or Motif Programming Environment

Directory	Contents	New Location
DECW$INCLUDE	C header files	SYS$SYSROOT:[DECW$INCLUDE.DECW$113]
SYS$SYSTEM	UIL compiler	SYS$SYSROOT:[SYSEXE.DECW$113]
SYS$LIBRARY	Non-C language bindings	SYS$SYSROOT:[SYSLIB.DECW$113]

To program with these files, include the new [.DECW$113] subdirectories in the search path for each of the logical names. For example, perform the following:

- Change the definition of DECW$INCLUDE as follows:

```
$ SHOW LOGICAL DECW$INCLUDE
   "DECW$INCLUDE" = "SYS$SYSROOT:[DECW$INCLUDE]" (DECW$LOGICAL_NAMES)
        = "SYS$SYSROOT:[DECW$INCLUDE.EXTENSIONS]"

$ DEFINE/EXECUTIVE/TABLE=DECW$LOGICAL_NAMES DECW$INCLUDE -
 SYS$SYSROOT:[DECW$INCLUDE.DECW$113], -
 SYS$SYSROOT:[DECW$INCLUDE], -
 SYS$SYSROOT:[DECW$INCLUDE.EXTENSIONS]
```

Place the new [.DECW$113] subdirectory first in the search list, since many of the files that have been updated for this release have the same name as the files that were moved to the [.DECW$113] subdirectory. Thus, the files in [.DECW$113] are used for software development.

If DECW$INCLUDE is redefined in the SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM command procedure, then modify the previous instructions accordingly.

To use the UIL compiler for XUI or OSF/Motif Release 1.1.3, perform the following steps:

1. Create a file called SYS$SYSROOT:[SYSEXE.DECW$113]DECW$UILCOMPILER.CLD that contains the following text:

```
define type trace_keywords
 keyword tokens
 keyword symbols

define type warning_keywords
 keyword nowarnings
 keyword noinformationals

define type version_keywords
 keyword V1, syntax=xui_uil
 keyword V2, syntax=xui_uil, default
 keyword MOTIF11, syntax=motif_uil

define syntax xui_uil
 image decw$uilcompiler

define syntax motif_uil
 image decw$uilmotif
```

```
define verb uil
 image decw$uilcompiler
 parameter p1, label=source_file,
  prompt="File", value(required,noconcatenate,type=$infile)
 qualifier trace, label=trace_qual,
  value(list,noconcatenate,type=trace_keywords), nonnegatable
 qualifier warnings, label=warnings_qual,
  value(list,noconcatenate,type=warning_keywords)
 qualifier list, label=listing_file,
  batch, value(type=$outfile)
 qualifier machine, label=machine_qual,
 qualifier output, label=resource_file,
  default, value(type=$outfile)
 qualifier version, label=version_qual, default,
  value(type=version_keywords),
  nonnegatable
 qualifier XUI, default, nonnegatable, syntax=xui_uil
 qualifier MOTIF, nonnegatable, syntax=motif_uil

 qualifier widget_meta_description, label=widget_qual,
  value(required, noconcatenate, type=$infile)

 disallow XUI and MOTIF
```

2. Set the DCL command table to use the XUI UIL compiler as follows:

```
$ SET COMMAND SYS$SYSROOT:[SYSEXE.DECW$113]DECW$UILCOMPILER.CLD
```

Note

If you want to revert back to the previous UIL command definition, execute the following command:

```
$ SET COMMAND SYS$LIBRARY:DECW$UILCOMPILER.CLD
```

4.7 Adding POSIX for OpenVMS Support to DECwindows Transports

V1.1

Transports specific to DECwindows require code modifications to support POSIX for OpenVMS fork() requirements. If you have written a specific transport and also plan to implement POSIX for OpenVMS support, you must make changes to your transport source code. This note describes the required changes.

4.7.1 POSIX fork() Routine

V1.1

The POSIX fork() routine creates a copy of the currently executing process, completely cloning P1 and P0 space. Both the original process (the parent) and the new process (the child) proceed on identical paths from that point.

When using the POSIX fork() routine, data is not duplicated because:

- Files that are not opened through the POSIX for OpenVMS file system are not known to the child.
- Channels that are not shareable or opened through POSIX for OpenVMS are not known to the child.
- The AST queue of the parent is not copied to the child.

Because of these limitations, the POSIX for OpenVMS implementation has placed certain demands on protected shareable images, including transports specific to DECwindows:

- It is the responsibility of each protected shareable image to acknowledge that it can be forked.
- Based on how the child process should function, each protected shareable image must handle any cleanup or duplication of its own data in a routine called a fork callback.

Specific transports must acknowledge (on the client side) that they can be forked. The SYS$POSIX_FORK_CONTROL system service, described in Section 4.7.2, is provided for this purpose.

Specific transports need not perform any cleanup work.

4.7.2 SYS$POSIX_FORK_CONTROL System Service

V1.1

A new SYS$POSIX_FORK_CONTROL system service was added in VMS Version 5.5 that registers that a protected shareable image can be forked. It also registers the address and parameter to a fork callback.

The execution of the POSIX fork( ) routine fails if the application has been linked with, or has activated, a protected shareable image that has not called SYS$POSIX_FORK_CONTROL.

Therefore, in your specific transport's open routine, add a call to the SYS$POSIX_FORK_CONTROL system service similar to the following:

```
$ POSIX_FORK_CONTROL(CNTRL = POSIX$C_SETUP_CALLBACK,
  WHEN = POSIX$M_CALL_IN_CHILD,HANDLER = posix_fork_cb);
```

In this example, the CNTRL parameter indicates that the image is setting up a callback so that it can be forked, the WHEN parameter tells when to call the fork callback, and the HANDLER parameter gives the address of the fork callback.

Other parameters that are not used in this example allow for a parameter to be passed to the callback routine and to specify the access mode (maximized to that of the caller).

4.8 Translated Image Support

This section discusses restrictions and general information about translated-image support.

4.8.1 Running Translated Images on OpenVMS Systems

V1.2–3

Alpha

In versions of OpenVMS Alpha systems prior to Version 1.5, translation support was provided to address the following problems for users who migrated to OpenVMS Alpha systems:

- Lack of full-language programming support
- Unavailability of source code for recompilation
- Difficulty recompiling code that depended on VAX architecture features

For programming languages whose OpenVMS VAX versions are currently under development, native Alpha versions are available in the OpenVMS Alpha Version 6.1 operating system. The translated-image environment is maintained to support language features that are available as of the VAX VMS Version 5.5–2 release.

Similarly, translation is supported for images whose use of system services and run-time library entry points is restricted to those images that existed on the VAX Version 5.5–2 operating system.

If you have installed layered products on your system since the VAX Version 5.5–2 release, you may need to make additional changes if you rebuild an image to support translation. For instance, for applications that are included with DECwindows Motif Version 1.2–3 for OpenVMS VAX software, you must build images with the OSF Motif Release 1.1.3 library or the DECwindows XUI library, instead of with the OSF Motif Release 1.2.3 library.

The DECwindows Motif Version 1.2–3 for OpenVMS release includes the same translated-image support file that is provided with the DECwindows Motif Version 1.2 for OpenVMS Alpha release.

DECwindows Motif Version 1.2 for OpenVMS Alpha includes support for DECwindows Motif Version 1.1 for OpenVMS VAX images on an OpenVMS Alpha system.

V1.2 The DECwindows Motif Version 1.2 for OpenVMS Alpha release does not include support for running translated DECwindows Motif Version 1.2 for OpenVMS VAX images on OpenVMS Alpha systems.

If you want to run an application against DECwindows Motif Version 1.2 shareable images on an OpenVMS Alpha system, build it (compile and link) on an OpenVMS Alpha system. You cannot obtain a Version 1.2 translated image by building your application against the DECwindows Motif Version 1.2 shareable images on an OpenVMS VAX system, copying the resulting image file to an OpenVMS Alpha system, and translating it (using DECmigrate).

If you cannot build your application on an OpenVMS Alpha system and need to obtain a translated image with DECmigrate, you should continue to build your application against the OSF/Motif Release 1.1.3 programming environment provided in the DECwindows Motif Version 1.1 for OpenVMS VAX product. (Note that you are given the option to save your OSF/Motif Release 1.1.3 programming environment when you upgrade to DECwindows Motif Version 1.2 for OpenVMS VAX.) VAX images built against the OSF/Motif Release 1.1.3 programming environment can be copied to an OpenVMS Alpha system, translated, and executed successfully.

You can install translated-image support during the DECwindows Motif installation procedure. See the *DECwindows Motif Version 1.2–3 for OpenVMS Installation Guide* for instructions.

The OSF/Motif Release 1.1.3 shareable images that are used with translated images are different from and incompatible with the shareable images used for native images. As a result, the following image restrictions apply:

- Do not use both native and translated images that use DECwindows Motif software in a process.

  Either port or translate all the images that use DECwindows Motif software and all images that call each other.

- Translated images that dynamically activate a DECwindows Motif image using LIB$FIND_IMAGE_SYMBOL from a translated image are not supported.

For more information on translated-image support, see the *DECmigrate for OpenVMS AXP Systems Translating Images* guide.♦

4.8.2 DECwindows Motif Toolkit Functionality

V1.1

Alpha

On OpenVMS Alpha systems, DECwindows Motif Toolkit functionality is available with the following restriction: for the DECwindows Motif product, C modules are compiled using the default MEMBER_ALIGNMENT switch in the C compiler to maximize performance. It is recommended that you do not use the NOMEMBER_ALIGNMENT compile-time switch.

If you intend to port a toolkit application that is written in C, see the VAX C Run-Time Library release notes in the *OpenVMS Alpha AXP Version 1.0 Release Notes*.♦

4.9 Cross-Development Tools

V1.2

The OpenVMS Alpha Migration Toolkit runs on the OpenVMS VAX operating system and supports development for OpenVMS Alpha operating systems. To develop DECwindows Motif applications in a multiplatform environment, you need the OpenVMS Alpha Migration Toolkit and the DECwindows Motif Version 1.2 for OpenVMS Alpha software. DECwindows Motif Version 1.2 for OpenVMS is not part of the OpenVMS Alpha Migration Toolkit.

The following steps explain how to extract DECwindows Motif files from the save sets on the DECwindows Motif Version 1.2 for OpenVMS Alpha kit for use in a multitools environment.

Note

Because of OSF requirements, you must purchase a DECwindows Motif Version 1.2 license to unpack the DECwindows Motif save sets for multiplatform development.

During the VMSINSTAL installation of the OpenVMS Alpha Migration Toolkit, the logical name ALPHA$LIBRARY is created. This logical name is used as the destination of the files in the OpenVMS BACKUP commands that follow.

Unpack DECwindows Motif Version 1.2–3 for OpenVMS Alpha save sets for multiplatform development into the ALPHA$LIBRARY directory using the BACKUP command.

- Use the following commands to unpack the B and C save sets, which contain all bindings:

```
$ BACKUP DWMOTIF_AXP012.B/SAVE/SELECT=[SYSLIB]*.* ALPHA$LIBRARY:
$ BACKUP DWMOTIF_AXP012.C/SAVE/SELECT=[SYSLIB]*.* ALPHA$LIBRARY:
```

- Use the BACKUP command to unpack the language bindings from the following save sets:
  - E (MIT C bindings)
  - J (Fortran bindings)

  - L (Pascal bindings)

  For example:

```
$ BACKUP DWMOTIF_AXP012.E/SAVE ALPHA$LIBRARY:
$ BACKUP DWMOTIF_AXP012.J/SAVE ALPHA$LIBRARY:
$ BACKUP DWMOTIF_AXP012.L/SAVE ALPHA$LIBRARY:
```

- Use the BACKUP command to unpack the files from the following save set:
  - Q (Translated image support)

  For example:

```
$ BACKUP DWMOTIF_AXP012.Q/SAVE/SELECT=[SYSLIB]*.IIF  ALPHA$LIBRARY:
```

  Note that you only need the .IIF files from the Q save set.

The *OpenVMS Alpha Migration Toolkit Reference Manual* provides programming reference information on the tools included in the OpenVMS Alpha Migration Toolkit.

4.10 Compiling Applications Written in C

V1.0

During the VAX C layered-product installation procedure, you have the option of extracting the VAX C definition files (.h files) or leaving the .h files in the text library. If you extract the definition files, you can use #include control lines of the following form:

```
#include <filename.h>
```

The DECwindows header files assume that the .h files were extracted. They contain #include <module_name.h> notation for the included files. The DECwindows programming documentation also makes this assumption.

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

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

With DEC C, the header files do not need to be extracted from the text library if you define the DECC$TEXT_LIBRARY logical name as follows:

```
$ DEFINE DECC$TEXT_LIBRARY SYS$LIBRARY:DECC$RTLDEF.TLB
```

4.11 Compiling Applications Written in Fortran

V1.1

Alpha

On OpenVMS Alpha systems, some of the include files used for writing DECwindows Motif applications in Fortran, such as the DECW$MOTIF.FOR file, contain structure definitions that cause memory layout changes, depending on the compiler switches used.

You can do one of the following:

- Use the CDEC$ OPTIONS in the source code so that your Fortran programs work correctly with the DECwindows Motif run-time libraries:
  - Before you include DECwindows Motif files in your program, add the following statement:

```
CDEC$ OPTIONS /ALIGN=RECORDS=NATURAL
```

- Following the INCLUDE statement, add the following statement:

```
CDEC$ END OPTIONS
```

For example:

```
CDEC$ OPTIONS /ALIGN=RECORDS=NATURAL
INCLUDE "DECW$MOTIF.FOR"
CDEC$ END OPTIONS
```

- Use the Fortran compiler switch, /ALIGN=RECORDS=NATURAL, when you compile your programs.♦

4.12 CDA Application

This section contains programming information about the CDA application.

4.12.1 CDA—Implementing the Drag-and-Drop Feature

V1.2 The drag-and-drop feature, which lets you move or copy information between widgets, is implemented in the widgets listed in Table 4–4.

Table 4–4 Drag-and-Drop Widgets

Widget	Drag Operation	Drop Operation
XmText	copy and move	copy and move
XmTextField	copy and move	copy and move
XmLabel	copy	
XmPushButton	copy	
XmToggleButton	copy	
XmList	copy	

For information about how to include additional drag-and-drop functionality in applications and for an example of a drag-and-drop program, see the *Open Software Foundation: OSF/Motif Programmer's Guide, Revision 1.2* manual.

4.12.2 CDA Run-Time Services Features

V1.2 This section describes new programming features in this version of the CDA Run-Time Services. See Chapter 2 for new user features.

4.12.2.1 Version Number

V1.2–3 DECwindows Motif Version 1.2–3 for OpenVMS software incorporates additional problem resolutions that were not provided in the general release of the CDA Version 1.8 product.

CDA Version 1.6 was part of the DECwindows Motif Version 1.1 release. DECwindows Motif Version 1.2 for OpenVMS software provides the CDA Run-Time Services component of CDA Version 1.8; DECwindows Motif Version 1.2–3 for OpenVMS incorporates Version 1.8A.

4.12.2.2 Changes to the Programming Interface for CDA

V1.2

This section describes the changes to the programming interface for this version of CDA Run-Time Services.

This version provides a new set of header files that define CDA constants, types, and routines using portable naming conventions. By using these new naming conventions, you can use a wider variety of C compilers to minimize the amount of system-specific code in your CDA applications.

