AA–MG17A–TE

Overview of VMS DECwindows

December 1988

This book summarizes features of DECwindows for VMS Version 5.1 and provides an overview of VMS DECwindows documentation.

Revision/Update Information:	This is a new book.
Software Version:	VMS Version 5.1

digital equipment corporation
maynard, massachusetts

December 1988

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

ZK4727

Production Note

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

Contents

Figures

Tables

About This Guide

Who Should Use This Guide

This guide is intended for new users of VMS DECwindows and for developers who write applications for the DECwindows environment.

Structure of This Guide

This guide is organized as follows:

- Chapter 1 provides background information about the VMS DECwindows system and describes the DECwindows architecture.
- Chapter 2 describes the interfaces and applications that comprise the VMS DECwindows user environment.
- Chapter 3 describes the library routines and tools that comprise the VMS DECwindows programming environment.
- Chapter 4 describes the books included in the VMS DECwindows documentation set.

Related Documents

Other books in the VMS DECwindows documentation set provide complete information about the DECwindows programming and user environments. The VMS DECwindows documentation set includes the following books:

- *VMS DECwindows User's Guide*
- *VMS DECwindows Desktop Applications Guide*

- *XUI Style Guide*
- *VMS DECwindows Guide to Application Programming*
- *VMS DECwindows User Interface Language Reference Manual*
- *VMS DECwindows Toolkit Routines Reference Manual*
- *VMS DECwindows Guide to Xlib Programming: VAX Binding*
- *VMS DECwindows Guide to Xlib Programming: MIT C Binding*
- *VMS DECwindows Xlib Routines Reference Manual*
- *VMS DECwindows Device Driver Manual*

Conventions

The following conventions are used in this guide:

mouse	The term *mouse* is used to refer to any pointing device, such as a mouse, a puck, or a stylus.
boldface text	Boldface text represents the introduction of a new term.

1 Introduction to VMS DECwindows

If DIGITAL could take a snapshot that reflected the computing environment of most of its customers, the picture would show the following:

- The era of the standalone system is ending.
- The new "system" is the network.
- The network often consists of a variety of hardware, including PCs and graphic workstations.

A computing environment that emphasizes the network as the system creates a new set of customer needs. These needs include the following:

- The ability to distribute applications throughout the network, letting customers take full advantage of network resources.
- A software environment that supports the advanced graphics capabilities of workstations and PCs.
- A software development environment that makes it easy to create applications with a consistent look and feel—regardless of the hardware and operating system on which the applications run.
- The ability to use workstations and PCs to display multiple applications concurrently—accomplished most easily in a windowed environment. Each application has a window on the screen that displays output and accepts input.

To address these needs, DIGITAL provides VMS DECwindows.

This chapter summarizes the features of the VMS DECwindows system and describes the DECwindows architecture.

DECwindows Features

DECwindows is an advanced windowing system that extends and improves the X Window System, Version 11.

The X Window System was developed at the Massachusetts Institute of Technology to fulfill the needs of two groups. MIT's Project Athena, relying on a large network of graphic workstations, needed a windowing system to help make the workstation displays useful. Another group wanted a windowing system for developing distributed applications. While meeting the needs of these groups, MIT sought to develop a system that would support a variety of hardware configurations and operating systems.

In essence, MIT created a windowing system that met their needs by designing an architecture that allows the execution and display of applications to be independent. Specific components of the architecture control the display of applications. Different components determine how applications run.

Since its introduction by MIT, the X Window System has become an industry standard.

DECwindows incorporates and extends the X Window System to provide a windowing environment with the following features:

- Graphics-oriented interaction with the VMS operating system

  DECwindows handles all user interactions with the system. These interactions include initializing a session, managing your environment and resources, starting applications, and managing windows. You can create windows, move windows, resize windows, and shrink windows to icons.

- Consistent user interfaces

  DECwindows user interfaces consist of graphic objects that look and function the same, regardless of the application you are using. This makes it easy to learn about and use new applications.

- A library of desktop applications

  DECwindows provides a variety of applications, including the following: Bookreader, Calculator, Calendar, Cardfiler, Clock, DDIF Document Viewer, EVE Text Editor, FileView, Mail, Notepad Text Editor, Paint Graphics Editor, PostScript Previewer, and DECterm.

- Availability of a wide range of third-party applications

  Because DECwindows incorporates the industry standard X Window System, an extensive selection of third-party applications is available for VMS DECwindows systems.

- Powerful libraries of routines that simplify the development of graphics-oriented applications

  These routines perform functions such as drawing and window management. Applications using these routines run on all supported hardware without modifications. For example, an application designed for the VAXstation 2000 looks and functions the same on a VAXstation 3200.

- Libraries for creating and accessing documents containing text, graphics, and scanned images.

- Network transparent application interface

  The DECwindows architecture lets you run an application on a remote node, while displaying and inputting data on a local workstation. The network functions that make this possible are transparent; the application appears to run locally.

  Network transparency allows workstations to access the power and resources of other systems on the network.

- Compatibility with ULTRIX

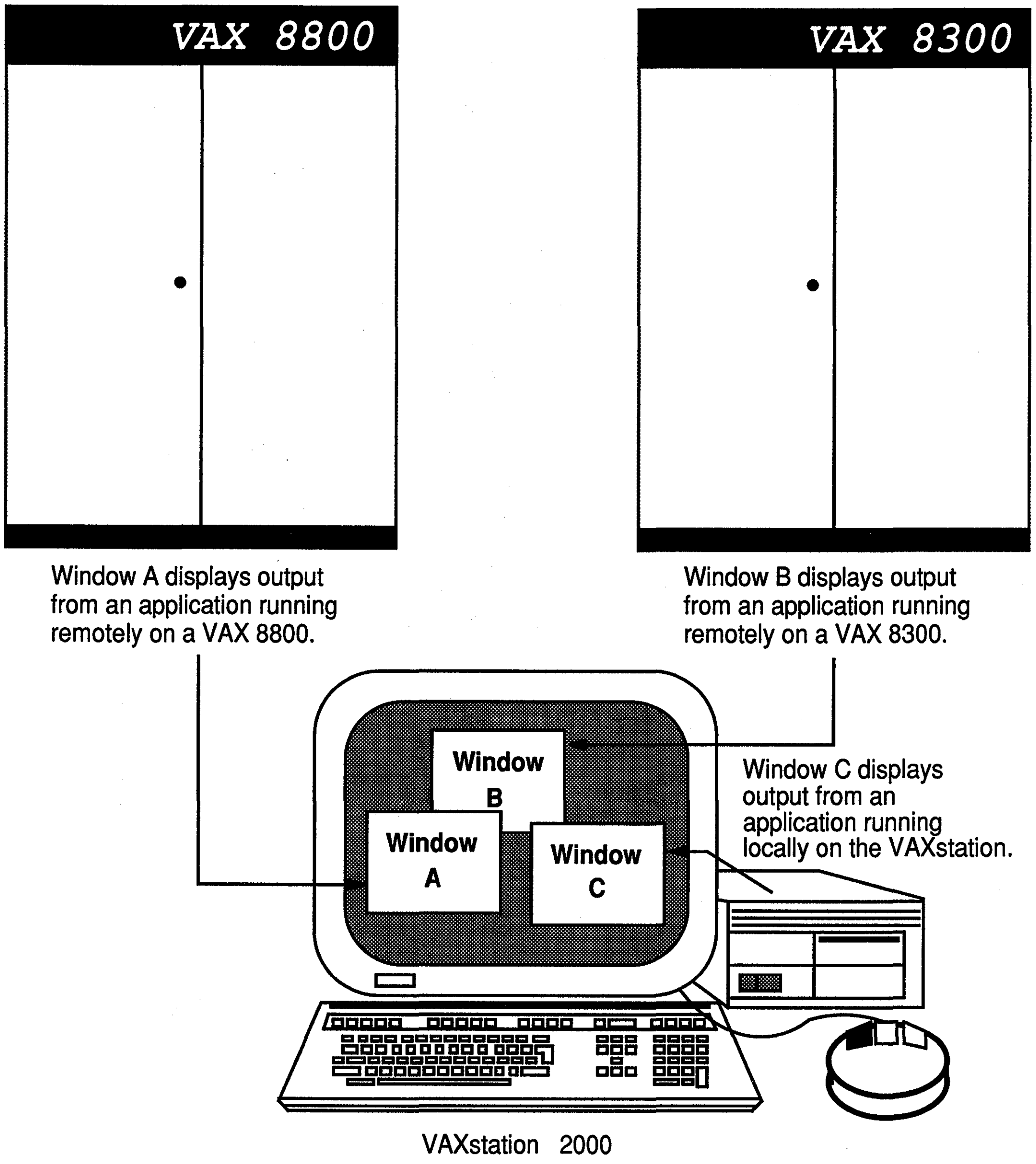
  Because VMS and ULTRIX DECwindows use the same set of programming libraries, you can easily port applications to and from ULTRIX DECwindows systems; no modifications to applications are required.

- Extensible architecture

  The DECwindows architecture has been designed to accommodate new technology, such as three-dimensional graphics, as it becomes available.

