Getting Started With the New Desktop

Order Number: AA–QUW1A–TE

May 1996

This manual is designed to enable DECwindows Motif general users, system managers, and programmers to successfully use the New Desktop, the alternate desktop provided with DECwindows Motif for OpenVMS Alpha.

Revision/Update Information:	This is a new manual.
Operating System:	OpenVMS Alpha Version 6.2
Software Version:	DECwindows Motif Version 1.2–4 for OpenVMS Alpha

Digital Equipment Corporation
Maynard, Massachusetts

Contents

1 Overview of the New Desktop

2 Using the New Desktop

3 Managing the New Desktop

4 Integrating Applications Into the New Desktop

5 New Desktop Programming Resources

A New Desktop and CDE Differences

B Multiple-Screen Support

C CDE API Routines

Index

Figures

Tables

Preface

Intended Audience

This manual is designed for users of the New Desktop component of DECwindows Motif Version 1.2-4 for OpenVMS Alpha. It provides information for general users, system managers, and programmers.

Document Structure

This manual is for all types of users. Table 1 shows the chapters and their intended audience.

Table 1   Chapters and Intended Audience

Chapter	Title	Intended Audience
1	Overview of the New Desktop	All users
2	Using the New Desktop	All users
3	Managing the New Desktop	System managers and advanced users
4	Integrating Applications into the New Desktop	Programmers
5	New Desktop Programming Resources	Programmers

Related Documents

The documents related to this manual are shown in Table 2.

Table 2 Documents Related to This Manual

Document	Description
DECwindows Motif Version 1.2–4 for OpenVMS Release Notes	Provides new features as well as release notes.
DECwindows Motif Version 1.2–4 for OpenVMS Installation Guide	Provides a detailed description of the installation procedure.
Guide to the DECwindows Motif Version 1.2–4 for OpenVMS CD–ROM	Includes a list of all the files on the kit, including the CDE manuals and this manual in HTML format.
CDE manuals	Developed by UNIX vendors. They are available on the kit in HTML format and also in print, as shown in Section 1.8.1.

For additional information on OpenVMS products and services, access the Digital OpenVMS World Wide Web site. Use the following URL:

```
http://www.openvms.digital.com
```

Reader's Comments

Digital welcomes your comments on this manual.

Print or edit the online form SYS$HELP:OPENVMSDOC_COMMENTS.TXT and send us your comments by:

Internet	**openvmsdoc@zko.mts.dec.com**
Fax	603 881-0120, Attention: OpenVMS Documentation, ZK03-4/U08
Mail	OpenVMS Documentation Group, ZKO3-4/U08 110 Spit Brook Rd. Nashua, NH 03062-2698

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 Caribbean, 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: 264–4446 603–884–4446	Fax: 603–884–3960	U.S. Software Supply Business Digital Equipment Corporation 8 Cotton Road Nashua, NH 03063–1260

ZK–7654A–GE

Conventions

The name of the OpenVMS AXP operating system has been changed to OpenVMS Alpha. Any references to OpenVMS AXP or AXP are synonymous with OpenVMS Alpha or Alpha.

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

The following conventions are also used in this manual:

MB1 (Select) MB3 (Menu)	MB1 indicates the left mouse button, and MB3 indicates the right mouse button.
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.

<kbd>Return</kbd>	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 indicate 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.
	Boldface text is also used to show user input in Bookreader versions of the manual.
italic text	Italic text indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (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.

`Monospace type`	Monospace type indicates code examples and interactive screen displays.
	In the C programming language, monospace type in text identifies the following 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 at the end of a command format description, command line, or code line indicates that the command or statement continues on the following line.
numbers	All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated.

1

Overview of the New Desktop

This chapter provides information about the following topics:

- What is CDE? (Section 1.1)
- Why is Digital offering CDE? (Section 1.2)
- What is the New Desktop? (Section 1.3)
- Familiarizing yourself with the New Desktop (Section 1.4)
- New Desktop productivity features (Section 1.5)
- User applications (Section 1.6)
- Application programming interfaces (APIs) and application integration services (Section 1.7)
- Documentation for the New Desktop (Section 1.8)

1.1 What is CDE?

The Common Desktop Environment (CDE), based on Motif and X11, is recognized as the next-generation desktop. CDE was created by a group of UNIX vendors (Hewlett-Packard, Sun, IBM, and Novell) to consolidate all the UNIX desktop interfaces and to define a consistent user and development environment.