The names of the new set of header files are the same as the names of the previous set of header files, except that the dollar sign ($) has been removed. For example, the cda$msg.h include file is now called cdamsg.h. Other examples include the following: The DDIF$K_DSC_MAJOR_VERSION symbol is now declared as DDIF_K_DSC_MAJOR_VERSION, and the CDA$_NORMAL status value is now defined as CDA_NORMAL.

The previous set of header files is also included in this version, but these files will no longer be updated. Changes introduced since the release of DECwindows Motif Version 1.1 (for example, the new definitions for audio support), are available only in the new set of header files. To use the new CDA features, change the file names in your source code.

The new set of header files supplements the previous set of header files. If you want to write ANSI-compliant applications using CDA definitions and CDA Toolkit calls, use the new set of header files. However, you can continue to use the header files that define symbols containing the dollar sign ($) provided you choose a non-ANSI compilation mode.

By using the previous set of header files, you can successfully build existing source code that uses the previous naming conventions.

See Table 4–5 for a list of new header file names.

Table 4–5 New Header File Names

Previous Name	New Name
cda$def.h	cdadef.h
cda$msg.h	cdamsg.h
ddif$def.h	ddifdef.h
dtif$def.h	dtifdef.h
cda$ptp.h	cdaptp.h
cda$typ.h	cdatyp.h
dvr$msg.h	dvrmsg.h
dvr$cc_def.h	dvrccdef.h
dvr$cc_ptp.h	dvrccptp.h
dvr$decw_def.h	dvrwdef.h
dvr$decw_ptp.h	dvrwptp.h

4.12.2.3 Changes to External Reference Processing

V1.2

CDA Run-Time Services supports relative file specifications for external references. Relative references are also supported: that is, a reference where the directory path is not fully specified but is relative to the directory path of the parent document.

4.12.2.4 Message for Style Guide Fallback

V1.2

The following message is displayed when a local style guide cannot be found:

STYGDEFBK, Fallback to nonlocale-specific style guide: *file-spec*

Level: Informational

Explanation: Since the locale-specific style guide cannot be found, the nonlocale-specific version of the style guide is used.

4.12.2.5 Using Logical Names with CONVERT Commands

V1.2

During a document conversion, if a logical name is used to specify the directory of the primary document and the document file extension is omitted, external references that contain relative file specifications cause the conversion to fail.

This failure occurs because the back-end converter examines the converter processing options, although the front-end converter opens the input file. Because the converter software cannot predict what default file extension the front-end converter might apply, it uses the OpenVMS file services to resolve the logical name and to find the actual file specification so that the directory can be parsed and extended with a relative specification.

The conversion fails only when the external reference uses a format such as the following:

```
sys$login:mydoc
```

In this case, the reference will not be resolved. The reference is resolved in all other cases. For example:

```
sys$login:mydoc.doc
disk$:[smith]mydoc
disk$:[smith]mydoc.doc
```

4.12.2.6 Restructuring Shareable Images

V1.2

The CDA Viewer includes two shareable images to allow installation on systems where DECwindows is not installed.

In DECwindows Motif Version 1.2 for OpenVMS software, shareable images that use the X services are renamed. Table 4–6 lists the shareable images in the various versions of the CDA Viewer.

Table 4–6 Names of Shareable Images

CDA Version	Image Name	Description
Version 1.6	CDA$ACCESS	CDA Run-Time Services shareable image.
	DDIF$VIEWSHR	Callable viewer widget.
Version 1.7	CDA$ACCESS	CDA Run-Time Services shareable image.
	DDIF$VIEWSHR DDIF$DECW_VIEWSHR DDIF$CC_VIEWSHR	The DDIF$DECW_VIEWSHR widget is a callable viewer widget that uses the LIB$FIND_IMAGE_SYMBOL routine to invoke the DDIF$DECW_VIEWSHR (DECwindows interface) and DDIF$CC_VIEWSHR (character-cell interface) widgets.
Version 1.8 Version 1.8A	CDA$ACCESS	CDA Run-Time Services shareable image.
	DDIF$VIEWSHR12	The DDIF$VIEWSHR12 widget is a callable viewer widget that uses LIB$FIND_IMAGE_SYMBOL to invoke the DDIF$DECW_VIEWSHR12 (DECwindows interface) and DDIF$CC_VIEWSHR (character-cell interface) widgets.

By using the LIB$FIND_IMAGE_SYMBOL routine to reference the entry points to the DDIF$DECW_VIEWSHR, DDIF$DECW_VIEWSHR12, and DDIF$CC_VIEWSHR images, an application can dynamically determine whether it can execute in a given environment. The DDIF$VIEW.EXE application now replaces this routine.

The previous DDIF$VIEWSHR.EXE shareable image is still included to maintain compatibility with applications linked against it. However, new applications (and previous applications that take advantage of new features) should use the new shareable images.

4.13 DECterm Programming

This section contains information about DECterm programming.

4.13.1 Page-Movement Escape Sequences

V1.2 The following page-movement escape sequences are implemented in DECterm:

```
NP      <CSI> Pn U      Next Page
PP      <CSI> Pn V      Previous Page
PPA     <CSI> Pn P      Page Position Absolute
PPB     <CSI> Pn R      Page Position Backward
PPR     <CSI> Pn Q      Page Position Relative
```

Note that “Pn” is the number of pages to move; the exception is PPA, where “Pn” is the actual page number.

Note

DECterm does not support cursor coupling; the cursor is always bound to the current (displayed) page.

4.13.2 DECCRA Sequence

V1.2 DECterm supports the DECCRA (Copy Rectangular Area) sequence in a limited way. The entire page needs to be copied at once, and either the source or the destination page must be the current page (for example, you cannot copy from one off-screen page to another).

4.13.3 DECLFKC Sequence

V1.2 Since DECterm does not have a user interface to redefine the F5 (Break) key, the DECLFKC (Local Function Key Control) sequence redefines the F5 key when “0” (all keys) is used to select which key is to be modified.

4.13.4 ReGIS Input Cursors

V1.2–3 DECterm supports the following input cursors: cross-hair, rubber-band line, diamond, and rubber-band rectangle. To select input cursors use the S(C(I*n*)) command. Table 4–7 shows the values of *n*.

Table 4–7 ReGIS Input Cursors—Cursor styles and Values

Cursor Style	Variable *n*
Cross-hair	Omitted
Cross-hair (default)	0
Diamond	1
Cross-hair	2
Rubber-band line	3
Rubber-band rectangle	4

Note

If a shape other than the diamond cursor is desired when *n* is equal to 1, define the logical name DECW$DECTERM_REGIS_CURSOR to be one of the numbers defined in the SYS$LIBRARY:DECW$CURSOR.H file.

4.14 DECwindows OSF/Motif Toolkit

This section contains information about notes, restrictions, and corrections for the DECwindows OSF/Motif Toolkit.

4.14.1 Compilation Error When Including Both CURSES.H and XMP.H Files

V1.2 When compiling a C program that includes the files curses.h and XmP.h, or any Motif header file that causes XmP.h to be included, such as LabelP.h, a compilation error similar to the following may be generated:

```
Boolean move;
%CC-E-MACREQARGS, Macro reference requires an argument list;
"move" not substituted.
At line number 407 in SYS$COMMON:[DECW$INCLUDE]XMP.H;1.
```

This error occurs as a result of the file curses.h defining a macro named "move", which conflicts with a member field in an OSF/Motif Release 1.2.2 structure for drag-and-drop support.

This error can be prevented by placing the include statement for the file curses.h after the include statements for any DECwindows Motif header files.

4.14.2 Motif Text Widget Translations

V1.0 To implement Digital virtual bindings, it is necessary for Digital to modify the default XmText and XmTextField translation manager syntax. Specifically, the following syntax line is removed:

```
Shift ~Ctrl ~Meta ~Alt <Key>osfDelete:   cut-clipboard()
```

If you use virtual bindings in which the osfCut virtual keysym is not bound, no key sequence is bound to the cut-clipboard action by default. To work around this limitation, override the XmText and XmTextField translations in your DECW$XDEFAULTS.DAT file.

4.14.3 Upward Compatibility

V1.0 The combination of DECW$DWTLIBSHR, DECW$DWTSHR, and DECW$XTSHR shareable images is intended to be binary and upwardly compatible with the previous DECW$DWTLIBSHR shareable image. A possible exception is applications that dynamically activate the image DECW$DWTLIBSHR using LIB$FIND_IMAGE_SYMBOL. The problem with dynamic image activation is that the semantics of the Intrinsics have changed in places between the X Toolkit Intrinsics Release 3 to the MIT X11 Release 4 Intrinsics. A previously linked XUI application calls through entries in the XUI transfer vector that use the X Toolkit Intrinsics Release 3 semantics of the Intrinsics. However, as soon as that application relinks (or dynamically activates DECW$DWTLIBSHR), it uses the MIT X Toolkit Intrinsics Release 4 semantics of the Intrinsics.

Applications that dynamically activate DECW$DWTLIBSHR should be changed to link directly against DECW$DWTLIBSHR. DECW$DWTLIBSHR is now a small, thin-layer, shareable image that dynamically activates automatically the majority of the toolkit code only when necessary. There is no reason for applications to continue to dynamically activate DECW$DWTLIBSHR.

In addition, Digital cannot guarantee binary upward compatibility in the future for applications that dynamically activate the toolkit shareable images. DECwindows toolkits (XUI and Motif) are based upon standards that Digital does not control. When the standards make an incompatible change, Digital will change the toolkit to follow it. However, the intent is to add code that allows existing executable images to run unchanged.

4.14.3.1 Restrictions on Mixing Motif and XUI Widgets

V1.0 Mixing Motif and XUI widgets is restricted. The problem results from the fact that both XUI and Motif have their own Vendor Shell widget class. If an application is linked against XUI and not Motif, the XUI Vendor Shell widget class is used in order to maintain compatibility with DECwindows XUI. If an application is linked against Motif, the Motif Vendor Shell widget class is used. Motif widgets require the Motif Vendor Shell; XUI widgets are compatible with the Motif Vendor Shell. The problem occurs when an XUI-only application dynamically activates (using LIB$FIND_IMAGE_SYMBOL) a shareable image that uses Motif. The toolkit makes the decision to use the XUI Vendor Shell when the toolkit is initialized and the Motif widgets in the dynamically activated shareable image do not work.

The workaround is to add the application image name to the DECW$USE_XM_VENDOR_SHELL logical name. This logical name contains a comma-separated list of image names for which the toolkit is to use the Motif Vendor Shell. The value of DECW$USE_XM_VENDOR_SHELL logical name defaults to NOTES$MAIN. To add additional image names to this logical name, enter the following:

```
$ DEFINE DECW$USE_XM_VENDOR_SHELL "NOTES$MAIN,-
_$ yourimage1,yourimage2,..."
```

Note that DECW$USE_XM_VENDOR_SHELL does not affect applications that use Motif since they are already using the Motif Vendor Shell.

4.14.4 Compile-Time Incompatibilities in Motif Header Files

V1.2–3 Several macro definitions are removed from the Motif header files in the OSF/Motif programming support. The changes affect the following header files:

- DECW$INCLUDE:XMP.H

  The following definitions are removed:

  - #define XmLONGBITS sizeof(Cardinal)*8
  - #define XmHALFLONGBITS (XmLONGBITS/2)

  These macros are replaced in the same header file with the following macro:

  - XmOFFSETBITS

  Update the code to call the XmOFFSETBITS macro.

- DECW$INCLUDE:TEXTP.H

  The following definitions are removed:

  - #define MAXINT 2147483647
    /* Biggest number that can fit in long */
  - #define NODELTA MAXINT

  The definition of MAXINT is operating-system dependent. Include this definition in applications in one of the following ways:

  - Include the appropriate system-header file
  - Define the constant

4.15 OSF/Motif Example Programs

This section describes example programs from the Open Softare Foundation (OSF) that are included in the DECwindows Motif Version 1.2 for OpenVMS software.

4.15.1 Cut and Paste Example Program

V1.2 This example demonstrates the use of the Motif Clipboard. You can cut and paste application-defined data formats using the XmClipboard API.

Run two cut/paste clients and transfer graph values using the pull-down and pop-up menus.

4.15.2 DNDDemo Example Program

V1.2 This example illustrates a typical use of the drag-and-drop feature.

When the example starts, it does not have any valid drop sites. It starts with different color rectangles that act as drag sources for dragging the colors around. Create the drop sites as follows:

- Move the pointer into the drawing area (with white background).
- Press MB1 and drag. You see the “rubber-banding” effect.
- Release MB1 after dragging a desired distance.

  A black rectangle appears. You can create as many rectangles as you like.

The rectangles inside the drawing area that you created act as valid drop sites for colors. You can test this by initiating the drag from the yellow color rectangle and dragging over to one of the rectangles inside the drawing area and releasing MB2.

To initiate a drag, move the pointer over to the drag source, press BTransfer (which is MB2 by default), and start to drag. Notice that the cursor changes to a painter’s palette in the same color as the color rectangle from which the drag is initiated.

While you are dragging, notice the following:

- When dragging over the root window, the drag icon consists of only the painter’s palette.
- When dragging over parts of the drawing area where there are no rectangles, the drag icon changes to include a "DO NOT ENTER!" sign on top of the painter’s palette.
- When dragging over any of the rectangles inside the drawing area, the "DO NOT ENTER!" sign is replaced with a painter’s brush.

To perform a drop, release MB2. If you release MB2 with the pointer over any part of the drawing area outside of the rectangles, or anywhere on the root window, the drag icon snaps back to the point of drag initiation and disappears. This is an indication that the drop you attempted failed. If you release MB2 with the pointer over any of the rectangles inside the drawing area, the drag icon disappears into the background; this indicates that the drop was a success. The rectangle on which the drop was attempted, becomes filled in with the color that was dragged.

The rectangles inside the drawing area act as drag sources. Initiate the drag with the pointer on any rectangle, and the cursor changes to a rectangle (either of the same size or a smaller size depending on the X cursor size constraints). Drag the rectangle from one place and drop it onto a new place inside the drawing area. The rectangle is physically moved to the new position.

If you want to copy the rectangle to a new position, use the appropriate modifier key while dragging, or at the time you attempt the drop (Ctrl is the modifier key for copy in the current implementation). When you drag a rectangle with no modifier key pressed or with the Shift key pressed, the solid rectangle from where the drag was initiated is replaced by a hollow rectangle with dotted-line borders. It continues this way for the duration of the drag or until you press the Ctrl key, which changes the operation to Copy. At that time it regains its original solid form.

Note

During dragging you can cancel the drag by pressing the Cancel key (F11). Also, you can press the Help key to get information about whether the drop will succeed and possible drop operations.

4.15.3 Dogs Example Program

V1.1 This example uses the dog and square widgets. It shows how to incorporate new widgets into the UIL source by using the user-defined function.

The dogs example program allows you to change the DogNwagTime and SquareNmakeSquare resources dynamically.

If you have a system with sound generation features, you might want to change the bark callback to something other than XBell( ).

4.15.3.1 Dog Widget

V1.1 The dog widget (DogWidget) demonstrates how to subclassify a primitive widget that remains binary compatible with future versions of Motif. It uses XmResolvePartOffsets( ) and associated macros and implements all the recommendations in the XmResolvePartOffsets manpage.

The dog widget is a subclass of XmPrimitive. It can bark and wag its tail. If you want more advanced tricks, you must subclassify it, or replace up.bm, down.bm, and bark.bm with more advanced bitmaps.