Figure 1–1 illustrates several features of DECwindows. Three applications display a window on a VAXstation. Two of the applications run on remote processors, yet all the applications display output and receive input from the local workstation.

Figure 1–1 Network Transparent DECwindows Applications


VAX 8800
VAX 8300
Window A displays output from an application running remotely on a VAX 8800.
Window B displays output from an application running remotely on a VAX 8300.
Window B
Window A
Window C
Window C displays output from an application running locally on the VAXstation.
VAXstation 2000

ZK–0085A–GE

DECwindows Architecture

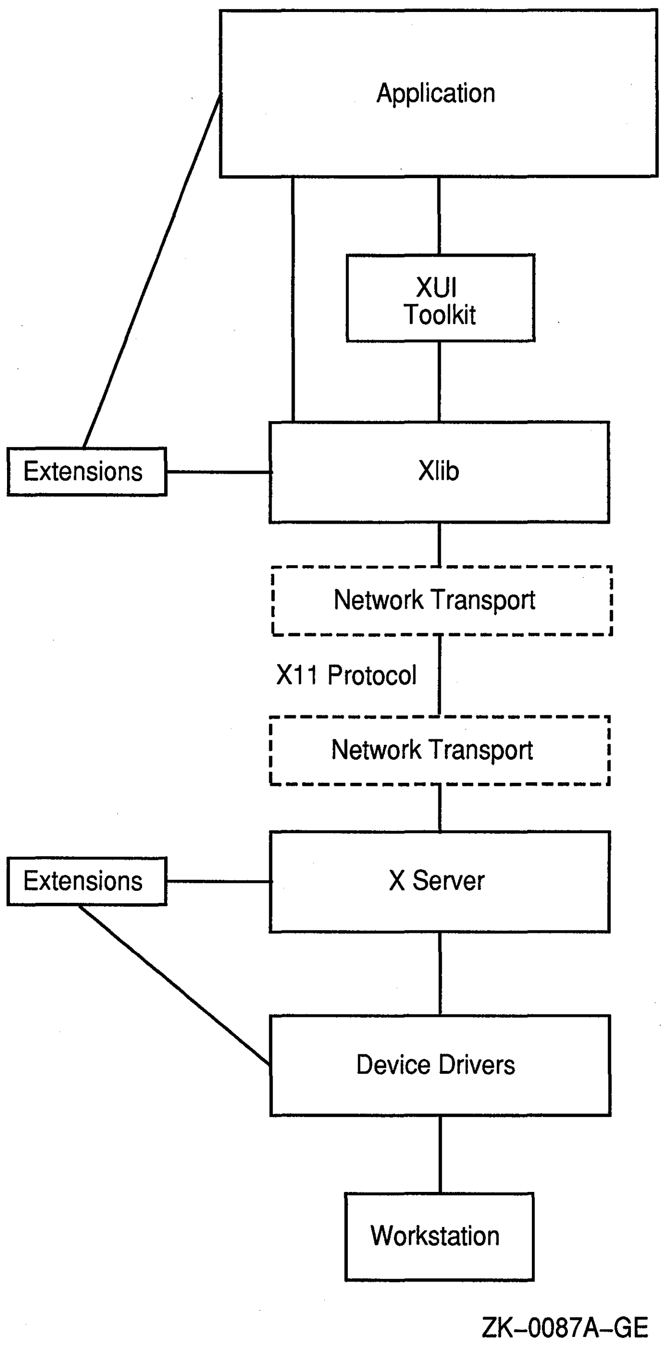
The DECwindows architecture is an extension of the X Window System architecture. Components of the architecture include the following:

- Applications
- X User Interface (XUI) Toolkit
- Xlib
- Network transport
- Server
- Device drivers

In addition to these basic components, DECwindows includes the Compound Document Architecture (CDA) Toolkit. The CDA Toolkit allows applications to create and access documents containing text, graphics, and scanned images. Figure 1–2 illustrates the relationships among the components of the DECwindows architecture.

The following sections explain the components.

Figure 1–2 DECwindows Architecture



Applications

To perform windowing and graphics functions or get input from the mouse and keyboard, applications call routines from the X User Interface (XUI) Toolkit and Xlib programming libraries. Applications can call library routines from any language using the VAX calling conventions or from the C language using the MIT C calling conventions.

XUI Toolkit

The XUI Toolkit consists of high-level library routines that allow applications to create and manage a user interface. Using XUI Toolkit routines, applications can create, modify, and control interface objects such as menus, scroll bars, and buttons. The XUI Toolkit includes DIGITAL's implementation of the X Window System's X Toolkit.

XUI Toolkit routines call Xlib routines to perform fundamental input and output functions.

Xlib Programming Library

The Xlib programming library is DIGITAL's implementation of the X Window System's graphics programming library. Xlib provides low-level routines for creating windows, managing windows, and performing graphic functions.

Xlib routines handle application requests by generating and sending X11 protocol requests through the network to another component of the architecture called the **server**; the server runs locally on the displaying workstation. The X11 protocol defines the common language through which an application and workstation communicate.

Xlib processes requests from applications in an asynchronous manner. Xlib processes some application requests (for example, queries about the characteristics of text) within the library; it sends other requests (for example, requests to display graphics) to the server.

When a user enters data using the keyboard or mouse, Xlib passes data back to the application.

Transport Layers

The network transport mechanism uses DECnet to handle communication, in the form of X11 protocol packets, between Xlib and the server.

The implementation of DECnet and the X11 protocol allows the application and the user interface to be physically separated across the network. In other words, you can run an application on a remote node while displaying the window locally on a workstation. The window accepts input and displays output for the application. The application appears to run on the workstation.

DECwindows also provides an optimized, shared-memory transport mechanism. When an application runs on the same workstation that displays the interface, the optimized transport reduces the time required to transfer a protocol packet to or from the server.

Server

The DECwindows server is DIGITAL's implementation of the X Window System server. The server is the component of the architecture that allows application interfaces to interact with all supported systems in the same way. Residing on the displaying system, the server receives protocol requests from the transport layers and performs the functions required to fulfill the request for a specific device. In essence, the server converts the data that represent the request to a command that can be executed by the appropriate device drivers.

When the user of an application enters data, the server receives input from the device drivers and passes protocol packets back through the transport layers to Xlib and XUI Toolkit routines.

The server supports asynchronous input from the user to the application and asynchronous output from the application to a display.

Server Components The server consists of two distinct components. The first component, the kernel, performs a variety of functions related to the management of requests. These functions include the following:

- Initializing the server
- Scheduling requests
- Interpreting and dispatching protocol requests

The kernel also performs device-independent graphic and windowing functions.

The second component of the server performs all device-dependent functions related to the request. These functions include device-dependent drawing and windowing functions, graphics attribute management, and the execution of drawing requests.