CDE is built on existing technologies from these UNIX vendors and from the Open Software Foundation, Inc. (OSF). CDE is not a replacement for Motif; rather, it is another software layer on top of Motif, as shown in Figure 1–1. CDE provides additional programming utilities and a consistent, integrated user and development interface.

1.2 Why is Digital Offering CDE?

Digital adopted CDE for its Digital UNIX operating system and is providing a major subset of CDE for OpenVMS Alpha. As a successor to the DECwindows desktop, CDE is a state-of-the-art graphical user interface. This interface is visually appealing and offers many productivity features.

Standardizing the user interface enables general users and system managers to work more effectively on systems from all vendors that provide CDE. It also reduces training costs. Standardizing the development environment for heterogeneous systems facilitates the creation and maintenance of applications.

1.3 What is the New Desktop?

The implementation of CDE on OpenVMS Alpha is called the New Desktop. The user environment provided with previous releases of DECwindows Motif is the DECwindows desktop. DECwindows Motif forms the foundation of the New Desktop, as shown in Figure 1–1. Any application at the top level can directly call any of the lower levels.

Choosing the New Desktop does not prevent you from running DECwindows Motif applications. You can run any DECwindows application from the New Desktop using a DECterm window. Furthermore, all DECwindows applications can be configured to run within the New Desktop environment.

On OpenVMS Alpha, the New Desktop is the default desktop, although the traditional DECwindows desktop is also available. During installation or from the New Desktop login box, you can select your preferred desktop. After installation, a privileged user can switch between the two desktops by editing a system setup file and restarting DECwindows (see Section 3.2).

Figure 1–1 The New Desktop Architecture

```
┌─────────────────────────────────────────────────────────┐
│              CDE and DECwindows Motif Applications        │
└─────────────────────────────────────────────────────────┘

┌──────────┐  ┌──────────┐  ┌──────────┐  ┌──────────────┐
│ CDE      │  │ CDE Drag"│  │ CDE      │  │ Digital      │
│ Help     │  │ and"Drop │  │ Custom   │  │ Extensions to│
│ Services │  │ Services │  │ Widgets  │  │ Motif (DXm)  │
└──────────┘  └──────────┘  └──────────┘  └──────────────┘

┌─────────────────────────────────────────────────────────┐
│              Motif Toolkit Library (Xm)                   │
└─────────────────────────────────────────────────────────┘

┌─────────────────────────────────────────────────────────┐
│              X Toolkit Intrinsics (Xt)                    │
└─────────────────────────────────────────────────────────┘

┌─────────────────────────────────────────────────────────┐
│              X Library (Xlib)                             │
└─────────────────────────────────────────────────────────┘

┌─────────────────────────────────────────────────────────┐
│              DECwindows Transport                         │
└─────────────────────────────────────────────────────────┘

┌──────────┐  ┌──────────┐         ┌──────────┐
│ TCP/IP   │  │ DECnet   │         │ Local    │
│ Transport│  │ Transport│         │ Transport│
└──────────┘  └──────────┘         └──────────┘
        Ethernet
◄──────────────────►                    │
                                        ▼
   To X Display Server             To X Display Server
```
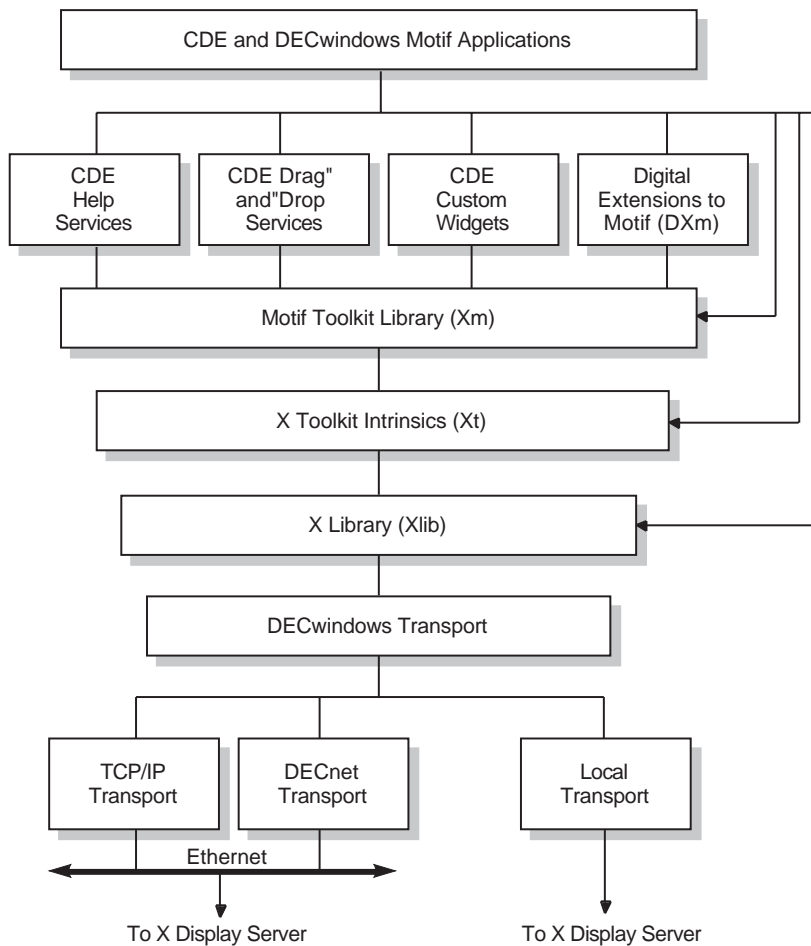
ZK-8648A-AI