The dog widget has the following resources:

- DogNwagTime: Time in milliseconds between each wag
- DogNbarkTime: Time in milliseconds the bark graphic is displayed
- DogNbarkCallback: Callback called by the bark action

The dog widget has the following translations:

- osfActivate/Return/Space/MB1 = Bark
- W/MB2 = Wag tail
- S/Shift-MB2 = Stop wagging tail
- osfHelp = Help

4.15.3.2 Square Widget

V1.1 The square widget (SquareWidget) demonstrates how to subclassify a constraint widget that remains binary compatible with future versions of Motif. It uses XmResolveAllPartOffsets( ) and associated macros and implements all the recommendations in the XmResolveAllPartOffsets manpage.

The square Widget is a subclass of XmBulletinBoard. It forces its children to be square using a constraint resource.

The square widget has the resource SquareNmajorDimension. This resource determines which dimension is used for the new size of the child. Values are SquareWIDTH or SquareHEIGHT.

The square widget has the constraint resource SquareNmakeSquare. This resource determines whether the child is forced to be square or set to its preferred shape.

4.15.4 Hellоint Example Program

V1.2

This example is a simple UIL/Xm program with a label and a push button.

With this version, you can choose a Kanji, Hebrew, French, or English interface at run time.

The program uses the file LOCALSTRINGS.UID for all language-specific interfaces. To change languages, copy one of the LOCALSTRINGS_*.UID files to DECW$USER_DEFAULTS:LOCALSTRINGS.UID. Then, use the logical name LANG or the xnlLanguage resource to set the locale. Set the LOCALSTRINGS suffix and the locale name to one of the following:

Locale	Interface Type
Japan	Kanji
Hebrew	Hebrew
French	French
English	Standard English

To define the xnlLanguage resource, add the following line to your DECW$XDEFAULTS.DAT file:

```
*xnlLanguage:locale
```

The following fonts are needed for the helloint example program:

Locale	Font Name
French	-adobe-helvetica-medium-r-normal–12-120-75-75-p-67-iso8859-1
English	-adobe-times-medium-r-normal–14-100-100-100-p-74-iso8859-1
Hebrew	-Misc-Fixed-Medium-R-Normal–13-120-75-75-C-80-ISO8859-8
Japan	-*-JISX0208.1983-1
	-*-JISX0201.1976-0

If you do not have these fonts installed on your system, either change the UIL files to reference other fonts or install the *.DECW$BDF fonts provided in the DECW$EXAMPLES directory. See *Managing DECwindows Motif for OpenVMS Systems* for instructions on how to install new fonts.

4.15.5 Hellomotif Example Program

V1.2

This example is a simple UIL/Xm program with a label and a push button.

4.15.6 Motifanim Example Program

V1.1

The motifanim program demonstrates the following OSF/Motif features:

- Declaration of icon/bitmap using UIL for portability across visuals
- Exchange of value between UIL and C using Mrm
- Use of the Motif form widget
- Background procedure running while the client process is waiting for input (Xt intrinsics feature)

This example lets you animate a set of pixmaps in an X window. The pixmaps are not fixed by motifanim but are read at startup from existing UID files.

The syntax of all the animation.uid files can add a new set of pixmaps to be animated using motifanim. Some tree examples of animation data files include:

dog.uil
plane.uil

The default animation is dog. To run the example with the other animations, use the following commands:

```
$ SET DEFAULT DECW$EXAMPLES
$ motifanim := "$DECW$EXAMPLES:motifanim"
$ motifanim -anim plane
```

The files motiflogo.uil, motifanim.uil, and motifanim.c are the core components of the program.

4.15.7 Motifgif and Pict Viewing Programs

V1.2 The following .gif files are provided:

- TOUCAN.GIF, which is a picture of a tropical bird.
- CHALLENGER.GIF, which is a picture of the space shuttle Challenger.

V1.1 The motifgif and pict programs work together to display a .gif formatted picture. The motifgif program displays a .gif file within Motif constructs. The motifgif program looks for .gif files in the current directory.

The motifgif program has three significant components:

- motifgif

  The source and executable code in this directory is a Motif program driving the display of a .gif picture.
- pict

  The program called by motifgif that displays a bitmap of a .gif picture file. This program used to be xgif but has since been converted to Motif.

The following problems exist with motifgif:

- Resizing a picture does not work.
- The -d option is not used when calling pict, so remote displays must set the DISPLAY environment variable.

4.15.8 Motifshell Example Program

V1.2 The motifshell program is an example of an environment created using Motif. The purpose of the example is to show how Motif may be used as a standalone environment for dispatching other programs and utilities.

The motifshell program is written in C code. There are no UIL references.

Note

Displaying a file listing or process status takes some time. These operations are accomplished by spawning a DCL command, capturing the output in a file, and then reading the file.

The default font is proportionally spaced. Some listings are more readable if a fixed space font is used, such as Courier. Use the Font option to select a different font.

4.15.9 Periodic Example Program

V1.2

The periodic example is a demonstration of the displayable widgets in Motif. The program displays Motif widgets in a periodic chart format.

4.15.10 Textedit Example Program

V1.2

The textedit program is a primitive text editor based on the Motif XmText widget that makes use of its XmNsource resource to allow multiple views of the same underlying text.

Table 4–8 describes the View menu options.

Table 4–8 View Menu Options

Option	Description
Split Pane	Adds a new independently scrollable view of the text.
Remove Pane	Removes a designated pane.
One Pane	Removes all panes except the designated pane.

The designated pane is determined in the following way:

- If using an explicit focus policy, it is the last view which had focus.
- If using a pointer focus policy, it is the last view with which the user interacted.

The designated pane is also the one whose clipboard contents are pasted when Paste is selected from the Edit Menu.

The textedit sources are separated into independent layers. Textedit contains the source files listed in Table 4–9.

Table 4–9 Textedit Source Files

File Name	Description
tfile.c	The code to read, write, and remove files. Hides any operating system or filesystem dependencies.
app.c	Uses tfile.c to provide all the code to manipulate files and their related buffers. This is toolkit-dependent.
tk.c	Isolates dependencies on the Xm Toolkit. The first half of the file provides utility routines and a toolkit-independent (but application-specific) interface that can be used by the toolkit-independent dialog layer of the application. The second half of the file defines all the callback routines. Many of these update state internal to this layer; others perform additional actions. Those actions unambiguously correspond to toolkit-specific action routines defined in the first half of the file. Otherwise, an upcall is made to the toolkit-independent dialog layer to decide what to do.
dlg.c	Uses tk.c to manage dialog with the user. The code is toolkit-independent.
textedit.c	Initializes the application and instantiates widget hierarchies. The code hides the use of UIL and Mrm from the rest of the application.

Note

Moving and removing files is not currently implemented.

4.15.10.1 Additional Translations

V1.2

Users familiar with emacs may want to add the following translations to their DECW$XDEFAULTS.DAT files:

```
*XmText.translations:     Mod1<Btn3Down>: scroll-cursor-vertically()\n\
  Mod1<Btn3Motion>: scroll-cursor-vertically()\n\
 Ctrl<key>a: beginning-of-line()\n\
 Ctrl<key>b: backward-character()\n\
 Ctrl<key>d: delete-next-character()\n\
 Ctrl<key>e: end-of-line()\n\
 Ctrl<key>f: forward-character()\n\
 Ctrl<key>g: beep()\n\
 Ctrl<key>h: delete-previous-character()\n\
 Ctrl<key>i: cut-primary()\n\
 Ctrl<key>j: newline-and-indent()\n\
 Ctrl<key>k: set-anchor() end-of-line() key-select() cut-clipboard()\n\
 Ctrl<key>l: redraw-display()\n\
 Ctrl<key>m: newline()\n\
 Ctrl<key>n: next-line()\n\
 Ctrl<key>o: newline-and-backup()\n\
 Ctrl<key>p: previous-line()\n\
 Ctrl<key>v: next-page()\n\
 Ctrl<key>w: cut-clipboard()\n\
 Ctrl<key>y: paste-clipboard()\n\
 Ctrl<key>z: scroll-one-line-up()\n\
 Mod1<key>b: backward-word()\n\
 Mod1<key>d: delete-next-word()\n\
 Mod1<key>f: forward-word()\n\
 Mod1<key>h: delete-previous-word()\n\
 Mod1<key>i: copy-primary()\n\
 Mod1<key>k: delete-to-end-of-line()\n\
 Mod1<key>v: previous-page()\n\
 Mod1<key>w: copy-clipboard()\n\
 Mod1<key>z: scroll-one-line-down()\n\
 Mod1 Shift<key>greater: end-of-file()\n\
 Mod1<key>less: beginning-of-file()\n\
 Mod1<key>]: forward-paragraph()\n\
 Mod1<key>[: backward-paragraph()
```

4.15.11 View Example Program

V1.2

The view program is similar to the DCL command TYPE/PAGE using Motif with internationalization support. It allows you to view files in several languages.

The command file DECW$EXAMPLES:FILEVIEW.COM is provided, which allows you to select a language. Valid languages are French, English, and German. To select a language, perform the following commands:

```
$ SET DEFAULT DECW$USER_DEFAULTS
$ @DECW$EXAMPLES:FILEVIEW language
```

This command file copies the .UID and .DAT files to the current directory. It sets the appropriate locale and executes the file viewing example program.

When the application is started, a primary top-level shell is created. From the primary top-level shell, you can create secondary shells.

Each top-level shell is a parent of a Main Window, the work area of which is a PanedWindow.

The menu bar has the following entries:

- File—opening and closing files, creating new shells, and exiting

- View—controlling the panes in the pane window

File Menu

The file menu contains the following options:

- Open New File

  A file selection box is mapped to choose the file. If OpenFile is successful, the current file is closed, all existing panes are destroyed, and the new file is displayed.

- Open New Shell

  Creates a secondary shell similar to the primary shell. Files can be viewed in each shell independently.

- Close

  This entry only exists on the secondary windows. It destroys the top-level shell and closes the file.

- Exit

  This entry only exists on the primary shell; it closes the file and the shell.

View Menu

The view menu contains the following options:

- New Pane

  Creates a new pane in the paned window.

- Delete Pane

  Deletes the current pane.

- Search

  Causes a dialog box to pop up for searching text in the file. The OK callback of the dialog box searches the string. If the string is found, it is displayed in the current pane. If the string is not found, then the dialog box pops up.

Opening a second file in a window causes the application to end abruptly. The View options do not work correctly.

4.15.12 Xmpiano Example Program

V1.2 The xmpiano program shows how to write a Motif Interface to Dumb Instruments (MIDI) application. Both a staff, for writing music, and keyboard are provided.

At the bottom of the window is a set of notes that may be used on the staff. Selecting one of these note buttons changes the active note accordingly. The selected note is also displayed as the new mouse cursor.

Though the note selection is limited, it is possible to play sharps as well as naturals. The program interface has not been written to play flats. The staff automatically resizes larger as notes are added past the right side of the staff. To see these notes, scroll the score window as needed.

To use this application, press the right mouse button on a staff that is visible. The following menu items are displayed:

- Add Voice

Connects another display to a new staff. After selecting this command, you are prompted for the display to connect. Any music written in the new staff plays to this display. Note that pressing the right mouse button on the new staff shows the name of the display in the menu title. In addition, the menu commands in the menu bar relate to the staff that is being displayed.

- Remove Voice

  Removes the staff as well as the connection to the corresponding display. There is no undo for this command—all music written in this staff is deleted.

- Clear Voice

  Erases all notes on the staff. This does not affect the display connection.

- Play Voice

  Plays the voice of the staff in which the right mouse was pressed. If the voice is the same as the local host, the keys on the keyboard that correspond to the notes appear as though they are being pressed as the notes are played. It is possible to make the keyboard play along with all voices at the same time; however, the code must be compiled with the -DCHORDS option. Note that this can slow down the application significantly on many servers.

- Play All

  Plays all voices at the same time on each of the specified displays.

- Save Voice

  Saves the corresponding voice to a file. A FileSelectionDialog is displayed to prompt for the name of the file in which to save the voice. The display connection information is not saved.

- Load Voice

  Loads (appends) a previously saved voice to the corresponding staff.

- Quit

  Exits the example program.

This version does not allow editing of notes. To edit, clear the staff and start again, or read the data from a saved file.

To set the color of the notes, change the foreground color of the application. To do this, use the -fg option on the command line. For example:

```
$ xmpiano :== $DECW$EXAMPLES:xmpiano
$ xmpiano -fg blue
```

4.15.13 Motif Sample Programs

V1.2

Table 4–10 lists sample programs showing various Motif Toolkit functionality.

Table 4–10 Motif Sample Programs

File Name	Description
xmdialogs.c	dialog sampler
xmfonts.c	font browser
xmeditor.c	text editor

(continued on next page)

Table 4–10 (Cont.) Motif Sample Programs

File Name	Description
xmlist.c	list example
xmprotocol.c	window manager protocols example
xmter.c	shape & animation example
xmform.c	form attachment example
xmforc.c	form attachment + rowcolumn example
xmmap.c	drawingarea + scrolledwindow example
xmgetres.c	resource fetching example
xmapdef.c	app defined scrolled window example

The following notes apply to these sample programs:

1. The xmfonts program defaults to displaying the fonts whose name length are less than 10 characters. On some systems, the font path contains only long XLFD font names. It may be necessary either to set the resource XMFONTS*maxLen to a larger number (80), or to specify a numColumns resource of 1.
2. If you run the program xmter with a window manager other than Motif Window Manager, that window manager has to remove all the window decorations (for example, borders). The program xmter directs Motif Window Manager to remove the window decorations.
3. The xmform program displays a string made of Motif widgets embedded in a Form. Use the following commands:

   ```
   $ xmform :== $DECW$EXAMPLES:xmform
   $ xmform string
   ```

   The variable *string* is optional. If *string* is supplied, the available letters for *string* are F,I,M,O,T. The string of letters that you use must be in uppercase and enclosed in quotation marks. If *string* is omitted, xmform returns "MOTIF".
4. The xmgetres program is an example of how to use XmGetSecondaryResourceData. Use the following commands to invoke xmgetres:

   ```
   $ xmgetres :== $DECW$EXAMPLES:xmgetres
   $ xmgetres WidgetClass
   ```

   The variable *WidgetClass* is optional. If *WidgetClass* is omitted, it defaults to a class named “Widget”. You can also use the class “All”, which displays the resources for all the Xt and the Motif widgets. See the *OSF/Motif Programmer’s Reference* manual for a list of available widget classes.

4.15.14 Xmtravel Example Program

V1.2

The xmtravel example is a front end to a travel agent client and flight database. The program is designed to illustrate various user-interface design concepts as well as be compliant with the OSF/Motif Style Guide.

The program is just an example, many of the functions are either not implemented or use predefined settings.

4.15.15 Resource Files for Example Programs

V1.2 Many of the example programs have associated resource files for defining various display attributes. To use these files, either copy them from the directory DECW$EXAMPLES to your DECW$USER_DEFAULTS directory, or add their contents to your DECW$XDEFAULTS.DAT file.