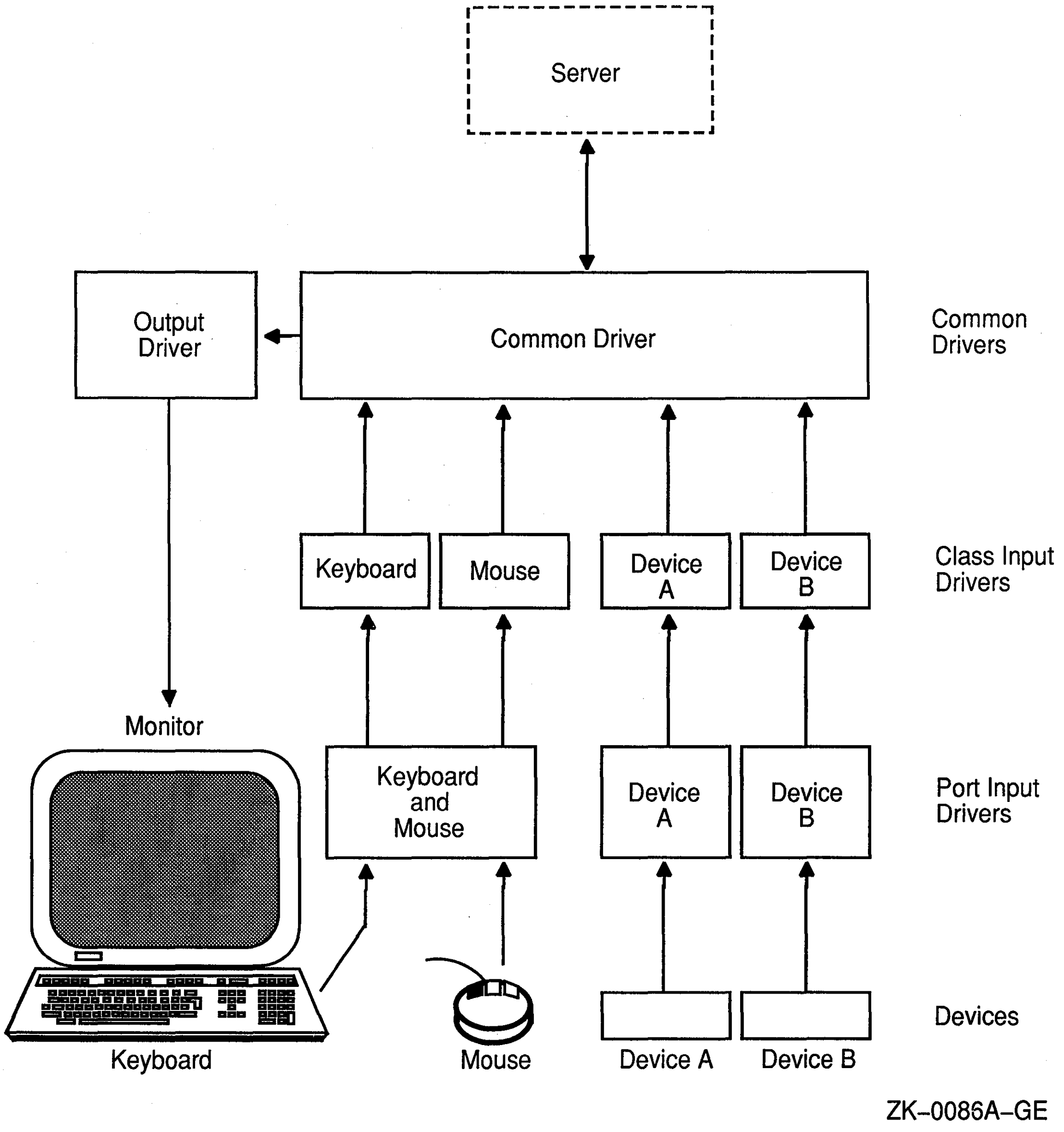
Device Drivers

Device drivers process requests from the server to the display and pass data from the input devices to the server. The device driver architecture consists of the following components:

- Port input drivers
- Class input drivers
- Common driver
- Output driver

Figure 1–3 illustrates the components of the DECwindows device driver architecture. The sections that follow explain the components. For a complete description of the device drivers, see the *VMS DECwindows Device Driver Manual.*

Figure 1–3 DECwindows Device Drivers



Port Input Drivers The port input drivers service hardware interrupts from input devices. With the exception of the keyboard and mouse, each device has its own port input driver; the keyboard and mouse share a port driver. The port input drivers pass byte-stream input to the class input drivers.

Class Input Drivers The class input drivers decode the byte streams provided by the port input drivers and translate the data to a format the server can interpret.

Common Driver The common driver is the interface between the server and drivers. It passes input to the server from the class input drivers, and passes output requests from the server to the output driver.

Output Driver After receiving directions from the common driver, the output driver sends device-specific output to the workstation.

Extending the Architecture

One of the most important features of the DECwindows architecture is its extensibility. The programming libraries, X11 protocol, and server can be extended to support new hardware or to implement new features.

For example, extending DECwindows to support three-dimensional applications would require the following:

- Developing a new library of routines to create and manage three dimensional images
- Adding protocol requests that correspond to the new routines
- Extending the server so that it can interpret and execute the requests

The DECwindows architecture is also designed to support optional libraries that layer on top of Xlib and the XUI Toolkit, such as the GKS programming library. These libraries do not require extensions to the protocol or server.

Security

The full range of VMS security features are available to DECwindows systems. In addition to VMS security features, DECwindows provides the ability to specify which users and systems can access a server.

Architecture Summary

The following example provides an overview of the DECwindows architecture:

A customer runs a DECwindows application on a VAXstation 2000. Using the mouse, the customer selects a menu option to create a window. Input from the mouse passes from the device drivers to the server and then to Xlib, the XUI Toolkit, and the application. The application then calls XUI Toolkit routines to perform high-level windowing functions. XUI Toolkit routines call low-level Xlib routines to perform specific windowing and graphic functions. Xlib converts the calls to protocol requests that DECnet sends to the server. The server performs the translations required to create the window on the VAXstation 2000. Finally, the device drivers send the translated request to the workstation display, creating the window.

2

The User Environment

By providing consistent, graphics-oriented user interfaces, the DECwindows user environment makes it easy to interact with the operating system and use applications. The following chapter provides an overview of the user environment and describes the DECwindows desktop applications.

Overview of the User Environment

The DECwindows user environment is designed to increase productivity by saving time. To understand how DECwindows saves time, consider the following situation.

Imagine that you have been hired by a game company to test and evaluate children's board games currently on the market. You must purchase several hundred games, learn all the rules, play the games, and write evaluations. You have two weeks to complete the project. After a wild shopping spree at the local toy store, you bring the games home and start opening boxes. Some games have spinners, some standard dice, some special dice. Several use decks of cards, others use tables of information. Many include game boards, but just as many do not. In short, all the "game objects" used by the games are different; the rules for using those objects are also different.

You realize you can never meet your deadline. It will take two months just to learn to play the games.

Typically, the software application market has been similar to the children's game market. Application interfaces, analogous to game objects, show little consistency. As a result, you waste a great deal of time learning how to use new applications.

DECwindows provides an environment in which applications look and respond similarly. For example, suppose two different applications display a menu. The menu looks the same for both applications, and the steps required to pick a menu option are the same—you choose the option using the mouse and then press the mouse button. You interact with a DECwindows system by finding an object on the screen that represents the task you want to perform, then using the mouse to select that object. Because application interfaces consist of graphic objects, the functions of interfaces are often obvious.

Typical interface objects include the following:

- Menu bars
- Scroll bars
- Pop-up menus
- Pull-down menus
- Buttons

The DECwindows environment provides shortcuts, called **accelerators**, that let you interact with applications more quickly. **Keyboard accelerators** associate keyboard keys to the function of menu items. For example, you might press a key to copy text instead of selecting Copy from a menu. Accelerators differ from application to application, but the concept is the same: saving time for experienced users by eliminating unnecessary steps.

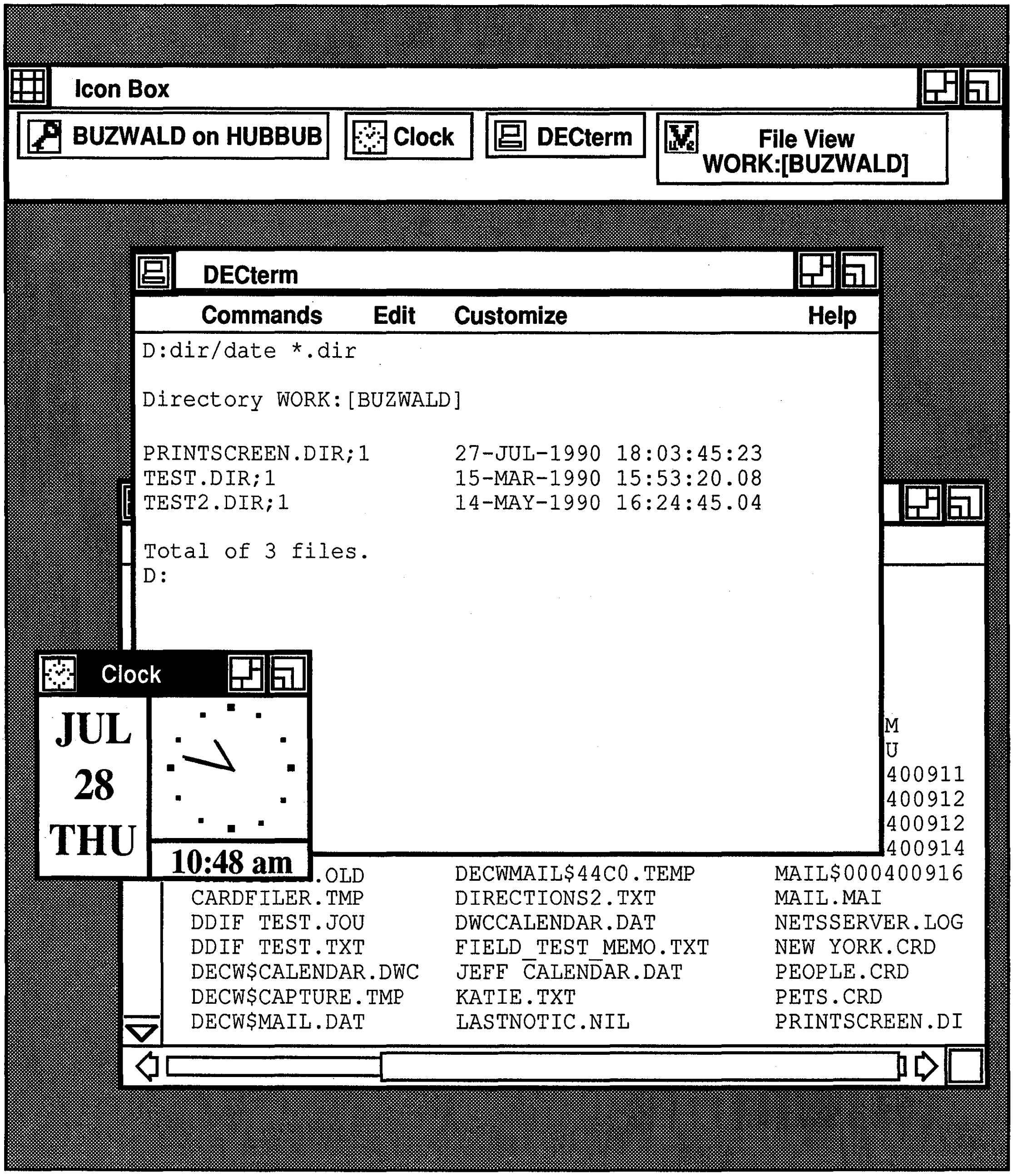
The DECwindows user environment also saves time by allowing you to perform several tasks at once. For example, you can run a program in one window, read mail in a second window, and execute a command procedure in a third.