1.4 Familiarizing Yourself With the New Desktop

You can become familiar with the New Desktop by using the documentation and applications in the following sequence:

1. Read Chapters 1 and 2 of this manual and, optionally, Chapters 3, 4, and 5, as appropriate to your work.

2. Use the online help, "Introducing the CDE Desktop," especially "A Quick Tour of the Desktop" (also known as "The CDE Desktop at a Glance"), available from the Front Panel.

   This online help is displayed when you first log in to the New Desktop and is also accessible from the Help icon on the Front Panel.

3. Try one or more of the new CDE applications and refer to its help or online manual.

1.5 New Desktop Productivity Features

The New Desktop provides many features not available in DECwindows Motif that enable users to be more productive in their work. Some of the most important productivity features follow.

Figure 1–2 Front Panel

ZK–8649A–GE

- Front Panel

  The Front Panel, as shown in Figure 1–2, provides fast access to applications. From the Front Panel, you can select one of your customized workspaces or launch any application that is represented by an icon. You do this with a *single* click—on the workspace icon or the application icon. The same Front Panel appears in every workspace.

  Three of the icons on the Front Panel have subpanels. The existence of a subpanel is indicated by an arrow above the icon. A single click on the arrow produces a short menu from which you can make a selection. When a subpanel is displayed, the arrow points downward. When you are finished with the menu, you can remove it with a single click on the same arrow.

  You can customize many aspects of the Front Panel, such as the icons, the names of the workspaces, and the applications listed on the subpanels.

  The Front Panel provides many of the functions provided by the DECwindows Session Manager. You can select applications by clicking on their icons, pause your session by clicking on a lock icon, and exit by clicking on the Exit button. The Front Panel is part of the New Desktop's Window Manager.

- Workspaces

  A workspace is a *virtual screen*. When a workspace is selected, only those applications included within the workspace are displayed. Multiple workspaces allow you to organize your work into groups of applications that you typically use together. For example, one workspace could be used for project management tasks, another for communicating on the Internet, and another for financial management.

  Workspaces can be added, deleted, and renamed dynamically. The default number of workspaces is 4 (see Figure 1–2), which can be increased to 64. The number of workspaces may be limited by your system resources.

- Icon-based File Manager

  The icon-based File Manager, combined with the drag-and-drop feature, enables you to manage files quickly and conveniently. File Manager also supports additional features for two special types of top-level windows: application views (via Application Manager) and Trash Can.

  File Manager allows easy navigation of a directory hierarchy. It includes several viewing options, such as a tree format and a view with expanded information, including file sizes and revision dates. You can filter files by file type and sort them by several criteria.

- Automatic save and restore of open applications in each workspace

  When you log out, the state of your session is saved. The next time you log in, every application that supports automatic save and restore that was open when you logged out will be open again. Alternatively, you can save a snapshot of your session at any time and restore to that point when you log in. All the applications included with the New Desktop provide this feature.

Note

The level of detail that is automatically saved and restored depends on the application. Some applications restore the entire visual state of the application at the point it was saved, while others restore only the main screen of the application.