The list of example programs and the resource files associated with them is as follows:

Example Program	Resource File
DECW$CDPLAYER.EXE	DECW$CDPLAYER.DAT
FILEVIEW.EXE	FILEVIEW.DAT
PERIODIC.EXE	PERIODIC.DAT
MOTIFANM.EXE	MOTIFANIM.DAT
XMAPDEF.EXE	XMDEMOS.DAT
XMDIALOGS.EXE	XMDEMOS.DAT
XMEDITOR.EXE	XMDEMOS.DAT
XMFONTS.EXE	XMDEMOS.DAT
XMFORC.EXE	XMDEMOS.DAT
XMFORM.EXE	XMDEMOS.DAT
XMGETRES.EXE	XMDEMOS.DAT
XMLIST.EXE	XMDEMOS.DAT
XMMAP.EXE	XMDEMOS.DAT
XMPROTOCOL.EXE	XMDEMOS.DAT
XMTER.EXE	XMDEMOS.DAT
XMTRAVEL.EXE	XMTRAVEL.DAT

If a resource file is not found, the example programs run, but some of the display attributes may be incorrect.

4.15.16 UID Files for Example Programs

V1.2 The UID files used by the example programs must be located in either the current directory or your DECW$USER_DEFAULTS directory. If they are not found, the application fails to start. The UID files can be copied from the DECW$EXAMPLES directory.

4.16 DECwindows Extensions to Motif

This section contains information about the Digital extensions to the Motif Toolkit.

4.16.1 DXmCSText Input Method Support

V1.2 X11 R5 input method support is added to the DXmCSText widget. Specify input methods using the vendor shell XmNinputMethod resource. However, to maintain backward compatibility, the existing input method resources DXmNinputMethod and DXmNinputMethodType are still available.

4.16.2 SVN—Horizontal Live Scrolling Not Supported

V1.0 Horizontal live scrolling is not yet supported in the Structured Visual Navigation (SVN) widget.

4.16.3 SVN Horizontal Separator Line

V1.1 When displaying an application that uses the SVN widget on a Sun system running OpenWindows Version 2, the horizontal separator line of the SVN widget is not always displayed due to a problem with the OpenWindows server. The problem does not exist with OpenWindows Version 3.

4.16.4 DXmFormSpaceButtonsEqually Restriction

V1.1 The convenience routine DXmFormSpaceButtonsEqually sizes and spaces all widgets or gadgets equally if they have a subclass of XmLabel or XmLabelGadget. The results are undefined if a widget or gadget is not a subclass of XmLabel or XmLabelGadget.

4.17 Display Server Extensions

This section contains information about the display server extensions.

4.17.1 Display PostScript (DPS) Server Extension—Color Cells

V1.0 You can associate a colormap with a DPS context using the DPSCreateContext routine or the setXgcdrawablecolor operator. You can also specify that DPS allocate its own colors from the specified colormap using the *actual* parameter to DPSCreateContext or the setXgcdrawablecolor operator.

If you have created the colormap using the AllocAll flag, you must not also specify a nonzero value for the actual parameter. DPS always allocates colors as shareable and AllocAll forces all possible colormap entries to be allocated as nonshareable.

4.17.2 X Image Extension

V1.1 The X Image Extension (XIE) allows image display processing using resources on the server side of the X client-server model. XIE eliminates the need to transmit image data repeatedly from the client to the server and also allows data to be transmitted in compressed form, reducing the network load.

VMS DECwindows Motif Version 1.1 includes the XIE client side sharable library (XIE$SHRLIB.EXE) and C language header files. These allow applications to communicate with any X11 server that supports the XIE extension.

An XIE program uses a structure called the XIEImage to describe image data on the client side. This general mechanism describes data that the destination server is incapable of processing. Consult the documentation for the server system for information on what data types and sizes are supported. Unless the documentation specifies different limits, the server is capable of processing unsigned byte (UdpK_DTypeBU), unaligned bit field (UdpK_DTypeVU), and aligned bit field (UdpK_DTypeV) data, with a maximum depth of 8 bits per pixel per component. The XIE client library supports these data types, as well as unsigned word (UdpK_DTypeWU), and a depth of up to 16 bits per pixel per component.

The XIE protocol and programming interface are being standardized within the X Consortium for R6, and programs that use XIE will probably have to be modified. You can use the Image Display Services (IDS) component of DECimage Application Services for VMS as an alternative to the XIE library interface. IDS provides a higher level model of image display and automatically uses XIE when it is available and appropriate.

XIE is documented in the *DECimage Application Services for VMS X Image Extension Programmer's Reference Manual*.

4.17.3 Client Side Extension Library

V1.1

Starting with Version 1.1, Xlib added a client side library that allows VMS clients to issue Shape, XInput, Multibuffer, and shared memory extension requests to servers that provide these features. (For example, the DECwindows X11 display server for OpenVMS VAX does not support the Shape extension while the DECwindows X11 display server for OpenVMS Alpha system does support Shape.) The name of this library is DECW$XEXTLIBSHR.EXE.

You must modify the linking file options for client applications that issue Shape, XInput, Multibuffer, or shared memory extension requests to link to the Xlib extensions shareable image in SYS$LIBRARY:DECW$XEXTLIBSHR.EXE. Add the following line to your linker options file:

```
SYS$LIBRARY:DECW$XEXTLIBSHR/SHARE
```

For more information on Shape, XInput, and Multibuffer extensions, see the following text files in SYS$HELP:

DECW$SHAPE.TXT
DECW$XINPUT.TXT
DECW$MULTIBUFFER.TXT

4.17.4 Shared Memory Extension Support

V1.2

Alpha

On OpenVMS Alpha systems, shared memory extension support provides the capability to share memory XImages. This is a version of the XImage interface where the actual image data is stored in a shared-memory segment. Consequently, the image does not need to be moved through the Xlib interprocess communication channel. For large images, use of this extension can result in dramatic performance increases.

Support for shared memory pixmaps is also provided. Shared memory pixmaps are two-dimensional arrays of pixels in a format specified by the X server, where the image data is stored in the shared memory segment. Through the use of shared memory pixmaps, you can change the contents of these pixmaps without using any Xlib routines.

These routines are included in the client side extension library. See Section 4.17.3 for details on linking this library.

4.17.4.1 How to Use Shared Memory Extension

Code that uses the shared memory extension must include the following header files:

```
# include "DECW$INCLUDE:Xlib.h"
# include "DECW$INCLUDE:shm.h"
# include "DECW$INCLUDE:XShm.h"
```

Any code that uses the shared memory extension should first check that the server provides the extension. In some cases, such as running over the network, the extension does not work.

To check if the shared memory extension is available on your system, call one of the following routines:

```
Status XShmQueryExtension (display)
       Display *display

Status XShmQueryVersion (display, major, minor, pixmaps)
       Display *display;
       int *major, *minor;
       Bool *pixmaps
```

The following table lists each argument and its description.

Argument	Description
display	The current display. If the shared memory extension is used, the return value from either function is True. Otherwise, your program operates using conventional Xlib calls.
major	Major version number of the extension implementation. Returned by XShmQueryVersion.
minor	Minor version number of the extension implementation. Returned by XShmQueryVersion.
pixmaps	True, if shared memory pixmaps.

4.17.4.2 Using Shared Memory XImages

The following sequence shows the process for creating and using shared memory XImages:

1. Create the shared memory XImage structure.
2. Create a shared memory segment to store the image data.
3. Attach the shared memory segment.
4. Inform the server about the shared memory segment.
5. Use the shared memory XImage.

The following sections explain each step in this process:

Step 1—Creating a Shared Memory XImage Structure

To create a shared memory XImage, use the XShmCreateImage routine, which has the following format:

```
XImage *XShmCreateImage (display, visual, depth, format, data,
            shminfo, width, height)
       Display *display;
       Visual *visual;
       unsigned int depth, width, height;
       int format;
       char *data;
       XShmSegmentInfo *shminfo;
```

Most of the arguments are the same as for XCreateImage (See the *X Window System* for a description of the XCreateImage routine.) Note that there are no *offset*, *bitmap_pad*, or *bytes_per_line* arguments. These quantities are set by the server, and your code needs to abide by them. Unless you have already allocated the shared memory segment (see step 2), you pass in NULL for the *data* pointer.

The argument *shminfo* is a pointer to a structure of type XShmSegmentInfo. Allocate one of these structures so that it has a lifetime at least as long as that of the shared memory XImage. There is no need to initialize this structure before the call to XShmCreateImage.

If successful, an XImage structure is returned, which you can use for the subsequent steps.

Step 2—Creating the Shared Memory Segment

Create the shared memory segment after the creation of the XImage because the Ximage returns information that indicates how much memory to allocate.

The following example illustrates how to create the segment:

```
shminfo.shmid = shmget (IPC_PRIVATE,
        image->bytes_per_line * image->height, IPC_CREAT|0777);
```

This example assumes that you called your shared memory XImage structure. A return value of 0 indicates the shared memory allocation has failed. Use the *bytes_per_line* field, not the *width* you used to create the XImage, as they may be different.

Note that the shared memory ID returned by the system is stored in the shminfo structure. The server needs that ID to attach itself to the segment.

Step 3—Attaching the Shared Memory Segment

Attach the shared memory segment to your process as in the following example:

```
shminfo.shmaddr = image->data = shmat (shminfo.shmid, 0, 0);
```

The address returned by shmat is stored in *both* the XImage structure and the shminfo structure.

To finish supplying arguments in the shminfo structure, decide how you want the server to attach to the shared memory segment, and set the *shminfo.readOnly* field as follows:

```
shminfo.readOnly = False;
```

If you set the structure to True, the server cannot write to this segment, and XShmGetImage calls fail.

Note

The shared memory segment routines are provided with DECwindows Motif Version 1.2 for OpenVMS Alpha. Using global sections, these routines emulate the shared memory routines on UNIX systems.

Step 4—Informing the Server About the Shared Memory Segment

Tell the server to attach to your shared memory segment as in the following example:

```
Status XShmAttach (display, shminfo);
```

If successful, a nonzero status is returned, and your XImage is ready for use.

Step 5—Using the Shared Memory XImage

To write a shared memory XImage into an X drawable, use the XShmPutImage routine. The XShmPutImage routine uses the following format:

```
XShmPutImage (display, d, gc, image, src_x, src_y,
                     dest_x, dest_y, width, height, send_event)
       Display *display;
       Drawable d;
       GC gc;
       XImage *image;
       int src_x, src_y, dest_x, dest_y;
       unsigned int width, height;
       Bool send_event;
```

The interface is identical to the XPutImage routine (see the *X Window System*), except for one additional parameter, *send_event*. If this parameter is passed as True, the server generates a completion event when the image write is complete. This allows your program to know when it is safe to begin manipulating the shared memory segment again.

The completion event is of the type XShmCompletionEvent, which is defined as follows:

```
typedef struct {
    inttype;              /* of event */
    unsigned long serial; /* # of last request processed */
    Bool send_event;      /* true if came from a SendEvent request */
    Display *display;     /* Display the event was read from */
    Drawable drawable;    /* drawable of request */
    int major_code;       /* ShmReqCode */
    int minor_code;       /* X_ShmPutImage */
    ShmSeg shmseg;        /* the ShmSeg used in the request */
    unsigned long offset; /* the offset into ShmSeg used */
} XShmCompletionEvent;
```

To determine the event type value that is used at run time, use the XShmGetEventBase routine as in the following example:

```
int CompletionType = XShmGetEventBase (display) + ShmCompletion;
```

Note

If you modify the shared memory segment before the arrival of the completion event, the results may be inconsistent.

To read image data into a shared memory XImage, use the XShmGetImage routine, which uses the following format:

```
Status XShmGetImage (display, d, image, x, y, plane_mask)
       Display *display;
       Drawable d;
       XImage *image;
       int x, y;
       unsigned long plane_mask;
```

The following table lists each argument and its description.

Argument	Description
display	The display of interest.
d	The source drawable.
image	The destination XImage.
x	X-offset within the source drawable.
y	Y-offset within the source drawable.
plane_mask	The planes that are to be read.

To destroy a shared memory XImage, first instruct the server to detach from it, then destroy the segment itself. The following example illustrates how to destroy a shared memory Ximage:

```
XShmDetach (display, shminfo);
XDestroyImage (image);
shmdt (shminfo.shmaddr);
shmctl (shminfo.shmid, IPC_RMID, 0);
```

4.17.4.3 Using Shared Memory Pixmaps

Unlike X images, for which any image format is usable, the shared memory extension supports only a single format for the data stored in a shared memory pixmap (XYPixmap or ZPixmap). This format is independent of the depth of the image and independent of the screen. (For 1-bit pixmaps the format is irrelevant.)

The XShmPixmapFormat routine returns the shared memory pixmap format for the server. The XShmPixmapFormat routine has the following format:

```
int XShmPixmapFormat (display)
       Display *display;
```

Your application can only use shared memory pixmaps in the format returned by the XShmPixmapFormat routine (including bits-per-pixel). To create a shared memory pixmap do the following:

- Create a shared memory segment and shminfo structure exactly the same way as is listed for shared memory XImages steps 1 through 4 (see Section 4.17.4.2). While it is not necessary to create an XImage first (step 1), doing so incurs little overhead and provides an appropriate bytes_per_line value to use.
- Call the XShmCreatePixmap routine, which has the following format:

```
Pixmap XShmCreatePixmap (display, d, data, shminfo, width,
                         height, depth);
       Display *display;
       Drawable d;
       char *data;
       XShmSegmentInfo *shminfo;
       unsigned int width, height, depth;
```

The arguments are the same as for XCreatePixmap (see the *X Window System*) except for two additional parameters, *data* and *shminfo*. The *data* parameter is the pointer to the shared memory segment and is the same as the shminfo.shmaddr field. The *shminfo* parameter is the same as the previous structure.

If successful, a pixmap is returned, which you can manipulate. You can manipulate its contents directly through the shared memory segment. Shared memory pixmaps are destroyed with the XFreePixmap routine, although you should detach and destroy the shared memory segment (see step 4 in Section 4.17.4.2).♦

4.17.5 Using Extension Include Files

V1.2 To ensure that programs that contain extension include files compile properly, add the logical name DECW$INCLUDE to the C include directory search list.

To add the logical name for VAX C, enter the following command:

```
$ DEFINE C$INCLUDE DECW$INCLUDE
```

To add the logical name for DEC C, enter the following command:

```
$ DEFINE DECC$USER_INCLUDE DECW$INCLUDE
```

4.18 Xlib Issues

This section contains information about Xlib.

4.18.1 xlibint.h Is Now a Public Header File

V1.2 To write application interfaces for your server extensions, DECwindows Motif for OpenVMS provides the OpenVMS header file called xlibint.h in the DECW$INCLUDE directory.

4.18.2 Command Procedure Builds .PEN Files

V1.0 To allow Pascal programs to inherit environment files for Xlib and Motif, execute the command procedure SYS$LIBRARY:DECW$PEN_BUILD.COM. This command procedure generates the DECW$XLIBDEF.PEN and DECW$MOTIF.PEN files. The .PEN files compile into Pascal programs faster than the provided .PAS files.

4.18.3 Parameter/Protocol Datasize Mismatches

V1.0 Several Xlib routines accept longword parameters that are not sent in their entirety in the X Protocol message to the server. In each case, the Xlib routine sends out only the least significant 16 bits of the parameter value. This is a constraint of the field size within the X Protocol message. Table 4–11 lists routine names and the longword arguments that are sent only as 16-bit values.