The DECwindows user environment includes the following components:

- The Session Manager
- The Window Manager
- FileView
- DECwindows applications
- Compound Document Architecture (CDA) file converters
- Online help

Figure 2–1 illustrates the DECwindows user environment. The sections following the figure describe the components of the user environment.

Figure 2–1 DECwindows User Environment



```
D:dir/date *.dir

Directory WORK:[BUZWALD]

PRINTSCREEN.DIR;1      27-JUL-1990 18:03:45:23
TEST.DIR;1             15-MAR-1990 15:53:20.08
TEST2.DIR;1            14-MAY-1990 16:24:45.04

Total of 3 files.
D:
```

```
                                                    M
                                                    U
                                                    400911
                                                    400912
                                                    400912
                                                    400914
      .OLD                DECWMAIL$44C0.TEMP        MAIL$000400916
CARDFILER.TMP             DIRECTIONS2.TXT           MAIL.MAI
DDIF TEST.JOU             DWCCALENDAR.DAT           NETSSERVER.LOG
DDIF TEST.TXT             FIELD_TEST_MEMO.TXT       NEW YORK.CRD
DECW$CALENDAR.DWC         JEFF CALENDAR.DAT         PEOPLE.CRD
DECW$CAPTURE.TMP          KATIE.TXT                 PETS.CRD
DECW$MAIL.DAT             LASTNOTIC.NIL             PRINTSCREEN.DI
```

ZK–0210A–GE

The Session Manager

The Session Manager is the user's entry point into DECwindows. In order to perform tasks in the DECwindows environment, you must establish a session. The Session Manager controls and coordinates the sessions that you create. You use the Session Manager to do the following:

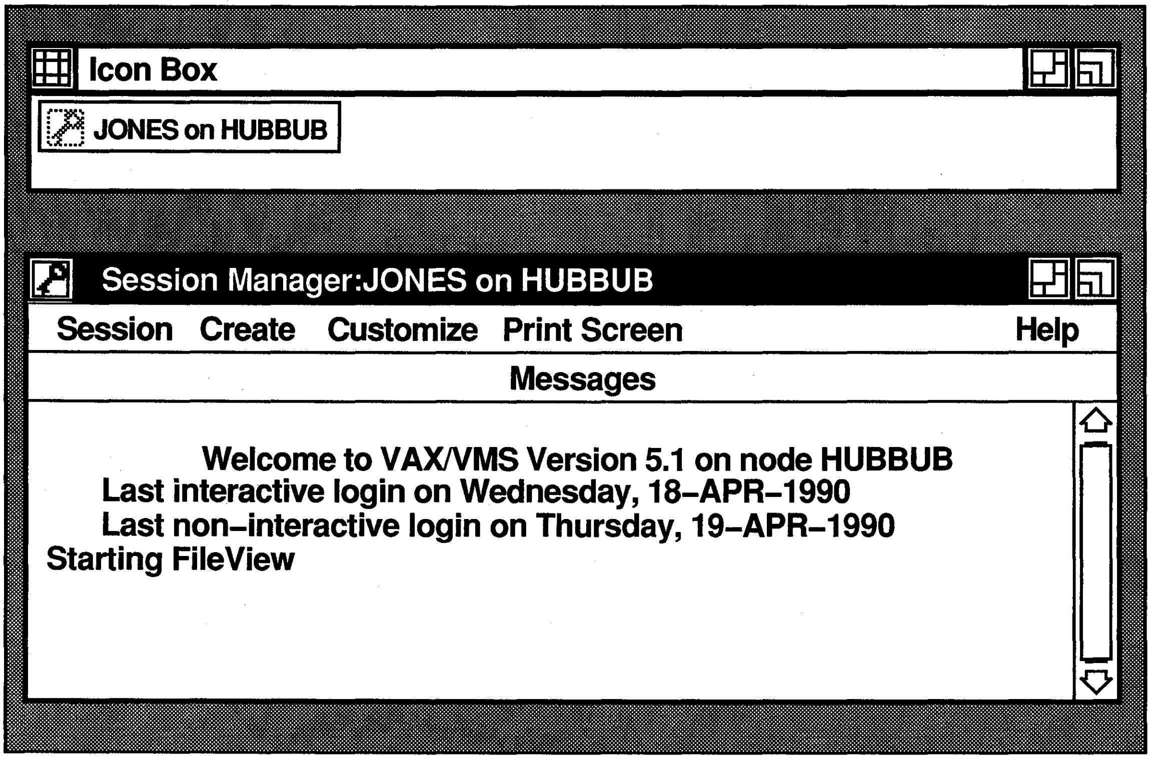
- Create new sessions
- Start the Window Manager and FileView
- Access the VMS operating system
- Print the contents of a display screen
- Customize the user environment
- End sessions

The Session Manager lets you create multiple windows and move freely between them.

You can also use the Session Manager to customize the user environment. For example, you can have the Session Manager run DECterm each time a new session begins. DECterm displays a window that looks and functions like a VT340 terminal.

Figure 2–2 shows the Session Manager window. For more information about the Session Manager, see the *VMS DECwindows User's Guide*.

Figure 2–2 Session Manager Window and Icon Box



```
        Welcome to VAX/VMS Version 5.1 on node HUBBUB
    Last interactive login on Wednesday, 18-APR-1990
    Last non-interactive login on Thursday, 19-APR-1990
Starting FileView
```

ZK–0209A–GE

The Window Manager

As you start applications and create windows, you might want to change the characteristics of windows. The Window Manager lets you make the following modifications:

- Change the size of windows
- Move windows
- Push windows to the bottom of a stack
- Pop windows to the top of a stack

If your screen becomes too crowded, you can use the Window Manager to shrink windows to icons. When you shrink a window, the Window Manager puts the icon in an area of the screen called the icon box. Windows that you shrink remain active but are hidden from view. To redisplay the entire window, you select the icon from the icon box.

For more information about the Window Manager, see the *VMS DECwindows User's Guide*.

FileView

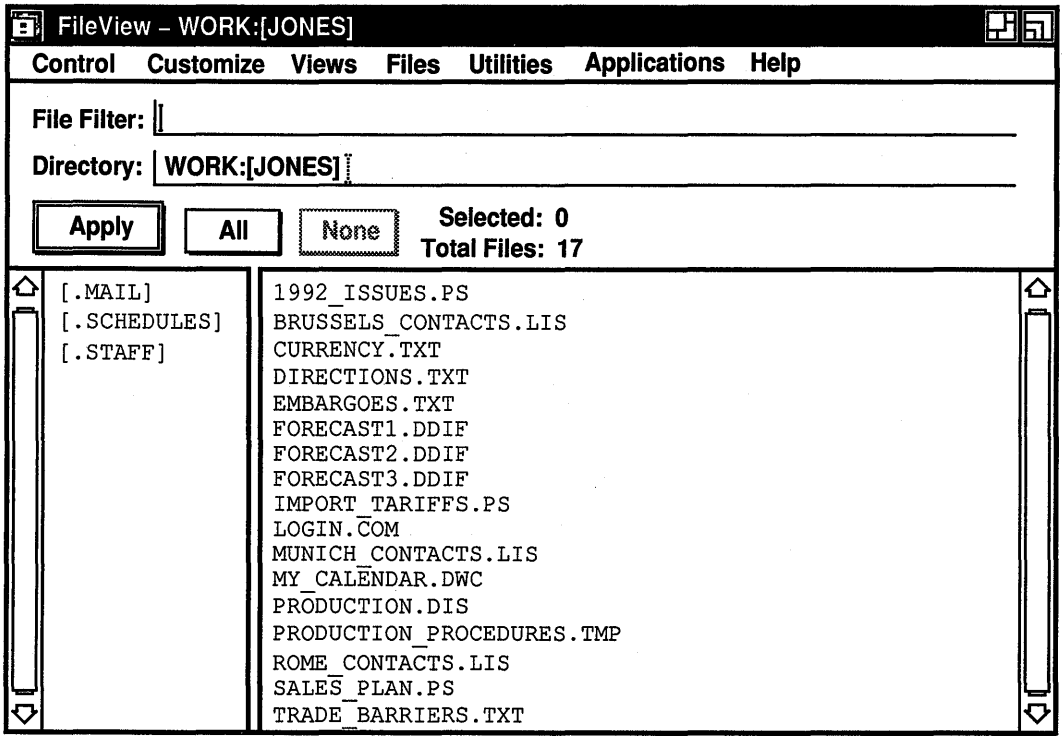
The FileView is a graphic interface to the VMS operating system. Using FileView, you can execute system commands and run DECwindows applications. Because FileView lets you make selections from menus, you do not need to remember the format of system commands.