1.6 User Applications

Table 1–1 lists all the CDE and DECwindows user applications that are integrated into the New Desktop and indicates the following information about each application:

- Application name

- Executable name

- CDE application or DECwindows application

- Whether it is accessible from the preconfigured Front Panel

In addition to the user applications that are integrated into the New Desktop, all DECwindows applications that were included in DECwindows Motif Version 1.2–3 are part of the New Desktop and are accessible from Application Manager. Note that two help viewers are provided with the New Desktop: Help Viewer for New Desktop help and Bookreader for DECwindows help.

Table 1–1   User Applications and Services

Application Name	Executable Name	CDE or DECwindows	Accessible From Front Panel
Action database and execution management utility (Create Action)	dtcreate	CDE	No
Action utility	dtaction	CDE	No
Application Manager, File Manager	dtfile	CDE	Yes
Bookreader	DECW$BOOKREADER	DECwindows	Yes[1]
Calculator	dtcalc	CDE	No
Calendar	DECW$CALENDAR	DECwindows	Yes
Clock	DECW$CLOCK	DECwindows	No[2]
DECterm	DECW$TERMINAL	DECwindows	Yes
Front Panel, Workspace Manager	dtwm	CDE	—

[1]Accessible from the Help subpanel.

[2]The clock on the Front Panel is an animated icon. You cannot access the Clock application by clicking on this icon. Instead, you can access it from the DECwindows Apps group in Application Manager.

Table 1–1 (Cont.)   User Applications and Services

Application Name	Executable Name	CDE or DECwindows	Accessible From Front Panel
Help Print utility	dthelpprint	CDE	No
Help Viewer	dthelpview	CDE	Yes
Icon Editor	dticon	CDE	Yes[3]
Mail	DECW$MAIL	DECwindows	Yes
Print Dialog	PRINTDIALOG	DECwindows	Yes[4]
Session Manager	dtsession	CDE	No
Style Manager	dtstyle	CDE	Yes
Text Editor	dtpad	CDE	Yes[3]
Trash Can	dtfile	CDE	Yes
Watch Errors	DECW$MESSAGEPANEL	DECwindows	No

[3]Accessible from the Personal Applications subpanel.

[4]Based on the DECwindows print widget.

1.7 APIs and Application Integration Services

A subset of CDE application programming interfaces (APIs) and application integration services is available with the New Desktop (see Section 5.2).

In addition to these new APIs and application integration services, all the DECwindows Motif APIs and application integration services are available from the New Desktop.

1.8 Documentation for the New Desktop

The following types of documentation are included in the DECwindows Motif kit to support the New Desktop:

- Online introduction in Help

- Online reference pages (also known as manpages)

- Help for all CDE applications

- Help for all DECwindows applications

- CDE documentation set in HTML format

- DECwindows Motif documentation set (most manuals) in Bookreader format

- Printed copies of this manual and selected DECwindows Motif manuals

See the *Guide to the DECwindows Motif Version 1.2–4 for OpenVMS CD–ROM* for a list of all manuals provided on line. The CDE documentation set that is provided on line is also available in print. The same is true for the entire DECwindows Motif documentation set.

This manual and the *Guide to the DECwindows Motif Version 1.2–4 for OpenVMS CD–ROM* are provided in printed format with DECwindows Motif Version 1.2–4. Additional DECwindows Motif printed documentation is available through service contracts or by ordering from DECdirect, as described in the Preface of this manual. Table 1–2 briefly describes the documentation and how to access or order it.

For more information about the documentation, see the Bookreader versions of *Overview of DECwindows Motif for OpenVMS Documentation* and the *DECwindows Motif Version 1.2–4 for OpenVMS Release Notes*, as described in this table.

Table 1–2   Documentation for the New Desktop

Documentation	Description	How to Access
Online Help		
Front Panel Help	Overview help for the CDE desktop; access to the help for all CDE applications in the New Desktop; and access to Bookreader	Click on the Help icon on the Front Panel to display the Help Manager, or display the Help subpanel and select an introductory topic, Help Manager, or Bookreader. From the Help Manager, you can select introductory topics and CDE application help.
CDE On Item Help	Brief description of icons and controls on Front Panel and within applications	Click on On Item Help, then position the cursor on the icon or control for which you want help and click; repeat these two steps for each request for On Item help.
Application help (CDE and DECwindows)	Help menu for each application, including context-sensitive help for objects within an application	Start the application and select Help from the menu bar; then make your selection from the Help menu.