Table 4–11 Routine Names and Arguments Sent as 16-Bit Values

Routine Name	Routine Arguments
XAllocColorCells/ALLOC_COLOR_CELLS	nplanes,npixels
XDrawArc/DRAW_ARC	x,y,width,height, angle1,angle2

(continued on next page)

Table 4–11 (Cont.) Routine Names and Arguments Sent as 16-Bit Values

Routine Name	Routine Arguments
XDrawLine/DRAW_LINE	x1,x2,x3,x4
XDrawPoint/DRAW_POINT	x,y
XDrawRectangle/DRAW_RECTANGLE	x,y,width,height
XDrawString/DRAW_STRING	x,y
XDrawString16/DRAW_STRING16	x,y
XDrawText/DRAW_TEXT	x,y
XDrawText16/DRAW_TEXT16	x,y
XFillArc/FILL_ARC	x,y,width, height,angle1,angle2
XFillRectangle/FILL_RECTANGLE	x,y,width,height

4.18.4 XSelectAsyncEvent and XSelectAsyncInput Routines

V1.1 The XSelectAsyncEvent and XSelectAsyncInput routines allocate memory for the storage of AST delivery information. This memory is freed in the following ways:

- If you close a display (XCloseDisplay), all the AST delivery information associated with all windows on that display is freed.
- If you destroy a window (XDestroyWindow), the AST delivery information for that window is freed.

The AST delivery information for subwindows is not freed by XDestroyWindow.

If you want to turn off AST notification for all event types within a given window and also free the AST delivery information, the client application can call XSelectAsyncEvent or XSelectAsyncInput passing the **event_mask** argument equal to minus one (all bits set) and the **ast_routine** argument equal to zero.

4.18.5 Xlib Internationalization

V1.2 The X Window System Version 11, Release 5 (X11 R5) defines a number of services to support writing internationalized X applications. Internationalization of X is based on the concept of a locale. A locale defines the localized behavior of a program at run time. Locales affect Xlib by:

- Encoding and processing input method text
- Encoding resource files and values
- Encoding and imaging text strings
- Encoding and decoding for interclient text communication

The X Window System defines a general methodology and a set of application programming interfaces (APIs) to standardize programming in X. Standards have not been established for implementing these internationalization features. Currently, the X11 R5 distribution makes two sample implementations of Xlib internationalization support available: Xsi and Ximp. In addition, Digital provides an implementation called Xi18n. You can select which I18N implementation you want. All three implementations provide the same functionality through the same set of public APIs, but their underlying processing differs. These differences are described in the following sections.

4.18.5.1 Vendor Pluggable Layer

V1.2

Digital provides a general mechanism called the vendor pluggable layer, which allows you to choose your own internationalization implementations. Different implementations can be built as standalone shareable libraries and can be selected through the logical DECW$XVENDORLAYER.

If this logical is not defined, the mechanism searches for an internationalization implementation library in the following order:

DECW$XI18NLIBSHR (Xi18n)
DECW$XSILIBSHR (Xsi)
DECW$XIMPLIBSHR (Ximp)

If a shareable library is not found, the default is the Xi18n implementations that are already linked with Xlib.

The following functions act as interfaces between Xlib and the internationalization implementation shareable library:

XDefaultString
XwcFreeStringList
XwcTextListToTextProperty
XmbTextListToTextProperty
XwcTextPropertyToTextList
XmbTextPropertyToTextList
_XrmInitParseInfo
_XlcDefaultLoader

When creating the Xsi or the Ximp shareable library, you need to know the names of the interfaces because they are defined within Xlib. Digital recommends that you rename the functions during compilation by adding the following compilation flags:

```
/define=(-
    "XDefaultString"="_XDefaultString",-
    "XwcFreeStringList"="_XwcFreeStringList",-
    "XwcTextListToTextProperty"="_XwcTextListToTextProperty",-
    "XmbTextListToTextProperty"="_XmbTextListToTextProperty",-
    "XwcTextPropertyToTextList"="_XwcTextPropertyToTextList",-
    "XmbTextPropertyToTextList"="_XmbTextPropertyToTextList",-
    "_XrmInitParseInfo"="__XrmInitParseInfo",-
    "_XlcDefaultLoader"="__XlcDefaultLoader")
```

4.18.5.2 Digital Internationalization Xlib Implementation

V1.2

The Digital implementation (Xi18n) provides enhanced support and a stable internationalization environment. The Digital implementation (Xi18n) provides the following advantages over the Xsi or Ximp environments provided with the X distribution:

- Most European locales and Asian locales are supported by the DECwindows Motif for OpenVMS version of Xlib.
- New locales can easily be supported without changing Xlib. New locales can be added by plugging in a locale-specific shareable library and a corresponding X locale database file for that locale.
- Encoding of input processing can be different from output encodings. The output methods create a default font set and use available font resources installed in the X server. The code conversion is transparent to the user.

- The XIM protocol (protocol between Xlib and input method server) has been proposed to the X Consortium for inclusion in the X11 R6 standard.

4.18.5.3 Locale in OpenVMS Systems

V1.2

DECwindows Motif Version 1.2 for OpenVMS software is dependent on a locale environment that conforms to the ANSI C library. In the DEC C environment, the set of supporting locales does not fully utilize the internationalized Xlib. Therefore, Xlib provides a minimal locale support environment that allows DECwindows Motif for OpenVMS to run on a host environment without locale support.

The locale support environment can use the locale environment provided by DECC$SHR if it is available; otherwise, Xlib uses its internal locale environment. Locales can be set by using the setlocale( ) function if the locale is supported by Xlib.

The following functions and macros are provided in the OpenVMS locale environment:

setlocale
mblen
mbtowc
mbstowcs
wctomb
wcstombs
MB_LEN_MAX

If you write internationalized applications using these functions in the locale environment, do the following:

- For Xlib applications, include <X11/Xlocale.h>. If you include <stdlib.h>, you must do so before <X11/Xlocale.h>.
- For Motif applications, <X11/Xlocale.h> is automatically included.
- Turn on the following compilation flags:

```
/define=(X_LOCALE,X_WCHAR,_WCHAR_T_,XLIB_XPG4_FUNCS)
```

5

Documentation Release Notes

This chapter describes the following information:

- Section 5.1, Documentation Release Notes Specific to the DECwindows Motif Version 1.2–3 for OpenVMS Product
- Section 5.2, DECwindows Motif Version 1.2–3 for OpenVMS Documentation
- Section 5.3, DECwindows Motif Version 1.2–3 for OpenVMS Documentation Corrections
- Section 5.4, OSF/Motif List of Known Problems

5.1 Documentation Release Notes Specific to the DECwindows Motif Version 1.2–3 for OpenVMS Product

V1.2–3 The release notes in this chapter are cumulative from DECwindows Motif Version 1.0 and still apply to Version 1.2–3. The following sections contain documentation release notes that pertain specifically to the DECwindows Motif Version 1.2–3 for OpenVMS release:

- Section 5.2, DECwindows Motif Version 1.2–3 for OpenVMS Documentation
- Section 5.3.1.1, Enhancing Information About the Finish Printing Option
- Section 5.3.2.2, Tear-Off Menus
- Section 5.3.3.1, Global Symbols
- Section 5.3.4.1, GET_CHAR_STRUCT Function
- Section 5.3.6.1, DXmNlayoutDirection Resource Constants
- Section 5.3.7.1, UIL Source Code for the OpenVMS DECburger Application
- Section 5.3.7.2, Help Widget Implementation Code
- Section 5.3.7.3, Help Widget Implementation—Callbacks
- Section 5.3.7.4, Using UIL to Create the Help Widget

5.2 DECwindows Motif Version 1.2–3 for OpenVMS Documentation

V1.2–3 Table 5–1 lists DECwindows Motif for OpenVMS documentation. Information for the documents that is not updated for the DECwindows Motif Version 1.2–3 for OpenVMS release is included in this chapter of the Release Notes.

Table 5–1 DECwindows Motif for OpenVMS Documentation

Title	Software Version
DECwindows Motif Version 1.2–3 for OpenVMS Release Notes	Version 1.2–3
DECwindows Motif Version 1.2–3 for OpenVMS Installation Guide	Version 1.2–3
DECwindows Motif for OpenVMS Applications Guide	Version 1.2[1]
DECwindows Motif for OpenVMS Guide to Non-C Bindings	Version 1.2[1]
Using DECwindows Motif for OpenVMS (Digital Press)	Version 1.1[1]
DECwindows Motif for OpenVMS Quick Reference Card	Version 1.2[1]
Managing DECwindows Motif for OpenVMS Systems	Version 1.2[1]
Overview of DECwindows Motif for OpenVMS Documentation	Version 1.2[1]
DECwindows Extensions to Motif	Version 1.2[2]
DECwindows Companion to the OSF/Motif Style Guide	Version 1.2[2]
DECwindows Motif Guide to Application Programming	Version 1.2[2]
VMS DECwindows Device Driver Manual	Version 1.0[1]
VMS DECwindows Transport Manual	Version 1.0[1]
Porting XUI Applications to Motif	Version 1.0[1]
VMS DECwindows Guide to Xlib (Release 4) Programming: MIT C Binding	Version 1.0[1]
VMS DECwindows Guide to Xlib (Release 4) Programming: VAX Binding	Version 1.0[1]
VMS DECwindows Display PostScript System Programming Supplement	Version 1.0[1]
X Window System (Digital Press)	Release 5 Third Edition
X Window System Toolkit (Digital Press)	Release 4
X and Motif Quick Reference Guide (Digital Press)	Release 5
OSF/Motif Style Guide (Prentice–Hall)	Release 1.2
OSF/Motif Programmer's Guide (Prentice–Hall)	Release 1.2
OSF/Motif Programmer's Reference (Prentice–Hall)	Release 1.2

[1] Updates to these documents, if any, are noted in this chapter of the Release Notes.

[2] The recently updated DEC OSF/1 version of this document is provided with the DECwindows Motif Version 1.2–3 for OpenVMS documentation kit. The information is applicable to the DEC OSF/1 and DECwindows Motif for OpenVMS releases.

5.3 DECwindows Motif Version 1.2–3 for OpenVMS Documentation Corrections

This section contains corrections and enhancements to existing DECwindows Motif Version 1.2–3 for OpenVMS documentation.

5.3.1 *DECwindows Motif for OpenVMS Applications Guide*

This section contains documentation corrections to the *DECwindows Motif for OpenVMS Applications Guide* manual.

5.3.1.1 Enhancing Information About the Finish Printing Option

V1.2–3 The section called "Printing Information" in the chapter on DECterm provides information about the Print menu. To further clarify the information in the *Finish Printing* section, note the following:

Selecting the Finish Printing option on the Print menu closes the print job and toggles Auto Print mode back to Normal Print mode.

5.3.2 *Using DECwindows Motif for OpenVMS*

This section contains documentation corrections and enhancements to the *Using DECwindows Motif for OpenVMS* manual.

5.3.2.1 Using the Drag-and-Drop Feature

V1.2 The drag-and-drop feature lets you move or copy screen objects. For example, you can move text from buttons and paste it elsewhere.

To drag and drop text into a new location:

1. Select the text to be copied or moved with MB1.
2. To move the text, press and hold MB2; to copy the text, press and hold Ctrl/MB2.

   A move or copy icon appears.
3. Drag the icon to the location where you want to drop the text and release MB2.

   If the object is highlighted as you drag the icon across it, you can drop the text into that location.

Drag-and-drop is provided primarily for programmers to incorporate the feature into their applications.

The DECwindows Motif Version 1.2 for OpenVMS applications support the drag-and-drop feature, with the exception of Notepad. DECwindows Mail supports drag-and-drop in all windows except the main message area, where DECwindows Mail has its own drag-and-drop feature; you can use MB2 to move messages around with the SVN interface.

5.3.2.2 Tear-Off Menus

V1.2–3 The DECwindows Mail application supports tear-off menus.

V1.2 The DECwindows Motif applications allow you to tear off pull-down and popup menus. Tear-off menus let you keep frequently used menus displayed without repeatedly pulling them down or popping them up.

To tear off a menu:

1. Display a pull-down or popup menu.

   If the menu is a tear-off menu, a dotted line is displayed at the top of the menu.
2. Click on the dotted line with MB1.

   The menu remains active until it is closed or until the parent application is closed.

To close a tear-off menu:

1. Click on the Window menu button in the tear-off menu.
2. Choose the Close menu item.

The following applications do not support tear-off menus:

- CDA Viewer
- DECwindows Mail
- Notepad
- Print Screen

5.3.2.3 Adding Target Screen Options to Application Menu Items

V1.2

The example “Adding Target Screen Options to Application Menu Items” in *Using DECwindows Motif for OpenVMS* is incorrect. To correct the problem, remove the first occurrence of the following line:

```
$ select_qualifiers:
```

5.3.2.4 Changing the Startup Environment

V1.2

The example “Changing Your Logo” is incorrect. To correct the problem, change the following code example in step one:

```
$ COPY SYS$COMMON:[SYSMGR]DECW$PRIVATE_APPS_SETUP.TEMPLATE -
_$ SYS$SPECIFIC:[SYSMANAGER]DECW$PRIVATE_APPS_SETUP.COM/LOG
```

The code example should read as follows:

```
$ COPY SYS$COMMON:[SYSMGR]DECW$PRIVATE_APPS_SETUP.TEMPLATE -
_$ SYS$SPECIFIC:[SYSMGR]DECW$PRIVATE_APPS_SETUP.COM/LOG
```

5.3.2.5 Enhancing Startup Performance

V1.1

Extensive SYLOGIN.COM or LOGIN.COM command procedures slow down application startup. Many of the operations performed in a SYLOGIN.COM or LOGIN.COM are meaningless for DECwindows application startup. Therefore, the SYLOGIN.COM and LOGIN.COM files should be conditionalized for DECwindows application startup performance. When starting a DECwindows application, a minimum of SYLOGIN.COM and LOGIN.COM commands should be executed.

Typically, the commands that should be executed are the redefinition of DECW$USER_DEFAULTS (if present), and other logical name definitions if the user will be referencing them from within the context of a DECwindows application. The following code segment can be inserted into SYLOGIN.COM and LOGIN.COM immediately following the commands necessary for DECwindows:

```
$ mode = f$mode()
$ tt_devname = f$trnlnm("TT")
$ session_mgr_login = (mode .eqs. "INTERACTIVE") .and.  -
      (f$locate("WSA",tt_devname) .ne. f$len(tt_devname))
$ session_detached_process = (mode .eqs. "INTERACTIVE") .and. -
      (f$locate("MBA",tt_devname) .ne. f$len(tt_devname))
$ if session_mgr_login .or. session_detached_process then exit
```

Applications continue to run even if these lines are not added to the SYLOGIN.COM and LOGIN.COM files.

5.3.2.6 Enhancing Hold-Screen Response Time

V1.1

If the Hold Screen key response time is too slow, add the following lines to your DECW$TERMINAL_DEFAULT.DAT file:

```
DECW$TERMINAL.main.terminal.syncFrequency: 1
DECW$TERMINAL.main.terminal.batchScrollCount: 1
```

Using this resource can affect the performance of the DECterm window. The actual impact on performance varies from site to site. You can trade off scrolling speed to hold-screen response time. A faster hold-screen response results in a slower scrolling speed. The default values for these resources are 10 and 0, respectively.

5.3.3 *Managing DECwindows Motif for OpenVMS Systems*

This section contains documentation enhancements and corrections to the *Managing DECwindows Motif for OpenVMS Systems* manual.

5.3.3.1 Global Symbols

V1.2–3

The following additional global symbols are available in the DECwindows Motif Version 1.2–3 for OpenVMS product:

- DECW$CONSOLE_SELECTION

  Specifies how to display operator-messages options.

- DECW$CONSOLE_GEOMETRY

  Specifies the value of the -geometry option in the DECW$MESSAGEPANEL.EXE command line.