When you run FileView, the main window displays the names of the files in your current directory. To execute a command, you select the file to which the command applies, then choose the appropriate command. FileView prompts you for parameters, then executes the command.

You can customize FileView to suit your work environment. For example, you can add menu options that perform additional file operations.

Figure 2–3 shows the FileView window. For more information about FileView, see the *VMS DECwindows User's Guide*.

Figure 2–3 FileView Window



```
WORK:[JONES]

[.MAIL]
[.SCHEDULES]
[.STAFF]

1992_ISSUES.PS
BRUSSELS_CONTACTS.LIS
CURRENCY.TXT
DIRECTIONS.TXT
EMBARGOES.TXT
FORECAST1.DDIF
FORECAST2.DDIF
FORECAST3.DDIF
IMPORT_TARIFFS.PS
LOGIN.COM
MUNICH_CONTACTS.LIS
MY_CALENDAR.DWC
PRODUCTION.DIS
PRODUCTION_PROCEDURES.TMP
ROME_CONTACTS.LIS
SALES_PLAN.PS
TRADE_BARRIERS.TXT
```

ZK–0501A–GE

DECwindows Applications

DECwindows includes a variety of applications that are designed to increase productivity by automating basic tasks. For example, Mail automates sending and receiving interoffice mail; the Cardfiler automates creating and filing index cards. Application interfaces are consistent, easy to learn, and easy to use.

Because DECwindows is based on the industry standard X Window System, customers will be able to choose from an extensive offering of applications from DIGITAL and other vendors.

DECwindows includes the following applications:

- Bookreader
- Calculator
- Calendar

- Cardfiler
- Clock
- DDIF Document Viewer
- EVE text editor
- Mail
- Notepad text editor
- Paint
- PostScript Previewer
- Puzzle
- DECterm

The following sections describe the DECwindows applications. For more information about DECwindows applications, see the *VMS DECwindows Desktop Applications Guide*.

Bookreader

The Bookreader allows you to read books in the VMS DECwindows documentation set on your workstation display.[1] The table of contents and index serve as tools for navigating the book. When you choose an entry from the table of contents or index, the Bookreader displays the text, figure, or table relating to that entry. The Bookreader provides several techniques for quickly scanning the contents of a book.

Calculator

The Calculator performs simple arithmetic functions: addition, subtraction, multiplication, division, percentages, and square roots. You can enter data by using the mouse to click buttons on the screen, or you can use the keyboard. The Calculator has two displays. One display shows the current operation. The other display shows the contents of memory.

[1] The complete online documentation set is available as a separate product.

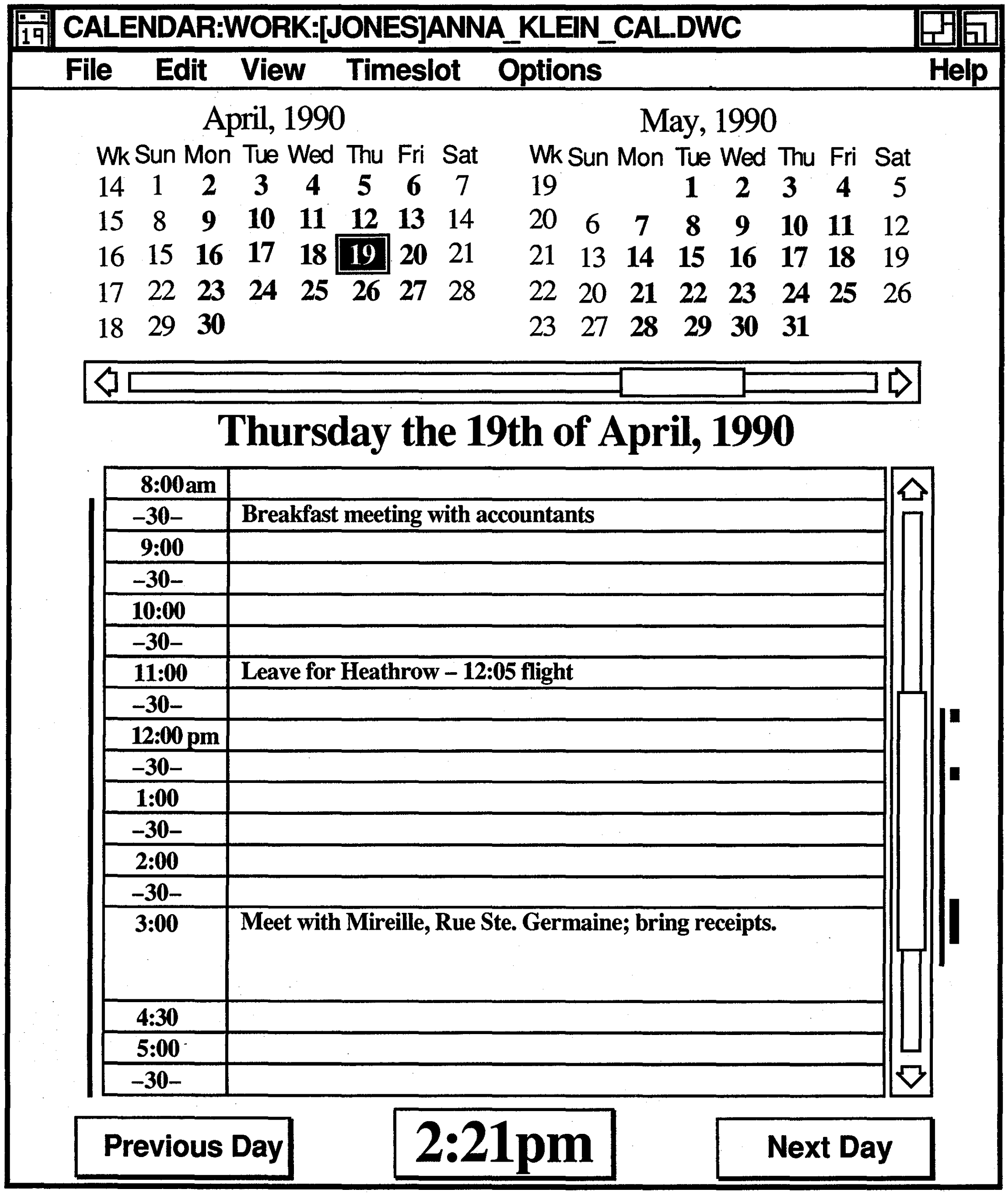
Calendar

The Calendar provides a convenient method for scheduling appointments and planning work. Using the Calendar, you can do the following:

- Display a day, week, month, or year
- Set alarms that remind you of upcoming events
- Create and maintain multiple calendars
- Customize the display

Figure 2–4 shows the Calendar window.

Figure 2–4 Calendar Window



April, 1990

Wk	Sun	Mon	Tue	Wed	Thu	Fri	Sat
14	1	2	3	4	5	6	7
15	8	9	10	11	12	13	14
16	15	16	17	18	19	20	21
17	22	23	24	25	26	27	28
18	29	30					

May, 1990

Wk	Sun	Mon	Tue	Wed	Thu	Fri	Sat
19			1	2	3	4	5
20	6	7	8	9	10	11	12
21	13	14	15	16	17	18	19
22	20	21	22	23	24	25	26
23	27	28	29	30	31		

Thursday the 19th of April, 1990

8:00am	
–30–	Breakfast meeting with accountants
9:00	
–30–	
10:00	
–30–	
11:00	Leave for Heathrow – 12:05 flight
–30–	
12:00 pm	
–30–	
1:00	
–30–	
2:00	
–30–	
3:00	Meet with Mireille, Rue Ste. Germaine; bring receipts.
4:30	
5:00	
–30–	