(continued on next page)

Table 1–2 (Cont.)   Documentation for the New Desktop

Documentation	Description	How to Access
Online Manuals and Reference Pages		
CDE online manuals	Documents prepared by the UNIX vendors who developed CDE	Use the Enhanced Mosaic web browser, as described in Section 1.8.2.
CDE online reference pages	Reference documentation for many CDE system commands, written by the UNIX vendors who developed CDE	Select the reference page directory[1] from File Manager; then select the file you want to view. From Application Manager, select Desktop Apps; then select Man Page Viewer; and enter the name of the file you want to view.
DECwindows Motif online manuals	Digital documents	Click on the arrow above the Help icon on the Front Panel to display the Help subpanel; then select the Bookreader icon, then select a manual.[2]
Printed Manuals		
CDE printed manuals	Documents prepared by the UNIX vendors who developed CDE (with Digital UNIX covers)	See Section 1.8.1 for the order numbers; then order them by following the instructions in this Preface.
DECwindows Motif printed manuals	This manual, the *DECwindows Motif Version 1.2–4 for OpenVMS Installation Guide*, and the *Guide to the DECwindows Motif Version 1.2–4 for OpenVMS CD–ROM* are shipped with the software.	See the online version of the *Overview of DECwindows Motif for OpenVMS Documentation* for brief descriptions and order numbers; then order them by following the instructions in this Preface.

[1]The directory name is CDE$SYSTEM_DEFAULTS:[MAN].

[2]To use this selection method, the Bookreader system logical, DECW$BOOK, must point to the directory where the Bookreader files reside.

1.8.1 CDE Manuals

The CDE manuals and their Digital order numbers are listed in Table 1–3. You can order them from Digital, as described in the Preface of this manual.

Table 1–3 CDE Manuals With Order Numbers

Title	Order Number	Pages
Common Desktop Environment: User's Guide	AA-QTLNA-TE	374
Common Desktop Environment: Advanced User's and System Administrator's Guide	AA-QTLQA-TE	324
Common Desktop Environment: Programmer's Overview	AA-QTLWA-TE	118
Common Desktop Environment: Programmer's Guide	AA-QTLXA-TE	204
Common Desktop Environment: Help System Author's and Programmer's Guide	AA-QTLYA-TE	320
Common Desktop Environment: Internationalization Programmer's Guide	AA-QTM2A-TE	166
Common Desktop Environment: Style Guide and Certification Checklist	AA-QTM3A-TE	316
Common Desktop Environment: Product Glossary	AA-QTM4A-TE	144

1.8.2 Using Enhanced Mosaic to Access Manuals

To access this manual or any of the CDE manuals provided in HTML format on the kit, use the Enhanced Mosaic web browser in the following way:

1. Start the Application Manager and click on the DECwindows Applications Group icon.

2. Click on the Mosaic icon.

3. From the File menu, choose the Open Local... item.

4. In the Selection field, enter the following file specification:

    ```
    device_name:[DOCUMENTATION.HTML]INDEX.HTML
    ```

    where *device-name* is the name of the drive on which you have placed your New Desktop CD–ROM. For example:

    ```
    DKB700:[DOCUMENTATION.HTML]
    ```

5. The set of titles is displayed; click on the title of the manual that you want to open.

2

Using the New Desktop

This chapter provides information about the following topics:

- CDE user applications in the New Desktop (Section 2.1)

- Accessing DECwindows applications (Section 2.2)

- Making applications more accessible (Section 2.3)

- CDE user documentation (Section 2.4)

2.1 CDE User Applications in the New Desktop

The CDE user applications that are included in the New Desktop are briefly described in this section. The focus of these descriptions is on the differences that users accustomed to the DECwindows desktop applications might experience. Differences between CDE on UNIX systems and the New Desktop are described in Appendix A.

2.1.1 Login Manager

The login manager provides login and authentication processing for the New Desktop in a manner similar to the login box provided with the traditional DECwindows desktop. In addition to user authentication, it also provides several options, as shown in Figure 2–1.