For information about using these symbols, refer to Section 1.8 in these Release Notes. Refer to the chapter “Using DECwindows” in *Managing DECwindows Motif for OpenVMS Systems* for the complete list of global symbols.

5.3.3.2 Security Options

V1.2

In the Session Manager Security Options dialog box, place the node name within quotation marks if the name contains any of the following:

- Reserved characters: space, tab, comma (,) or double quotation mark (")
- Double colon (::)
- A colon (:) as the final character in the node name

Session Manager automatically adds quotation marks to the node name if they are needed, unless the node name begins with a double quotation mark. If the node name begins with a double quotation mark, Session Manager assumes that the user has already quoted the node name and does not change it.

Within a quoted string, a double quotation mark should be replaced by two double quotation marks (" "). For example, the quoted string `"DEC:.zko."my node""` should be changed to the following:

```
("DEC:.zko.""my node""").
```

5.3.3.3 Displaying a Customized Logo

V1.1

By default, if there is no DECwindows Motif license registered for the SYSTEM account, DECwindows does not display customized login logos. This is a problem on systems with DECwindows Motif personal-use licenses that do not include SYSTEM on the list of authorized DECwindows users.

To display a customized logo without a DECwindows Motif license for SYSTEM, add the following definition to the SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM file:

```
$ DECW$LOGINLOGOSUB == "TRUE"
```

Note

If the SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM file does not exist, copy it from the file SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.TEMPLATE.

After editing the setup file, restart DECwindows Motif using the following command:

```
$ @SYS$MANAGER:DECW$STARTUP RESTART
```

DECwindows Motif login starts the logo process as a subprocess instead of as a detached process. The license check sees that the logo process is a child of the login process and that the X connection is opened.

5.3.3.4 Enabling and Disabling Access Control

V1.0

DECwindows Motif does not enable access control by default. Instead, it uses the access control set by the server. The DECwindows X11 display server enables access control at startup time.

To force the DECwindows Session Manager to enable or disable access control explicitly at login time, you can define one of the following logical names:

```
$ DEFINE/SYSTEM/EXECUTIVE DECW$LOGIN_ACCESS_CONTROL ENABLE
```

```
$ DEFINE/SYSTEM/EXECUTIVE DECW$LOGIN_ACCESS_CONTROL DISABLE
```

If the logical name is not defined, or if it is defined to some other value, such as “SERVER”, DECwindows login neither enables nor disables access control.

In most cases, it should not be necessary to define the logical name.

5.3.4 *DECwindows Motif for OpenVMS Guide to Non-C Bindings*

This section contains documentation enhancements and corrections to the *DECwindows Motif for OpenVMS Guide to Non-C Bindings*.

5.3.4.1 GET_CHAR_STRUCT Function

V1.2–3

The access related to the char_struct argument for the XLIB GET_CHAR_STRUCT function is incorrectly documented. The correct access is *write*.

5.3.5 *VMS DECwindows Guide to Xlib (Release 4) Programming: VAX Binding*

V1.2 Example 1-1 in the *VMS DECwindows Guide to Xlib (Release 4) Programming: VAX Binding* is incorrect. To correct the problem, change the following two lines in the section "Create the WINDOW_1 window":

```
WINDOW_1X = (X$WIDTH_OF_SCREEN(DPY) - WINDOW_1W) / 2
WINDOW_1Y = (X$HEIGHT_OF_SCREEN(DPY) - WINDOW_1H) / 2
```

The example should read as follows:

```
WINDOW_1X = (X$WIDTH_OF_SCREEN(SCREEN) - WINDOW_1W) / 2
WINDOW_1Y = (X$HEIGHT_OF_SCREEN(SCREEN) - WINDOW_1H) / 2
```

Example 3-1 in the *VMS DECwindows Guide to Xlib (Release 4) Programming: VAX Binding* is incorrect. To correct the problem, change the following two lines:

```
WINDOW_1X = (X$DISPLAY_WIDTH_OF_SCREEN(SCREEN) - WINDOW_1W) / 2
WINDOW_1Y = (X$DISPLAY_HEIGHT_OF_SCREEN(SCREEN) - WINDOW_1H) / 2
```

The example should read as follows:

```
WINDOW_1X = (X$DISPLAY_WIDTH_OF_SCREEN(DPY) - WINDOW_1W) / 2
WINDOW_1Y = (X$DISPLAY_HEIGHT_OF_SCREEN(DPY) - WINDOW_1H) / 2
```

5.3.6 *DECwindows Extensions to Motif*

This section lists documentation corrections and enhancements for the *DECwindows Extensions to Motif* document.

5.3.6.1 DXmNlayoutDirection Resource Constants

V1.2–3 The section on *DXmNlayoutDirection Resource* in Chapter 2 lists the following constants:

- DXmLAYOUT_LEFT_DOWN
- DXmLAYOUT_LEFT_UP
- DXmLAYOUT_RIGHT_DOWN
- DXmLAYOUT_RIGHT_UP

Note that the following constants are not currently available for the DXmNlayoutDirection resource:

- DXmLAYOUT_LEFT_UP
- DXmLAYOUT_RIGHT_UP

The introduction to Table 2–1 is incorrect. Table 2–1 describes the effect of the constants DXmLAYOUT_LEFT_DOWN and DXmLAYOUT_RIGHT_DOWN on the functions.

V1.2 In Section 2.2 of the *DECwindows Extensions to Motif* manual, the description of DXmChildren indicates that the routine can be used to learn the length of widget_list.

The documentation is incorrect. The sentence should read, "You can use the DXmNumChildren routine to learn the length of the widget list returned by DXmChildren."

5.3.7 *DECwindows Motif Guide to Application Programming*

This section lists documentation corrections and enhancements for the *DECwindows Motif Guide to Application Programming* manual.

5.3.7.1 UIL Source Code for the OpenVMS DECburger Application

V1.2–3 The introduction to the section in Chapter 4 called "Creating the Help Widget with UIL", which introduces Example 4–6, "UIL Help Widget Implementation," does not clearly state that the complete UIL source code for the OpenVMS DECburger application is included in DECW$EXAMPLES on OpenVMS systems.

5.3.7.2 Help Widget Implementation Code

V1.2–3 In the title of Example 4–7, "Help Widget Implementation—C Language Module", DECBURGER.C should be added to the title as follows:

"Help Widget Implementation—C Language Module (DECburger.C)"

5.3.7.3 Help Widget Implementation—Callbacks

V1.2–3 The title of Example 4-8 should read as follows:

"Help Widget Implementation—Callbacks (DECburger.C continued)"

5.3.7.4 Using UIL to Create the Help Widget

V1.2–3 The title of Section 4.10, "Using the Toolkit Help Widget Creation Routine," does not accurately reflect the contents of this section. The title should read, "Using UIL to Create the Help Widget".

The introductory paragraph to Example 4–9 in the section called "Using the Toolkit Help Widget Creation Routine" is incomplete. The information should state the following:

"The code in Example 4–9 is included in DECburger.C but is commented and will not be compiled when the example program is built."

The implementation in this example is not complete; it requires some sections from Examples 4-7 and 4-8.

To summarize, the documentation should state the following information:

- Examples 4-6 and 4-7 are UIL code (DECburger.UIL(D)) that, when combined, create the DECburger.EXE.
- Example 4-8 is C code for DECburger.EXE.
- Example 4-9 is C code for a Toolkit example.

5.3.8 *Overview of DECwindows Motif for OpenVMS Documentation*

V1.2 Table 2-2 in the *Overview of DECwindows Motif for OpenVMS Documentation* Version 1.2 indicates that the *DECwindows Motif Version 1.2 for OpenVMS Installation Guide* is included in the User Kit. This is incorrect. The installation guide is included with the media kit.

5.4 OSF/Motif List of Known Problems

Included in the DECwindows Motif for OpenVMS software is an OSF file that contains the OSF/Motif known problems. During the installation this file is copied from the kit to SYS$HELP:DECW$MOTIF_OSF_BUGLIST_V12.TXT.

A
OSF/Motif Release 1.2 Release Notes

The release notes in this document are based on Chapter 3 of the OSF/Motif Release Notes for Release 1.2 and include a few other notes relevant to programmers developing OSF/Motif applications. Most of the notes describe changes made for OSF/Motif Release 1.2. The first two notes discuss performance improvements and information about backward compatibility.

These Release Notes support the OSF/Motif software provided with the DECwindows Motif for OpenVMS software, DEC OSF/1 software, the eXcursion Software Development Kit, and ULTRIX Worksystem Software.

For the OSF/Motif Release 1.2.3 update, approximately 425 problems were resolved.

A.1 Performance Improvements

The Open Software Foundation set as its goal for OSF/Motif Release 1.2 to improve performance where possible and, at a minimum, to not allow it to degrade below the performance of OSF/Motif Release 1.1.

Performance testing was done in three areas:

- Aspects obvious to end users
- Data space usage
- Memory leakage

The performance of user-perceptible events such as posting and unposting dialog boxes and pop-up menus are comparable or better than the performance for OSF/Motif Release 1.1.4. A significant improvement was made for scrolling inside a Scrolled Text region that contains a large amount of test.

Data space usage has improved throughout the OSF/Motif Toolkit, particularly for the Text widget. In some cases, data space usage has been reduced by as much as 40%. Also, the memory used for the text in a Text widget is now correctly reduced when that text is replaced by a smaller amount of text.

Memory leakage has been reduced to a minimum for multiple creates and destroys of all Toolkit widgets. Although the Motif tests showed small memory leaks, the OSF believes that this amount of memory is required as part of the startup overhead and is not a true memory leak.

The OSF did find some memory leaks that they plan to fix in a future release. Three widgets—File Selection Box, Command, and Drawn Button - leak approximately 500 bytes of memory per instance.

A.2 Backward Compatibility

The OSF tested OSF/Motif Release 1.2 for both link-time compatibility as well as visual and behavioral compatibility.

A.2.1 Visual and Behavioral Compatibility

The OSF ran automated tests that compared current visuals with those recorded using OSF/Motif Release 1.1.4 libraries. Once all differences between the Release 1.2 and Release 1.1.4 versions were accounted for, the visuals were rerecorded using Release 1.2 visuals. These new recorded visuals were used in all subsequent tests.

The OSF believes that Motif Release 1.2 is visually and behaviorally compatible with Release 1.1.4. However, they have made extensive improvements to the Traversal and Geometry Management algorithms that result in some differences between those versions. These differences reflect efforts to fix defects in the earlier release.

For example, one such modification involves the new policy in which an initial size set for a manager widget in an application is now honored by the Toolkit. In OSF/Motif Release 1.1, applications set the initial size for a manager widget, but did not, in fact, use that size. In Release 1.2, Motif now uses that size setting and the initial layout is changed accordingly.

A.3 Changes and New Features for OSF/Motif Release 1.2

This section summarizes changes and new features that the OSF has made to OSF/Motif Release 1.2. Detailed information about these modifications is contained in the Motif reference pages and the Motif Release 1.2 revisions of following books:

- *OSF/Motif Style Guide*
- *OSF/Motif Programmer's Guide*

The following sections discuss the OSF/Motif Release 1.2 enhancements.

A.3.1 General Toolkit Changes

This section discusses the changes made to the overall OSF/Motif Release 1.2 Toolkit.

A.3.1.1 Include File Changes

The following header files that were in OSF/Motif Release 1.1 are obsolete in Release 1.2:

ExtObject.h
Traversal.h
VaSimple.h
VendorE.h
VendorEP.h

A new public header file, `XmAll.h`, has been added to OSF/Motif Release 1.2. This header file includes all the documented header files.

A.3.2 Change in XT Translations

As a result of fixing a problem in the XT translation code (Patch 25 for X11 R5), the translations in Xt are now handled strictly and no longer accept any possible match as they did before. This change has caused a change in the behavior of the QATS and Motif VTS test suites, which now make incorrect assumptions for certain keyboards, such as those that have the arrow keys defined in the keypad. Other Motif applications might be affected as well.

You can avoid this problem by creating a file to redefine the bindings for the keys in question and then passing that file to the xmodmap utility.

The following example changes the bindings for the keypad keys that match the arrow keys so that only the arrow keys are produced, not the keypad numbers. The new definitions allow the application to use modifiers with the arrow keys.

```
!
! Always force:
!   KP_2 = Down
!   KP_8 = Up
!   KP_4 = Left
!   KP_6 = Right
keycode 120 = Down
keycode 76  = Up
keycode 98  = Left
keycode 100 = Right
```

A.3.3 ANSI C Compliance

All references to caddr_t have been changed to XtPointer. This change affects all callback routines and any other routines that reference caddr_t. The OSF made this change so that OSF/Motif Release 1.2 would be compliant with the ANSI C specification.

A.3.4 Display and Screen Specific Data

Motif now has an XmDisplay object that supports per-display data and resources. An XmScreen object has been added that supports per-screen data and resources.

A.3.5 Drag and Drop

OSF/Motif Release 1.2 supports the drag and drop metaphor for data interchange. The drag-and-drop specification has been fully implemented. See the Release 1.2 version of the OSF/Motif Programmer's Guide for information on the drag and drop interface.

Note

If you want to use Btn2 to have pop-up menus pop up, drag and drop will not function properly. You need to disable drag and drop in such instances.

A.3.6 Tear-Off Menus

With tear-off menus, the user can retain menus on the display area for subsequent selections. Each tearable menupane is a tear-off button. When the mouse drag button is pressed on the tear-off button, the pane tears off and can be dragged and then placed by releasing the mouse drag button. The window manager surrounds the tear-off menupane with a menu button and a title. Shifting focus to a torn-off menu's windowpane follows the standard window manager policy.

Tear-off behavior is enabled by setting the `XmNtearOffModel` resource to `XmTEAR_OFF_ENABLED`. (The default is `XmTEAR_OFF_DISABLED`).

Note that there is no resource converter preregistered by `XmNtearOffModel`. To allow the tear-off function to be enabled through the resource database, an application must register its own resource converter for the `XmNtearOffModel` resource using the `XmRepTypeInstallTearOffModelConverter` function.

The converter is not automatically installed because many applications use map or cascading callbacks to dynamically set the sensitivity of items within their menus. However, if a tear-off menu is mapped, the sensitivity of its menu items must be changed immediately to reflect changes in other application states. Existing applications are unlikely to change menu item sensitivity in this manner. Thus, allowing their menus to be torn off could result in operations being enabled at unexpected times. If a user activates one of these menu items, the application can crash or the persistent data can be corrupted.

A.3.7 Insensitive Visuals

Motif provides visual indications to show whether a component can respond to input from users. Labels and buttons have had this behavior in previous Motif releases. In OSF/Motif Release 1.2, this behavior has been extended to the following widgets:

```
XmArrowButton
XmList
XmScrollBar
XmText
XmTextField
```

A.3.8 Other Visual Changes

OSF/Motif Release 1.2 has made other visual changes as follows:

- Several pixels have changed in the three-dimensional beveled look.
- Minor changes have been made to the color generation routines. In particular, the new `XmScreen` object contains resources that allow for tailoring the generation of default colors. Some of the default values for thresholds have been adjusted to produce more contrast on color monitors.
- Motif now adds a location cursor to surround all items in a List whenever a List widget has the focus and the current keyboard item is not visible.
- There are minor layout differences because of fixes in geometry management.