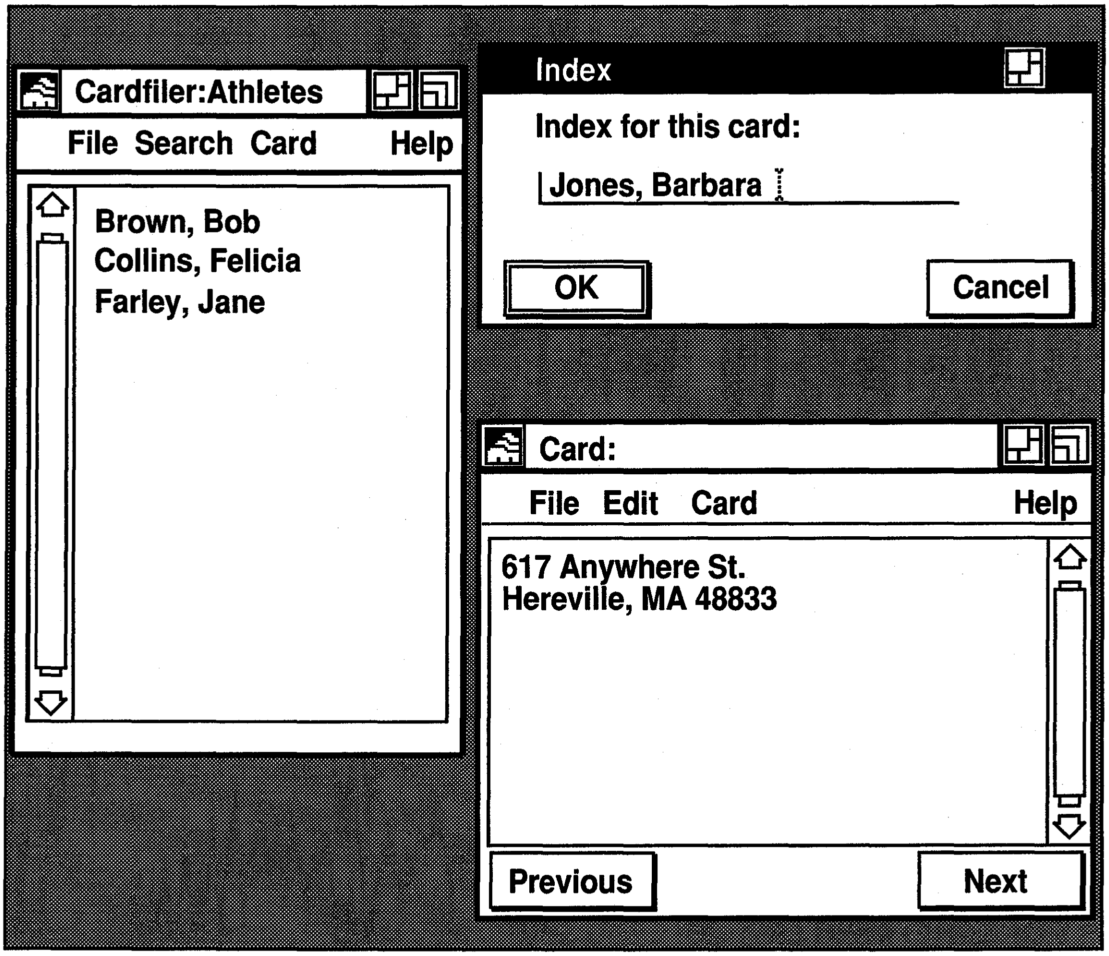
ZK–0098A–GE

Cardfiler

The Cardfiler provides a system for organizing information that is similar to using index cards. Using the Cardfiler, you create cards that contain information such as names, addresses, and phone numbers. You can also create files that group cards with similar information. For example, you might create a file for business cards and a file for personal cards.

Figure 2–5 shows the Cardfiler windows.

Figure 2–5 Cardfiler Windows



ZK–0099A–GE

Clock

The Clock displays the time (in analog and digital format) and date and provides an alarm to remind you of important events. You can also customize the display to produce different formats.

DDIF Document Viewer

The DDIF Document Viewer displays a DDIF file and lets you page through the contents. DDIF is a standard format for the storage and interchange of compound documents. Compound documents are documents that contain multiple elements (for example, text and graphics).

EVE Text Editor

The DECwindows EVE editor is a general-purpose text editor based on the VAX Text Processing Utility (VAXTPU). DECwindows EVE provides a graphic interface that lets you perform the following functions:

- Perform basic text-editing operations
- Create and edit more than one file in an editing session
- Use multiple buffers and windows, and resize windows
- Define keys and create learn sequences
- Set editing preferences, such as a bound or free cursor
- Use wildcards for searching text
- Execute system commands from within the editor
- Spawn subprocesses or attach to other processes
- Compile and execute VAXTPU procedures to extend the editor
- Create section files to save key definitions and extensions
- Use initialization files and set private defaults

Mail

The Mail application provides a way to communicate with other users on the system or network. Mail's graphic interface lets you perform the same functions as the VMS Mail Utility.

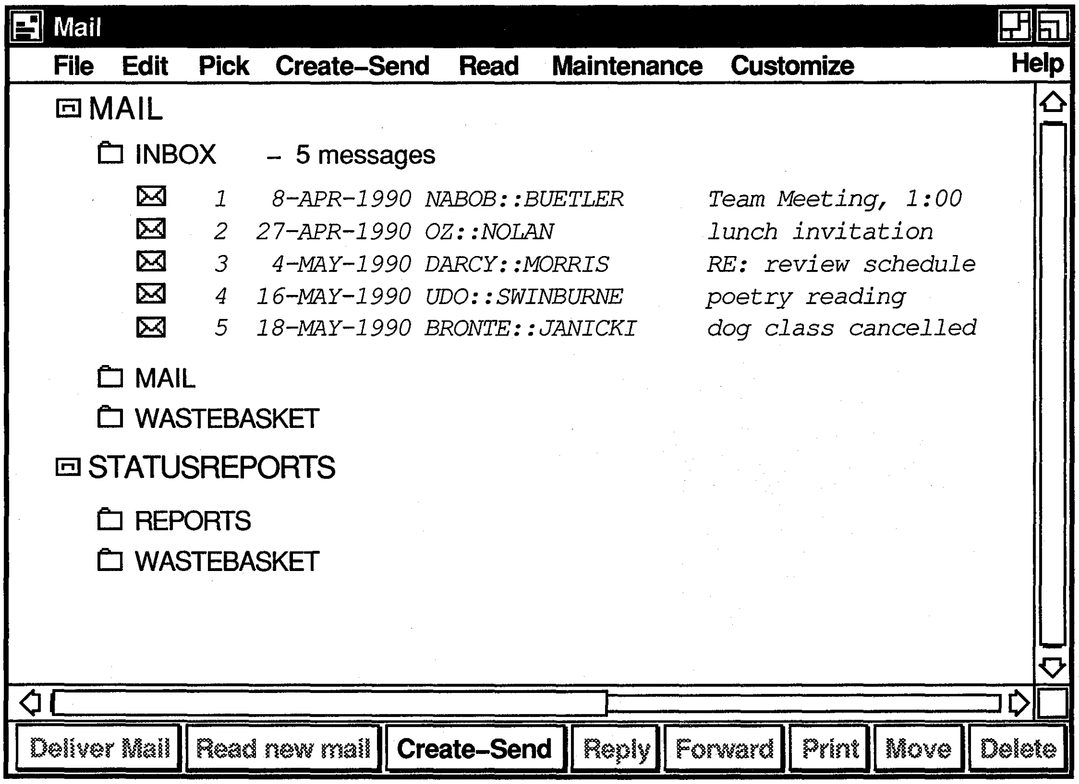
Using Mail, you can do the following:

- Send messages to any user on the system or network
- Read messages from other users
- Print messages
- Reply to messages
- File messages
- Forward messages

- Delete messages
- Send files
- Extract files from messages you receive

Figure 2–6 shows the main Mail window.

Figure 2–6 Main Mail Window



MAIL

INBOX – 5 messages

```
1   8-APR-1990 NABOB::BUETLER     Team Meeting, 1:00
2  27-APR-1990 OZ::NOLAN          lunch invitation
3   4-MAY-1990 DARCY::MORRIS      RE: review schedule
4  16-MAY-1990 UDO::SWINBURNE     poetry reading
5  18-MAY-1990 BRONTE::JANICKI    dog class cancelled
```

MAIL

WASTEBASKET

STATUSREPORTS

REPORTS

WASTEBASKET

ZK–0294A–GE

Notepad Text Editor

The Notepad text editor is a basic text editor that you can use to create and edit text files. The Notepad text editor also lets you perform the following functions:

- Search for text strings
- Move to specified line numbers
- Use a journaling feature to recover lost edits

Paint

Paint is an application that you use to create and save screen images. Using Paint is much like painting or drawing. You select a brush and trace designs. However, Paint makes the process easier. You can pick shapes from a menu and let paint draw the shape, and you can select fill patterns to shade or "color" your picture. To provide greater detail for part of your picture, you can use the zoom feature to magnify a specified area. If you are unhappy with your picture, you can erase it and start over. If you like the picture, you can print a copy.

Paint stores images as DDIF files. Applications that display graphics can use files that you create using Paint.

PostScript Previewer

The PostScript Previewer lets you display PostScript files on your workstation screen and page through the contents. Because the screen fonts correspond to fonts used by PostScript printers, you can see what a PostScript file looks like before you print it.

Puzzle

The Puzzle lets you solve a classic, sliding-tile puzzle on your workstation display.

DECterm

DECterm displays a window that looks and functions like a VT340 terminal. Applications written for VT52, VT100, VT220, VT240, VT320, and VT340 terminals run without modification in the DECterm window.

CDA File Converters

DIGITAL's Compound Document Architecture (CDA) specifies a framework for storing and interchanging compound documents. Compound documents are files with multiple elements (for example, text and graphics). The CDA defines a common format for compound documents: the DIGITAL Document Interchange Format (DDIF). You can use CDA converters to translate files to different formats. The converters accept an input file, convert it to an in-memory DDIF format, then produce an output file of the specified format.