From the Options pull-down menu you can select:

- Language of the user interface

- Type of session you want

- Whether to reset the login screen

You can select from the following session types:

- New Desktop session

Figure 2–1 Login Dialog Box

Welcome to BRUBEK

Please enter your user name

| OK | Start Over | Options | Help |

Language

Session ▷ ⃝ New Desktop

Reset Login Screen | Failsafe Session

DECwindows Session
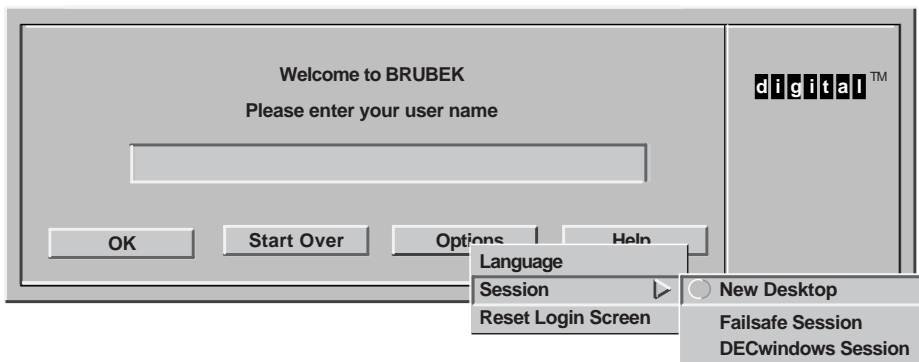
ZK–8688A–GE

New Desktop is the default environment.

- Failsafe session

  A failsafe session is a limited, single-DECterm window environment that allows you to correct startup problems in other environments.

- DECwindows session

  A DECwindows session is the traditional DECwindows environment.

Resetting the login screen is useful if you want to take advantage of customizations made to the login screen or, in an X terminal environment, if you want to remove the login box and terminate login processing.

2.1.2 Front Panel

The Front Panel, shown in Figure 2–2, provides quick access to applications and many other features. This section describes some aspects of the Front Panel that may not be immediately apparent to users accustomed to the DECwindows desktop. Here are some details to help you get started.

- Moving the Front Panel

  Position the cursor on the left or right border and press MB1 to move the Front Panel. The Front Panel can be moved up and down but cannot be rotated.

- A *single* click on any application icon (except Clock) starts the application.

Figure 2–2 Front Panel Details
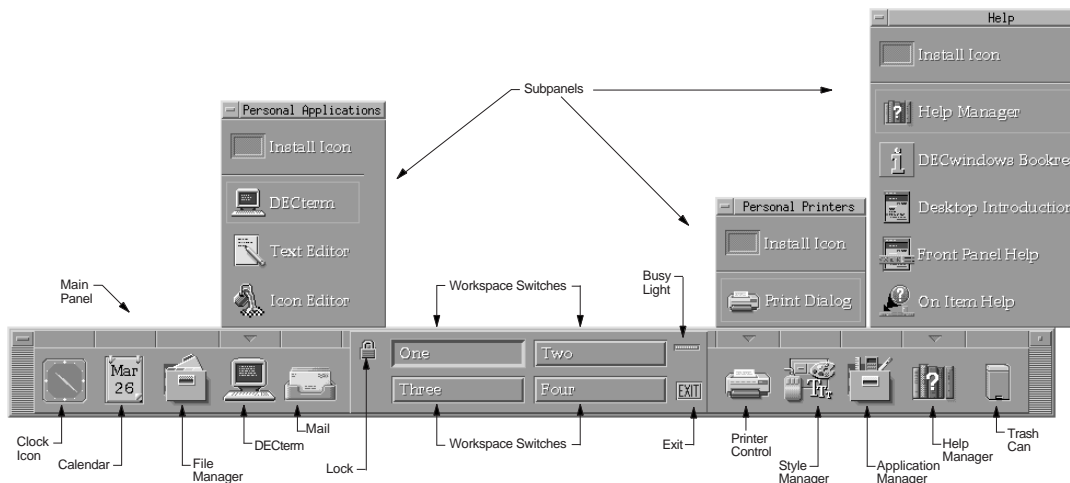
ZK–8650A–GE

- Getting information about each icon

  Position the cursor on any icon and press mouse button 3 (MB3) to either
  get help about the icon or add or delete a subpanel.

- Clock icon

  The clock displays the current time. However, unlike the other Front Panel
  controls, the clock is not a separate application. It is part of the Window
  Manager and can be customized by editing the date control definition in the
  Front Panel configuration file. (Refer to the *Common Desktop Environment:
  Advanced User's and System Administrator's Guide* for instructions.) The
  DECwindows Clock application can be accessed from Application Manager's
  DECwindows Apps group.