A.3.9 Titles for Frames

In OSF/Motif Release 1.2, you can specify a Title widget in a Frame widget. The release has added the following new constraint resources for specifying the position and alignment of the title in the Frame:

```
XmNchildHorizontalAlignment
XmNchildHorizontalSpacing
XmNchildType
XmNchildVerticalAlignment
```

A.3.10 Audible Warning

The VendorShell has a new resource, `XmNaudibleWarning`, that can specify whether an audible cue should accompany a warning message. Text widgets determine the value for this resource from the value of `XmNaudibleWarning`.

A.3.11 Color Enhancements

The following three resources have been added to the `XmScreen` widget so that users can specify the default background color and thresholds for shadow calculation:

```
XmNlightThreshold
XmNdarkThreshold
XmNforegroundThreshold
```

Motif has added the `XmChangeColor` function that changes the background and other colors for a specified widget.

A.3.12 Baseline Alignment

Motif has added two functions for baseline alignment. The `XmWidgetGetBaselines` function determines the position of the widget's text baseline. The `XmWidgetGetDisplayRect` function determines the size and position of the bounding box for the widget's character string.

A.3.13 Expanded Traversal Set

In OSF/Motif Release 1.2, you can use more widgets to support traversal using the keyboard. For example, inside a tab group, users can now use the arrow keys to traverse to all control descendants that are not contained within a nested tab group and that are eligible to receive focus, even if the controls are not direct children of the tab group.

A.3.14 Two-Dimensional Menu Traversal

With OSF/Motif Release 1.2, the left, right, up, and down traversal arrows now navigate within a menupane. The up and down arrow keys wrap between columns. The right and left arrow keys post the previous or next menupane when they are pressed in the rightmost and leftmost column of the menupane, respectively.

A.3.15 Input Focus

OSF/Motif Release 1.2 has added the `XmNinitialFocus` resource to the Manager class. This resource specifies the first widget to receive input focus. This resource can only specify a widget; it is ignored for all pop-up menus, menubars, option menus, and pull-down menus.

A.3.16 Traversal Access Functions

OSF/Motif Release 1.2 has added the following new functions to support better interaction with keyboard traversal:

```
XmGetFocusWidget
XmGetTabGroup
XmGetVisibility
XmIsTraversable
XmIsVisible
```

The `XmTrackingLocate` function has been modified to do the following:

- Field all events, not just a button press.
- Return on any keystroke or button press.
- Be called for nonsensitive widgets.

In addition, the `XmTrackingEvent` function has been added. This function is similar to `XmTrackingLocate`, except that it returns a pointer to the X event.

A.3.17 Virtual Keys

OSF/Motif Release 1.2 has added the `XmTranslateKey` function that allows applications to override the default `XtKeyProc` to handle Motif virtual keys.

Motif defines two new virtual keysyms:

```
osfPageLeft
osfPageRight
```

You must have the X11 Release 5 `XKeysymDB` installed in `/usr/lib/X11` to use these new virtual keys. Otherwise you get a warning message on application startup. Specify the following information in the `XKeysymDB` file:

osfPageLeft	:1004FF40
osfPageUp	:1004FF41
osfPageDown	:1004FF42
osfPageRight	:1004FF43

To comply with the OSF/Motif Style Guide, the default binding for osfMenu has been changed from `<key>F4` to `Shift<key>F10`.

In X11 Release 5, the HP keysyms in the `XKeysymDB` file have the prefix hp. This prefix is not reflected in the HP bindings file in the `/bindings` directory. If you are using an X11 Release 5 `XKeysymDB` file, you might see warning messages at application startup. To eliminate these warning messages, add the `hp` prefix as follows to the appropriate lines in the file:

```
osfDelete :               <key>hpDeleteChar
osfInsert :               <key>hpInsertChar
osfPrimaryPaste :         <key>hpInsertLine
osfQuickPaste :           <key>hpDeleteLine
```

OSF/Motif Release 1.2 has a new client, `xmbind`, that sets up the virtual bindings for use by Motif applications. Since virtual binding is automatically set up at Motif Window Manager (MWM) startup, you only need to use `xmbind` if MWM is not used or if you need to change the virtual bindings without restarting MWM.

Virtual bindings can now be specified by individual vendors. If there is no .motifbind file in the home directory, you can use the xmbind.alias file to provide a mapping from the server vendor name to the bindings file. You can set up user vendor bindings as well as system-wide vendor bindings.

A.3.18 Resource Management

OSF/Motif Release 1.2 has the following new functions for managing representation types:

```
XmRepTypeRegister
XmRepTypeAddReverse
XmRepTypeValidValue
XmRepTypeGetRegistered
XmRepTypeGetId
XmRepTypeGetNameList
XmRepTypeGetRecord
```

These functions are useful for developers who want to define new resource converters that use an enumerated set of values.

A.3.19 Changes for CUA and Windows Compliance

In OSF/Motif Release 1.2, pressing the Return key or using the key bound to osfActivate (usually the Enter key on the numeric keypad) no longer activates a button that is outside a menu. For example, pressing such a key no longer pops up an OptionMenu or activates a ToggleButton in a dialog box.

If your application has a default button associated with an XmBulletinBoard, pressing Return (except in a multiline XmText), Ctrl/Return, or the key bound to osfActivate while the focus is in the XmBulletinBoard now activates the default button.

A.4 Changes and Enhancements to Specific Widgets

This section summarizes the changes to specific widgets that were made in OSF/Motif Release 1.2.

A.4.1 XmClipboard

OSF made several corrections to the XmClipboard function parameters. These changes are binary compatible with earlier releases of Motif. However, in some instances, you might see warning messages when you recompile your applications. The modifications involved changing (char *) to XtPointer, int to long, and (int *) to (long *). The related functions affected by these modifications are:

- XmClipboardCopy
- XmClipboardCopyByName
- XmClipboardInquireCount
- XmClipboardInquireFormat
- XmClipboardRegisterFormat
- XmClipboardRetrieve
- XmClipboardStartCopy
- XmClipboardWithdrawFormat

A.4.2 XmCommand

In OSF/Motif Release 1.2, a correction was made to `XmCommandGetChild` so that it now accepts `XmDIALOG_WORK_AREA` as a value for the child argument.

A.4.3 XmList

To enchance its capabilities for managing lists, OSF/Motif Release 1.2 includes the following new functions:

- `XmListAddItemsUnselected`
- `XmListDeletePositions`
- `XmListGetKbdItemPos`
- `XmListIsPosSelected`
- `XmListPosToBounds`
- `XmListReplaceItemsUnselected`
- `XmListReplaceItemsPosUnselected`
- `XmListReplacePositions`
- `XmListSetKbdItemPos`
- `XmListUpdateSelectedList`
- `XmListYToPos`

The XmList widget includes a new translation:

- `<Copy>`

  Copies the selection to the clipboard.

`XmList` includes a new action:

- `ListScrollCursorVertically()`

  Scrolls the cursor vertically based on an input percentage or a `y` position.

Note

This action was mistakenly named `ListScrollCursorVisible` in OSF/Motif Release 1.2. The name will be corrected in a later release of Motif.

The `XmNvisibleItemCount` resource has been modified so that the default value is dynamic, and is based on the item count and the height.

In OSF/Motif Release 1.2, if the `selectedItems` and `selectedItemCount` resources for a list are set in a resource file, the location cursor appears over the last item in the `selectedItems` list, not the first selected item.

A.4.4 XmMessageBox

In OSF/Motif Release 1.2, `MessageBox` supports the addition of one `MenuBar`, one work area, and multiple `PushButton` children.

A new dialog type, `XmDIALOG_TEMPLATE`, creates a `MessageBox` that contains only a `Separator`. The application provides additional children.

XmCreateTemplateDialog creates an XmDIALOG_TEMPLATE XmMessageBox inside a DialogShell.

A.4.5 XmRowColumn and Menus

OSF/Motif Release 1.2 adds a new resource, XmNentryVerticalAlignment, that specifies the vertical alignment style.

Another resource, XmNunpostBehavior has been added to the XmScreen object. This resource can be set to enable external button events to be replayed after a menu is unposted.

A.4.6 XmScrollBar

In OSF/Motif Release 1.2, XmScrollBar includes a new translation:

- <Cancel>

  Cancels the current slider drag.

A.4.7 XmScrolledWindow

OSF/Motif Release 1.2 adds the function, XmScrollVisible, that scrolls an automatic scrolled window to make a partially or completely obscured widget visible.

Another resource, XmNtraverseObscuredCallback, has been added that specifies a list of callbacks that are called when a traversal event is requested to a nonvisible widget. A new callback structure, XmTraverseObscuredCallbackStruct, has been added to support this callback.

A.4.8 XmSelectionBox, XmFileSelectionBox

In OSF/Motif Release 1.2, the XmSelectionBox and XmFileSelectionBox widgets support the addition of MenuBar and PushButton children, as well as a work area child.

A new resource, XmNchildPlacement, controls the location of the work area child.

The value, XmDIALOG_TEMPLATE, has been added to the XmNdialogType resource.

By default, XmSelectionBoxDialog and its subclasses use XmTextField instead of XmText. You can revert to the earlier behavior by defining USE_TEXT_IN_DIALOGS when your application builds XmSelectionBox or any of its subclasses.

A.4.9 XmText

OSF/Motif Release 1.2 has added two functions to XmText for making update changes to the widget: XmTextDisableRedisplay and XmTextEnableRedisplay.

Two other functions facilitate string manipulation: XmTextFindString and XmTextGetSubstring.

In Release 1.2, the destination cursor now follows the insert cursor and is no longer independently drawn.

XmText includes three new translations:

- <Backspace>

  Deletes any non-null primary selection.

- <Delete>

  Deletes any non-null primary selection.

- <LeaveWindow>

Continues a selection action by scrolling after a time delay.

XmText includes two new actions:

- scroll-cursor-vertically()

  Scrolls the cursor vertically based on a y position.

- toggle-overstrike()

  Switches between insert and overstrike modes.

Note

There is a potential problem in both XmText and XmTextField with rendering strings in fonts or font sets that contain characters whose ascenders can rise above the font ascent. If the text containing these characters is highlighted, any overlapping descenders in the previous line may be overwritten by the ascenders in the succeeding line.

A.4.10 XmTextField

OSF/Motif Release 1.2, the XmTextField widget has a new resource, XmNfocusCallback, that specifies the callbacks to be called when the widget accepts input focus.

Another new function, XmTextFieldGetSubstring, gets a substring by length from a widget.

In Release 1.2, the destination cursor now follows the insert cursor and is no longer independently drawn.

XmTextField includes two new translations:

- <Backspace>

  Deletes any non-null primary selection.

- <Delete>

  Deletes any non-null primary selection.

A.4.11 XmToggleButton, XmToggleButtonGadget

In OSF/Motif Release 1.2, setting XmNfillOnSelect to be true when XmNindicatorOn is false now causes the background of a set XmToggleButton to be filled with XmNselectColor.

The default value for XmNfillOnSelect is dynamic so that it matches the state of XmNindicatorOn.

A.5 Motif Window Manager Enhancements

This section highlights the enhancements to the Motif Window Manager (MWM).

A.5.1 Changes to MWM

OSF/Motif Release 1.2 incorporates the following enhancements to MWM:

- An internationalized .mwmrc file
- Internationalized dialog messages
- A built-in default root menu

- Support for the <Return> continuation character (\) in the .mwmrc file
- Search capability with XBMLANGPATH for bitmap files
- Support for pop-down and replay event behavior in mwm menus
- Documentation of the widget names used by mwm
- Support for scrolled window traversal to scrolled-off children in the icon box
- Support for treating the <Alt> and <Meta> key modifiers as two distinct modifiers
- Support for the SHAPE nonrectangular window extension

A.5.2 New and Enhanced MWM Resources

The OSF/Motif Release 1.2 window manager includes the following new or enhanced resources:

- `feedbackGeometry`

  Sets the position of the move/resize feedback window. The default position is the center of the screen.

- `frameBorderWidth`

  Now bases its default value on the size and resolution of the screen.

- `iconPlacement`

  Now takes the addition value, `tight`, which specifies automatic icon placement with no gaps between icons.

- `maximumClientSize`

  Can now take the values `vertical` and `horizontal`.

- `moveOpaque`

  Controls whether an image of the window or just an outline of the window is moved.

- `resizeBorderWidth`

  Now bases its default value on the size and resolution of the screen.

- `usePPosition`

  Uses the values of `on`, `off`, or `nonzero` to control whether program-specified positions are used.

A.5.3 New and Enhanced MWM Functions

OSF/Motif Release 1.2 has the following new and enhanced MWM functions:

- `f.lower`

  Includes a `within` argument to move the window within the application stacking order, but retains the parent window below the children rule. The function also includes a `freeFamily` argument to move the window absolutely without regard to its local family stack. Both modifiers move the window within the local family stack, but do not move the family stack.

- `f.minimize`

  Can now be used from an icon in an icon box.

- `f.raise`

Includes a within argument to move the window within the application stacking order, but retains the parent window below the children rule. The function also includes a freeFamily argument to move the window absolutely without regard to its local family stack. Both modifiers move the window within the local family stack, but do not move the family stack.

- f.raise_lower

  Includes a within argument to move the window within the application stacking order, but retains the parent window below the children rule. The function also includes a freeFamily argument to move the window absolutely without regard to its local family stack. Both modifiers move the window within the local family stack, but do not move the family stack.

- f.restore

  Restores a window to its previous state. Double clicking on a root icon is bound to this function rather than to f.normalize.

- f.restore_and_raise

  Restores a window to its previous state and raises it to the top of the window stack. Double clicking on an icon in an icon box is bound to this function rather than to f.normalize.

- f.screen

  Traverses to the screen specified by arg. Legal values for arg are: next, prev, last, or a specific screen number.

A.5.4 New MWM Action

OSF/Motif Release 1.2 has one new MWM action:

- <Alt> <Esc>

  This key combination behaves similarly to f.next_key, except that the window is always raised, regardless of the value of focusAutoRaise.

A.6 Changes to the User Interface Language

OSF/Motif Release 1.2 has the following changes in the User Interface Language (UIL):

- There is a new command flag, -s, that enables the use of setlocale and the creation of localized Compound Strings.

Note

There is a serious problem in parsing double quoted strings with the -s flag. If you need to use this flag for parsing double quoted strings, you must obtain a patch from the OSF.

- New UIL syntax now supports font sets and font tables.
- New UIL syntax now supports wide character strings.
- Support has been added for using widget references as callback tags.
- New UIL syntax specifies the resources of automatically created children.
- Syntax changes to the Widget Meta-Language (WML) allow for the definition of automatically created children of composite widgets.

- UIL can now use the `-wmd file` flag to read binary databases (WML files) that contain WML information.
- Mrm includes two new functions:
  - `MrmOpenHierarchyPerDisplay`

    This function is the same as the old `MrmOpenHierarchy` function, except that in the new function, the display is passed as an explicit argument. This function replaces `MrmOpenHierarchy`.
  - `MrmFetchBitmapLiteral`

    This function fetches a bitmap literal with a depth of 1.