The DCL command CONVERT/DOCUMENT invokes the CDA converters.

Online Help

DECwindows provides online help for the user environment. For most applications, you get help from a pull-down menu; DECwindows displays a general help screen from which you can request more specific help. Some applications also provide context-specific help. For these applications, DECwindows displays help that is relevant to the window you are using.

3 The Programming Environment

DECwindows offers a rich environment for developing applications for graphic workstations. This chapter provides an overview of the programming environment, describes the libraries and tools available to the DECwindows application programmer, and summarizes the structure of DECwindows applications.

Overview of the Programming Environment

The DECwindows programming environment makes it easy to develop applications with simple, consistent, graphics-oriented interfaces. The core of the graphics programming environment consists of the Xlib and XUI Toolkit programming libraries. Xlib consists of low-level routines for performing basic graphic and windowing functions. The XUI Toolkit consists of high-level routines for creating and managing user interface objects like menus, scroll bars, and buttons. Applications written in any programming language can call routines from both libraries.

By using XUI Toolkit routines, you save time, because XUI Toolkit routines simplify the task of creating a user interface. For example, you can create a menu with a call to one XUI Toolkit routine. Creating the same menu using Xlib routines requires many more calls and program lines. Using XUI Toolkit routines also ensures that an application interface conforms to the XUI style. Applications that conform to the XUI style are easy to learn and use. The XUI style is fully defined and illustrated in the *XUI Style Guide*.

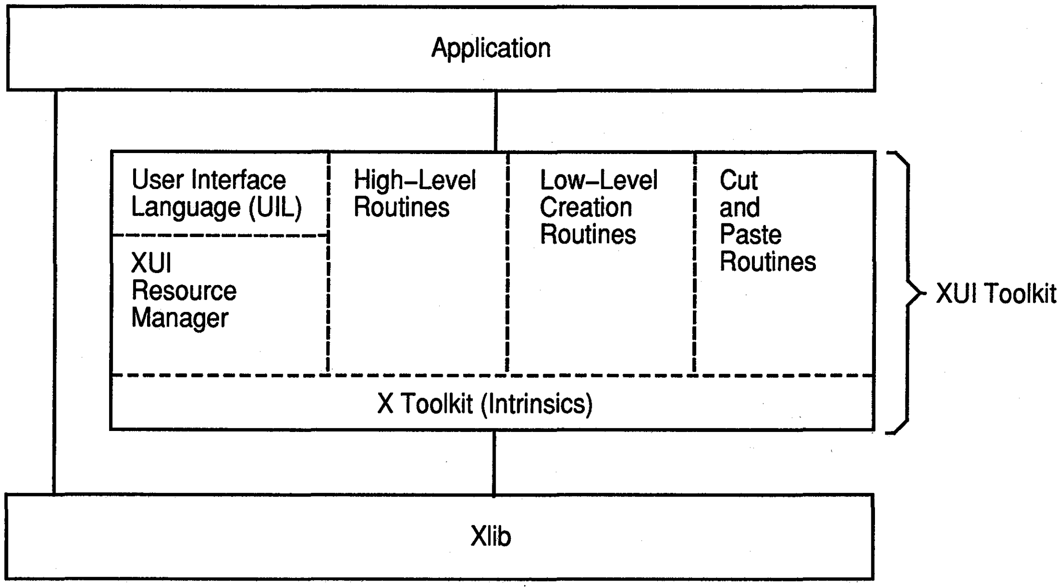
XUI Toolkit routines simplify the creation of a user interface. However, the XUI Toolkit includes additional tools that simplify the process further: the User Interface Language (UIL) Compiler and the XUI Resource Manager (DRM) routines. In essence, UIL and DRM let you create the entire interface with one library call. UIL and DRM also let you separate form and function in an application. You can store the user interface in a file that is independent from the application, and modify the file without recompiling the rest of the application.

Creating and Managing Windows

Using DECwindows library routines, applications create and manage windows. A window is an interface object that displays output or accepts input for an application. From the user's perspective, an application usually consists of a series of windows. For example, a spread-sheet application might display a main window that contains several menus. Each menu selection might display another menu or interface object. The user sees a series of interface objects that are logically connected. To create a set of interface objects for an application, you use library routines to construct and manage a hierarchy of windows.

Figure 3–1 illustrates the components of the DECwindows programming environment. The sections following describe the components.

Figure 3–1 DECwindows Programming Environment



ZK–0084A–GE

Xlib

The Xlib programming library is DIGITAL's implementation of the X Window System's C language, graphics programming library. DIGITAL has extended Xlib in the following ways:

- Added support for additional programming languages
- Integrated Xlib with the VMS operating system
- Added DECnet and shared memory transport support

Xlib is the lowest level programming interface to the DECwindows environment. Applications can call Xlib routines from a variety of programming languages to perform basic windowing and graphics functions.

Xlib routines can be organized according to function. Table 3–1 lists and describes the functional categories of Xlib routines.

Table 3–1 Functional Categories of Xlib Routines

Category	Function
Display routines	Connect and disconnect applications to the server, send and retrieve data to and from the server, and obtain default information about the display
Window routines	Create, map, move, change, and destroy windows
Window property routines	Create, use, and destroy window properties
Window clipping routines	Create, use, and destroy clipping regions within a window
Management routines	Manage windows, displays, and sessions
Event and error handling routines	Select and check input/output events; handle error conditions
Pixmap and bitmap routines	Create and free pixmaps and bitmaps
Graphics context routines	Create, change, and delete graphics attribute structures and attributes
Color routines	Manipulate color maps; allocate, define, and use colors
Graphics routines	Draw, fill, and erase points, lines, polygons and arcs
Text routines	Manipulate text in a window
Font routines	Specify, load, and free fonts
Cursor routines	Create, use, and delete cursors

You can call Xlib routines from an application using the standard VAX format or the MIT C format.

For more information about Xlib and Xlib routines, see the *VMS DECwindows Xlib Routines Reference Manual*, the *VMS DECwindows Guide to Xlib Programming: MIT C Binding*, and the *VMS DECwindows Guide to Xlib Programming: VAX Binding*.

XUI Toolkit

From a programming perspective, the XUI Toolkit is the key to the DECwindows environment. XUI Toolkit routines provide a straightforward way to create a user interface that conforms to the XUI style. The XUI Toolkit also provides development tools that allow you to separate form and function in an application; you can modify an interface without having to recompile and relink the application.

For more information about the XUI Toolkit, see the *VMS DECwindows Toolkit Routines Reference Manual* and the *VMS DECwindows Guide to Application Programming*.

Using XUI Toolkit Routines

XUI Toolkit routines allow you to create **widgets**. From a user's perspective, a widget is an interface object that displays output or accepts input. From a programmer's perspective, a widget is a window that is logically connected to application functions. When a user interacts with a widget (for example, by making a menu selection), information in the widget makes the application respond appropriately. Widgets can be simple interface objects, such as labels and buttons, or complex objects like menus and scroll bars.

Creating widgets in an application requires specification of the following:

- The hierarchy of widgets

  For example, an interface object might consist of a box with buttons inside the box. Both the box and buttons are widgets; specifying the hierarchy of widgets entails establishing the relationship between the box and buttons.

- The characteristics of each widget

  For example, you specify the height, width, and position of a widget.

- The routines your application executes when a user provides input to the interface

XUI Toolkit routines also let you create **gadgets**. Gadgets are similar to widgets with the following exceptions: gadgets are more functionally limited, but give applications better performance. Gadgets improve performance by using less memory.

Table 3–2 lists the widgets available in the XUI Toolkit and describes the function of each.

Table 3–2 Summary of XUI Toolkit Widgets

Widget	Function
Boxes	
Attached dialog box	A variation of the dialog box widget that has special positional properties.
Caution box	A type of dialog box that presents a warning message to the user before an irreversible action is executed.
Dialog box	A box that contains other widgets.
File selection	A type of dialog box that queries the user for a file specification.
Help	A window that presents the user of an application with helpful information.
List box	A box that presents a list of choices. It includes a scroll bar to aid in viewing list items.
Message box	A type of dialog box that displays a message to the user.
Pop-up attached dialog box	A variation of the attached dialog box that is temporary.
Pop-up dialog box	A variation of the dialog box that is temporary.
Radio box	A box that presents the user with a short list of choices, only one of which may be selected at any one time.
Selection box	A box that presents the user with a longer list of choices, only one of which may be selected at a time.
Work-in-progress box	A type of dialog box that displays a "Work in Progress" message.
Buttons	
Pull-down menu entry	A menu that presents a list of choices that are not visible until selected from the menu bar.
Push button	A button that invokes an immediate action when selected.
Toggle button	A button that provides control of on or off state.