- Arrows above icons

  The arrows indicate that a subpanel, or menu, is available. Position the
  cursor on any arrow and single click. A subpanel is displayed with an
  Install Icon item that enables you to add applications to the menu. To

close the subpanel, position the cursor on the same arrow, which now faces down, and single click on it.

- Moving a subpanel

  You can move any subpanel with mouse button 1 (MB1). Position the cursor on the subpanel menu bar, press and hold MB1, drag the subpanel to a new location, and release MB1.

- Relationship of Front Panel to workspaces

  You can have as many as 64 different workspaces. The Front Panel is the same for all workspaces and is displayed on each workspace.

- Relationship of applications to workspaces

  You can put the same application in more than one workspace. For example, you can put the Text Editor in two workspaces and the clock in every workspace. You do this by selecting either Occupy Workspace... or Occupy All Workspaces from the Window menu, shown in Figure 2–3. (You can access an application's Window menu by clicking on the control in the top left corner of its main window.)

- Managing workspaces

  Pressing MB3 on a workspace control button displays a menu that enables you to add, delete, or rename workspaces.

Figure 2–3 Window Manager Menu
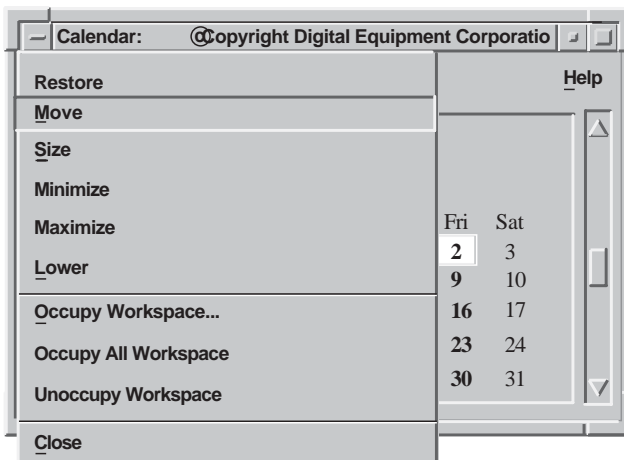
Calendar:	@Copyright Digital Equipment Corporatio		
Restore			Help
Move			
Size			
Minimize			
Maximize		Fri	Sat
Lower		2	3
		9	10
Occupy Workspace...		16	17
Occupy All Workspace		23	24
Unoccupy Workspace		30	31
Close			

ZK–8687A–GE

2.1.3 Application Manager

Application Manager is a repository for applications and tools available on a system. It supports application groups—those provided with the New Desktop (shown in Table 2–1) and those you create (see Section 4.4).

To view the Application Manager window, single click on the Application Manager icon, shown in Figure 2–2. From the Application Manager screen, you can select a group by double clicking on its icon. A screen with icons representing each application in the group is displayed. From this screen, you can start any application by double clicking on its icon.

Table 2–1 Preconfigured Application Groups

Group	Contents
DECwindows Apps	Applications such as DECwindows Mail, Calendar, FileView, Cardfiler, and all other DECwindows applications supplied with DECwindows Motif prior to Version 1.2–4
DECwindows Utilities	Most of the standard X utilities
Desktop Apps	CDE applications
Desktop Tools	Tools for common user tasks, such as comparing files, showing logical names, and viewing images

For more information about Application Manager, see its online help and the *Common Desktop Environment: User's Guide*.

2.1.4 Calculator

The New Desktop's Calculator application offers the following features:

- Three modes: Scientific (default), Logical, and Financial.

- Three trigonometric bases in Scientific mode.

- Four numeric bases: decimal, octal, hexadecimal, and binary.

- Ten memory registers for storing values, for use in each mode. The values can be recalled and exchanged between a specified register and the display area.

You can access the New Desktop's Calculator from the Personal Applications subpanel. For more information about it, access its help.

The DECwindows desktop Calculator is still available. You can access it from the DECwindows Apps group in Application Manager.

2.1.5 File Manager

File Manager presents an iconic representation of files. It allows intuitive user actions, such as dragging and dropping files. It also allows files to be placed on the background of the screen for fast access. You can open several views of different directories or of the same directory.

2.1.5.1 File Representation

File Manager supports four different representations of files:

- Large icons

- Small icons

- File name only

- Hierarchical tree view

You can choose the representation you prefer. If you frequently navigate up and down your directory tree, you may prefer the hierarchical directory view.