Index

A

Access control, 2–2
  enabling and disabling, 5–6
AccessX support
  keyboard enhancements, 2–28
Ada bindings
  Release 1.1.3, 4–5
Answerback message
  DECterm, 2–12
Applications
  compiling Fortran, 4–12
  for DECwindows Motif, 2–3
Auto Repeat setting
  changing in DECterm, 2–10

B

Bindings support
  languages, 4–5
Bookreader
  draft-quality printing, 2–3
  using comments in the DECW$BOOKSHELF file, 2–3

C

Calendar
  after installing, 2–3
  double clicking becomes disabled, 2–3
  small message dialog box, 2–3
Cardfiler
  XUI version, 2–4
CDA
  applications, 2–4, 2–5
    dynamic font support, 2–4
    packing, 2–6
    unpacking, 2–7
    WRITE$FONT logical name, 2–5
  changes to external reference processing, 4–15
  converters, 1–5
  converting Asian-language text files, 1–5
  DECfonts, 2–9
  defining logical names, 1–4
  Display Options..., 2–9
  Display PostScript, 2–8
  drag-and-drop
CDA
  drag-and-drop (cont'd)
    implementing, 4–13
  internationalization support, 1–3
  logical names with convert, 4–15
  message, 2–8
  packing and unpacking error messages, 2–8
  paper size button, 2–9
  PostScript documentation, 2–5
  programming
    external referencing, 4–15
    interface changes, 4–14
    logical names with convert, 4–15
    release notes, 4–13
    restructuring of shareable images, 4–15
    style guide message, 4–15
  restructuring of shareable images, 4–15
  specifying an options file, 1–3
  style guide message, 4–15
  Version 1.8A enhancements, 4–13
  viewing PostScript files with errors, 2–9
  Watch Progress, 2–9
C header files, 4–12
Clock
  DECsound for Clock, 2–9
Color Customizer, 3–2
  auto shadowing toggle button, 3–5
  building, 3–2
  command summary, 3–3
  DECterms unaffected, 3–5
  mapping color resources and color cells, 3–4
  modifying DECW$LOGIN.COM, 3–3
  running, 3–3
  supported
    applications, 3–2
    displays, 3–2
  using on multi-head systems, 3–6
  xsetroot_cust demo, 3–6
Command files
  for checking version, 3–9
Compatibility
  shareable images, 4–18
Compiling applications
  See also Applications
  Fortran, 4–12

Compound Document Architecture
  See CDA
Console port
  selecting, 1–17
Console Window
  console messages, 1–17
  controlling the initial position, 1–17
  defining a global symbol, 1–17
  displaying console messages, 1–17
    DISABLE (default), 1–17
    ENABLE, 1–18
    values, 1–17
    WINDOW, 1–17
  invoking, 1–17
  selecting the Alternate Console port, 1–17
Copyright notice
  displaying in a DECterm window, 1–2
Cross-development tools
  OpenVMS Migration Toolkit, 4–11
  unpacking save sets, 4–11
Cross-hair cursor support
  See Input cursors
Cursor support
  See Input cursors

D

Debugger
  using in DECterm, 3–11
DEC CDA Base Services, 4–13
DECchart application
  invoking from the Session Manager, 2–20
DECnet/OSI full names support, 1–3
DECsound
  notes, 2–9
  prerequisites, 2–9
  restrictions, 2–10
DECterm
  See also Input cursors
  answerback message, 2–12
  automatic window positioning, 3–11
  changing the Auto Repeat setting, 2–10
  DECCRA sequence, 4–17
  DECLFKC sequences, 4–17
  default font size, 2–15
  diagnostic crash file, 2–15
  diagnostic crash messages, 2–15
  displaying the copyright notice, 1–2
  escape sequences, 2–10
  finish printing option, 5–3
  graphics, 2–13
  hold-screen response time, 3–11, 5–5
  initialization, 2–14
  local echo, 2–11
  logicals, 3–10
  page-movement sequences, 4–16
  positioning, 2–11
DECterm (cont'd)
  /PROCESS problem, 2–13
  programming issues, 4–16
  ReGIS input cursors, 1–2, 4–17
  ReGIS locator report, 2–15
  reporting window size, 1–2
  resizing the terminal, 1–2
  resource usage, 2–15
  screen print services, 2–13
  scrolling through the keyboard, 1–2
  seven-bit printer support, 2–12
  system management, 3–10
  terminal emulator, 2–12
  user font selection, 2–11
  using the debugger, 3–11
  virtual terminal support, 3–12
DECW$CDPLAYER
  required privileges, 2–16
DECW$CONSOLE_GEOMETRY
  global symbol, 5–5
DECW$CONSOLE_SELECTION
  global symbol, 5–5
DECW$INCLUDE:TEXTP.H header file
  definition changes, 4–19
DECW$INCLUDE:XMP.H header file
  See also Header files
  changes, 4–19
  removed definitions
    XmHALFLONGBITS, 4–19
    XmLONGBITS, 4–19
DECW$LOGIN.DAT
  customizing the login screen, 3–7
DECW$UTILS
  global symbols, 3–6
DECwindows Extensions to Motif, 4–29
DECwindows OSF/Motif Toolkit
  See Toolkit
DECwindows transports
  POSIX support, 4–8
Development support, 4–1
Dialog boxes
  changing the position
    Set Password, 3–8
    Start Session, 3–8
Digital writers
  sending comments to, iii
Display PostScript Server Extension, 4–30
Display server
  shared memory extension, 4–31
  shared memory pixmaps, 4–35
  shared memory XImages, 4–32
  supporting AccessX, 2–28
Documentation
  sending comments to Digital writers, iii
Documentation titles and versions
  DECwindows Motif for OpenVMS, 5–1

Drag-and-drop
- using, 2–2, 4–13, 5–3

DXmCSText widget
- input method support, 4–29

DXmFormSpaceButtonsEqually routine
- sizing and spacing widgets, 4–30

DXmLAYOUT_LEFT_DOWN constant
- DXmNlayoutDirection resource, 5–7

DXmLAYOUT_LEFT_UP constant
- DXmNlayoutDirection resource, 5–7

DXmLAYOUT_RIGHT_DOWN constant
- DXmNlayoutDirection resource, 5–7

DXmLAYOUT_RIGHT_UP constant
- DXmNlayoutDirection resource, 5–7

DXmNlayoutDirection resource
- constants, 5–7

E

Escape sequences
- DECterm, 2–10

F

Feedback on documentation
- sending comments to Digital writers, iii

FileView
- application startup, 3–7
- private logos, 2–22
- removing
  - LinkWorks Manager verb, 2–20
  - LinkWorks Setup verb, 2–20

Finish printing option
- DECterm, 5–3

Font resources
- See Mosaic

Fortran
- compiling applications, 4–12

Full names support
- DECnet, 1–3

Functional resources
- See Mosaic

G

GET_CHAR_STRUCT function
- access, 5–6

Global symbols
- DECW$CONSOLE_GEOMETRY, 5–5
- DECW$CONSOLE_SELECTION, 5–5

H

Header files
- DECW$INCLUDE:TEXTP.H, 4–19
- DECW$INCLUDE:XMP.H, 4–19
- languages, 4–5

Help
- invoking through the workspace Help menu, 2–19

Hold-screen response time
- enhancing, 5–5

Home page
- See also Mosaic
- creating, 1–7

I

Input cursors
- cross-hair, 4–17
- cross-hair support, 1–2
- diamond, 4–17
- rubber-band line, 4–17
- rubberband-line support, 1–2
- rubber-band rectangle, 4–17
- selecting, 4–17

Internationalization
- converting files, 1–3
- viewing files, 1–3

Internet
- See also Mosaic
- browsing, 1–6

K

Keyboard
- enhancements for disabled users, 2–28
- scrolling in DECterm, 1–2

L

Languages
- available in the Session Manager Language Options dialog box, 2–21

LinkWorks Manager
- removing
  - from FileView, 2–20
  - from Session Manager, 2–20
- removing from Session Manager, 2–20
- support removed, 2–16

Local echo
- DECterm, 2–11

Logical names
- defining in the CDA Viewer, 1–4

Login logos
- customizing, 3–9, 5–6

Login screen
- changing colors, 3–8
- customizing, 3–7
  - DECW$LOGIN.DAT, 3–7

Logos
- modifying the Digital logo, 3–8

M

Mail
- color customizer, 2–16
- enhanced performance, 2–16
- keyboard actions, 2–16
  - activating pushbuttons in dialog boxes, 2–16
  - highlighting selections, 2–16
- performance
  - turning on the word-wrap option, 2–17
- selecting arrow buttons in Read window, 2–17
- setting input focus, 2–17
- tear-off menus, 1–2, 2–2
- using the message button, 2–17

Messages
- System Menu Bar: Pseudo Mouse not available, 3–10

Migration Toolkit for OpenVMS
- See Cross-development tools

MIT X11
- Release 3 Intrinsics
  - messages, 4–18
- Release 4 Intrinsics
  - messages, 4–18

Mosaic
- configuring
  - processing
    - Audio files, 1–7
    - GIF files, 1–7
    - MIME files, 1–7
    - MPEG files, 1–7
    - multimedia files, 1–7
    - PostScript files, 1–7
- creating a home page, 1–7
- determining an external viewer, 1–7
- external view setup
  - examples, 1–8
- file location, 1–7
- firewall, 1–16
- font resources, 1–14
- functional resources, 1–9
- mappings, 1–8
- NoProxy resource
  - restrictions, 1–16
- providing as the Internet browser, 1–6
- proxy gateway resources, 1–15
- restrictions, 1–16
- setting Mosaic X resources, 1–8
- starting, 1–7
  - an external program, 1–7
- supporting the NoProxy resource, 1–16
- supporting the ProxyGateway resource, 1–16
- visual resources, 1–13

Motif and XUI widgets
- mixing, 4–18

Motif Window Manager
- centering lines for multiline icon titles, 2–19
- color customizer, 2–17
- configuration file, 2–17
- customizing
  - colors, 2–18
  - colors on multihead systems, 2–19
  - icon box, 2–19
  - icon placement, 2–18
- invoking Help, 2–19
- modal dialog boxes, 2–18
- moving the icon box off the screen, 2–19
- restarting, 2–18
- running earlier versions of DECwindows, 2–17
- starting an application as an icon, 2–18
- supporting customization for monochrome monitors, 2–18
- truncated icon titles, 2–19
- using an icon box, 2–18

MWM
- See Motif Window Manager

N

Node name display
- disabling, 3–9

Notepad
- linking with the Release 1.1.3 Toolkit, 2–19

O

Options file
- specifying, 1–3

OSF/Motif Toolkit
- See also Toolkit
- examples, 4–19
  - cutpaste, 4–19
  - dnddemo, 4–20
  - Dogs, 4–21
    - widget, 4–21
  - helloint, 4–22
  - hellomotif, 4–22
  - MOTIFANIM, 4–22
  - MOTIFGIF, 4–23
  - motif samples, 4–27
  - motifshell, 4–23
  - periodic, 4–24
  - pict, 4–23
  - PICT, 4–23
  - resource files for example programs, 4–29
  - Square widget, 4–21
  - textedit, 4–24
  - uid files for example programs, 4–29
  - view, 4–25
  - xmpiano, 4–26
  - xmtravel, 4–28
- known problems, 5–8

Overlay support
  See also Window manager
  colormap
    avoiding potential problems, 1–19
  modifying applications, 1–19
  sharing overlay colormaps with the Window Manager, 1–19

P

Paint
  creating private colormaps, 2–20
  enhancing performance during basic operations, 2–20
Parameter datasize
  mismatches, 4–36
Pascal programming
  .PEN files, 4–36
PEN files
  using with Pascal programs, 4–36
POLYCENTER Software Installation utility
  description, 1–1
  using to install OpenVMS Version 6.2 systems, 1–2
POSIX support
  DECwindows transports, 4–8
  SYS$POSIX_FORK_CONTROL system service, 4–9
  using the POSIX fork routine, 4–8
Printing
  Bookreader, 2–3
Print Screen
  default icon, 2–20
Process
  creating with virtual terminal support, 3–12
Programming environments, 4–1
Programming support and XUI, 4–5
Proxy gateway resources
  See Mosaic
Public profiles
  creating files for layered products, 2–23
Pushbuttons
  DECwindows Mail
    activating in dialog boxes, 2–16

R

ReGIS input cursors
  See also Input cursors
  DECterm, 1–2
Resources
  See also Mosaic
  font, 1–14
  functional, 1–9
  proxy gateway, 1–15
  setting in Mosaic, 1–8
  visual, 1–13
Rubberband-line cursor support
  See Input cursors
Run-time support, 4–1

S

Symbols
Save sets
  unpacking using the BACKUP commands, 4–11
Security options, 5–5
  Session Manager, 2–22
Sending comments to Digital writers, iii
Session Manager
  color customizer, 2–22
  DECchart application, 2–20
  known problems
    private logos, 2–22
  language options, 2–21
  removing
    LinkWorks Manager verb, 2–20
    LinkWorks Setup verb, 2–20
  security options, 2–22
  stopping a process, 2–22
Set Password dialog box
  changing the position, 3–8
Seven-bit printer support
  DECterm, 2–12
Shareable images
  compatibility, 4–18
  upward compatibility, 4–18
Shareable linkage
  installing images, 1–18
Shared memory
  creating and using XImages, 4–32
  extension support, 4–31
  pixmaps, 4–35
Start Session
  blue login screen, 3–1
Start Session dialog box
  changing the position, 3–8
Structured Visual Navigation
  See SVN
SVN (Structured Visual Navigation)
  horizontal separator line, 4–30
  widget, 4–30
SYS$POSIX_FORK_CONTROL system service
  registering
    addresses and parameters, 4–9
    protected shareable images, 4–9
System menu bar
  messages, 3–10

T

Tear-off menus
- DECwindows Mail, 1–2
- using, 5–3
- using in Mail, 2–2

Terminal
- resizing, 1–2

Toolkit
- compatibility, 4–2
  - naming in OSF/Motif Release 1.2.2 and X11 R5 Shareable Libraries, 4–3
  - using _Xm routines, 4–3
- compilation errors, 4–17
- compiling C modules using the MEMBER_ ALIGNMENT switch, 4–11
- description of OSF/Motif Release 1.2.3 environment, 1–1
- description of Release 1.2.3, A–1
- extensions
  - DXmCSText widget, 4–29
- locale error message, 2–3
- mixing Motif and XUI Widgets, 4–18
- modifying XmText and XmTextField translation manager syntax, 4–18
- OSF/Motif Release 1.2.3, 4–1
- release 1.1.3 applications abort, 2–29
- saving programming environments, 4–5

Toolkit shareable images
- See Shareable images

Translated-image support, 4–9
- running on OpenVMS systems, 4–9

Translations
- unsupported with XmText Widget, 2–29

Transports
- POSIX support, 4–8

U

UIL source code
- documentation, 5–8

V

Version checking command files, 3–9

Virtual terminal support
- creating a process, 3–12

Visual resources
- See Mosaic

W

Widgets
- See also Motif and XUI widgets, 4–18
- sizing and spacing using DXmFormSpaceButtonsEqually, 4–30

Window Dump utility
- dump to print file (xpr), 2–23

Window Manager
- overlay support, 1–19

Window size
- reporting for DECterm, 1–2

X

X Image extension, 4–30

Xlib
- Digital international Xlib implementation, 4–38
- extensions
  - client side library, 4–31
- Internationalization, 4–37
- locale, 4–39
- routines
  - files for Pascal programs, 4–36
  - parameter datasize, 4–36
  - xlibint.h, 4–36
  - XSelectAsyncEvent, 4–37
  - XSelectAsyncInput, 4–37
- vendor pluggable layer, 4–38

XmNinputMethod resource
- using the shell to specify input methods, 4–29

XmText Widget
- unsupported translations, 2–29

XSelectAsyncEvent routine
- allocating memory, 4–37

XSelectAsyncInput routine
- allocating memory, 4–37