(continued on next page)

Table 3–2 (Cont.) Summary of XUI Toolkit Widgets

Widget	Function
Menus	
Menu	A menu that presents a list of choices to the user.
Menu bar	A bar that presents a list of choices arranged horizontally, each of which contains a set of pull-down menus.
Option menu	A menu that presents a "one of many" choice while displaying current status of choices.
Pop-up menu	A menu that "pops-up" at the current pointer position.
Text	
Text edit	A widget that allows lines of text to be edited.
Windows	
Command window	A window that serves as a command entry area.
Main window	A decorated window that serves as an application's main window.
Scroll window	A window that allows data within the window to be scrolled.
Window	A window that is undecorated.
Miscellaneous	
Label	Text that identifies a specific part of a window.
Scale	An object that allows the choice of a value from within a range of values.
Scroll bar	An object that shows position in a body of data that is being scrolled.
Separator	A bar or line that separates list or menu entries.

The XUI Toolkit includes gadgets for creating labels, push buttons, toggle buttons, and separators.

You can use the following components of the XUI Toolkit to create and manage widgets:

- High-level widget routines
- Low-level widget creation routines
- X Toolkit routines
- User Interface Language (UIL) Compiler
- XUI Resource Manager (DRM) routines

The XUI Toolkit also provides cut and paste routines to exchange data between widgets or applications.

The following sections describe the components of the XUI Toolkit.

High-Level Widget Routines

High-level routines let you create a widget hierarchy and specify initial attributes of a widget. High-level routines provide some control over widget attributes while ensuring that your application conforms to the XUI style.

Low-Level Widget Creation Routines

Low-level routines also provide ways to create a widget hierarchy and specify initial attributes of a widget. Compared to the high-level routines, low-level routines provide more control over widget attributes, and therefore give you more freedom to customize the widgets. However, low-level routines can be more difficult to use. Often, you must specify a lengthy argument list in calls to a routine.

X Toolkit Routines

X Toolkit routines, also called **intrinsics**, let you manage and manipulate widgets at run time. Using intrinsics, you can do the following:

- Control the location of widgets
- Map and unmap widgets to the screen
- Process input from the user of the application

User Interface Language (UIL) Compiler

The UIL compiler is a tool that simplifies the creation and customization of an application's user interface. Using UIL, you create a specification file that determines the form of the application—the user interface. You compile the specification file using the UIL compiler. The application's functional routines determine what happens when users interact with the interface. You compile the functional routines separately from the specification file. As a result, you separate form and function in an application.

The UIL specification file defines the following characteristics of the user interface:

- The widgets that comprise the interface
- The characteristics of the specified widgets
- The callback routines for each widget
- The hierarchy of widgets in the application

The UIL compiler provides the following benefits to application developers:

- You can create a variety of interfaces for a single application. For example, if you develop applications for international markets, you can customize the interface for each market.
- The UIL compiler detects a variety of errors that high-level and low-level XUI Toolkit routines do not detect. For example, the UIL compiler issues diagnostic messages if you specify the wrong type of value for an argument or if you specify an unsupported argument.
- UIL makes it easy to do prototyping of user interfaces for applications. You can quickly develop and test a variety of interfaces, even if the application's functional routines are unavailable.

For more information about UIL, see the *VMS DECwindows Guide to Application Programming* and the *VMS DECwindows User Interface Language Reference Manual.*

XUI Resource Manager (DRM) Routines

DRM is the facility that takes data from the compiled UIL specification file (called the UID database) and creates the widgets. DRM translates the information in the UID database into low-level widget creation calls. If you use UIL to create a user interface, your application must include a special set of routines that invoke DRM. Using DRM routines reduces the time required for applications to create widgets.

For more information about DRM, see the *VMS DECwindows Guide to Application Programming.*

Cut and Paste Routines

Cut and paste routines allow application users to easily transfer data between windows.

CDA Toolkit

The Compound Document Architecture (CDA) is DIGITAL's strategy for creating, storing, and interchanging compound documents. A compound document is a file that contains multiple elements such as proportionally spaced text, graphics, and scanned images. Using a component of the CDA called the converter, you can convert files to different formats, including a standard format called the DIGITAL Document Interchange Format (DDIF).

CDA Toolkit routines allow applications to create DDIF files, read DDIF files, and invoke the file converters.

For more information about the CDA Toolkit, see the *VMS Compound Document Architecture Manual.*

4

Overview of VMS DECwindows Documentation

The VMS DECwindows documentation set provides the following types of information:

- General user

  Describes the DECwindows end-user environment; explains how to use DECwindows applications.

- Programming

  Provides the information necessary to develop DECwindows applications.

The following sections describe the books in the DECwindows documentation set.

General User Documentation

The following manuals comprise the general user documentation:

- *VMS DECwindows User's Guide*

  The *VMS DECwindows User's Guide* describes the DECwindows user environment. The first chapter provides hands-on experience by taking users through a sample session. Subsequent chapters explain how to use FileView, manage files, and customize the DECwindows environment. The book also provides a quick reference that explains basic techniques used in the DECwindows environment, such as scrolling and selecting text.

- *VMS DECwindows Desktop Applications Guide*

  The *VMS DECwindows Desktop Applications Guide* describes, in a task-oriented manner, the DECwindows desktop applications. These applications include a mail handler, a text editor, a terminal emulator, a calendar, an application for drawing screen images, a clock, a cardfiler, a calculator, and game. A chapter is devoted to each application; users can read about one application at a time without having to read the entire book.

Programming Documentation

The following manuals comprise the programming documentation:

- *XUI Style Guide*

  The *XUI Style Guide* outlines standards for the development of DECwindows application interfaces. One of the major goals of the DECwindows product is to provide a user environment in which application interfaces are consistent. Using extensive illustrations of interface objects, the *XUI Style Guide* establishes guidelines that developers should follow to create applications that conform to the XUI style.

- *VMS DECwindows Xlib Routines Reference Manual*

  The *VMS DECwindows Xlib Routines Reference Manual* provides the following information:

  - A description of the documentation format for Xlib routines
  - Instructions for calling an Xlib routine using the VAX format and the MIT C format
  - A description of each argument in the routine
  - A description of how the routine functions
  - A list of values returned by the routine

- *VMS DECwindows Toolkit Routines Reference Manual*

  The *VMS DECwindows Toolkit Routines Reference Manual* provides the following information:

  - A description of the documentation format for XUI Toolkit routines
  - Instructions for calling an XUI Toolkit routine using the VMS format and the MIT C format

  - A description of each argument in the routine
  - A description of how the routine functions

- *VMS DECwindows Guide to Xlib Programming: MIT C Binding* and *VMS DECwindows Guide to Xlib Programming: VAX Binding*

  These books explain how to use Xlib routines in DECwindows applications. The books includes an overview of Xlib; a discussion of programming considerations in C, FORTRAN, ADA, and Pascal; and tutorials for using Xlib routines.

- *VMS DECwindows Guide to Application Programming*

  The *VMS DECwindows Guide to Application Programming* explains how to use the XUI Toolkit to develop applications. The book includes an explanation of the User Interface Language (UIL) compiler.

- *VMS DECwindows User Interface Language Reference Manual*

  The *VMS DECwindows User Interface Language Reference Manual* describes the syntax and features of the User Interface Language (UIL), a specification language for describing the initial state of a user interface for a VMS DECwindows application. The manual contains a description of the syntax of low-level elements and module components of UIL. It also contains information about how to use the UIL compiler and how to interpret compilation diagnostics.

- *VMS DECwindows Device Driver Manual*

  The *VMS DECwindows Device Driver Manual* provides reference information about the DECwindows device drivers. The book explains the architecture for DECwindows device drivers and provides the information necessary to write port and class drivers.

- *VMS Compound Document Architecture Manual*

  The *VMS Compound Document Architecture Manual* describes the concepts and tools associated with DIGITAL's Compound Document Architecture (CDA). This book includes documentation of the CDA Toolkit routines.

Index

Reader's Comments

Overview of VMS DECwindows
AA–MG17A–TE

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

Thank you for your assistance.

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

I would like to see more/less ______________________________

____

What I like best about this manual is ______________________________

____

What I like least about this manual is ______________________________

____

I found the following errors in this manual:
Page Description

_____ ______________________________

_____ ______________________________

_____ ______________________________

Additional comments or suggestions to improve this manual:

____

____

____

I am using **Version** _____ of the software this manual describes.
Name/Title ______________________ Dept. ____________

Company ______________________ Date ____________

Mailing Address ______________________________

______________________ Phone ____________

- Do Not Tear - Fold Here and Tape

digital™

No Postage
Necessary
if Mailed
in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

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

- Do Not Tear - Fold Here

Cut Along Dotted Line

Reader's Comments

Overview of VMS DECwindows
AA–MG17A–TE

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

Thank you for your assistance.

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

I would like to see more/less ______________________________

____

What I like best about this manual is ______________________________

____

What I like least about this manual is ______________________________

____

I found the following errors in this manual:
Page Description

_____ ______________________________

_____ ______________________________

_____ ______________________________

Additional comments or suggestions to improve this manual:

____

____

____

I am using **Version** _____ of the software this manual describes.
Name/Title ______________________ Dept. __________

Company ______________________ Date __________

Mailing Address ______________________________

______________________ Phone __________

– Do Not Tear - Fold Here and Tape

digital™

No Postage
Necessary
if Mailed
in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

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

– Do Not Tear - Fold Here

Cut Along Dotted Line