_____ **Note** _____

All executable programs and all command procedures, when invoked from File Manager by clicking on files with a .EXE or .COM extension, will be started with an associated DECterm window to capture output. A DECterm window is created regardless of the type of executable program or command procedure. If you do not want the DECterm window to appear, which is usually the case, you must create a new action definition file and an associated action (stub) file for invoking the executable or the command file. See Section 4.1 for information about creating action definition files and action files.

_____

2.1.5.2 File Filtering and Sorting

Files can be filtered and sorted by several criteria. To set filtering options, select View from the File Manager menu bar; then, from the View menu, select Set Filter Options. Filters for various data types can be turned on or off. For example, you can choose to view only files of the type COM_FILE or all files except files of the type COM_FILE (see Section 3.4.2.1). The filtering options also allow filtering by an explicit file type (for example, .RDB) and filtering by file name versions. You can choose to see all versions of your files or only the highest version.

To set sorting criteria, select View from the File Manager menu bar, then select Set View Options. You can choose to sort alphabetically, by data type, by modification date, or by size.

For more information about the capabilities of File Manager, see its online help, the *Common Desktop Environment: User's Guide*, and the *Common Desktop Environment: Administrator's Guide*.

2.1.6 Trash Can

File Manager supports dropping files on the Trash Can icon. Files that are moved to the Trash Can are not actually deleted and are still counted in your disk quota. To delete a file, you must shred the contents of the Trash Can. If the Trash Can contents have not been shredded, the files can be moved back to their original directory.

2.1.7 Help Viewer

Help Viewer, accessible from the Front Panel (see Figure 2–2) and from every CDE application, presents the help for all CDE applications within the New Desktop. In addition to application help, several other types of help are available, including Help on Help, Desktop Introduction, and On Item Help.

To use On Item Help, select it from the Help subpanel, position the cursor on the item for which you want help, and single click. Help is displayed in a full-size window.

For more information about the Help Viewer, see the online help and the *Common Desktop Environment: User's Guide*.

2.1.8 Icon Editor

The Icon Editor, accessible from the subpanel above the DECterm icon, enables you to create new icons or edit existing ones. You can resize icons, add hotspots to icons, and grab screen images and edit them.

For more information about Icon Editor, see its online help and the *Common Desktop Environment: User's Guide*.

2.1.9 Print Dialog

The printer icon on the Front Panel (see Figure 2–2) is configured to access the default printer for the desktop. The default printer is defined by the value of the SYS$PRINT logical on the system.

You can select different print job options for the default printer by using the Print Dialog. To access the Print Dialog, click on its icon on the Front Panel.

To print a file using the default printer and the options specified for it, drag the icon of the file you want to print and drop it on the Print Dialog icon.

If you would like to use a different printer as your default printer, see Section 3.4.5.

2.1.10 Text Editor

The Text Editor, accessible from the subpanel above the DECterm icon, is a full-function editor. It offers the standard text editing functions as well as more advanced features, including the drag-and-drop function. The drag-and-drop function can be used for moving text, copying text, and including entire files in another file.

If you would like to use a different editor as your default editor, see Section 3.4.6.

2.1.11 Session Manager

Session Manager functions are available from the Startup menu within Style Manager. From the Startup menu, you can select the application configuration of your next session. The Startup menu options are shown in Table 2–2.

Table 2–2 Startup Menu Options

Option	Description
Resume current session	The next time you log in, your session will start up where you left off when you logged out—as close as possible to the current configuration.
Return to home session	The next time you log in, your session will start up with a configuration that you have defined as your home session. If you have not defined a home session, the system default session is used.
Ask me at Logout	When you log out, you will be prompted to return to your current session or your home session.
Logout Confirmation Dialog	You can specify whether you want to be prompted to confirm your intent to log out.
Set Home Session	You can select your current session as your home session. You can do this at any time during a session. Customizations that you made during the session are included.

2.1.12 Style Manager

Style Manager enables you to control the following components of your desktop environment:

- Font size
- Backdrops (for each workspace)
- Keyboard attributes, including the language variant and clicking characteristics

- Mouse attributes

- Beep characteristics

- Screen characteristics, including blanking and backdrop to be used when the screen is locked

- Window behavior, including how to make windows active (by pointing or clicking)

- Startup attributes, as described in Section 2.1.11

- Security, which establishes the authorized users of your desktop

2.2 Accessing DECwindows Applications

All the applications provided with the previous release of DECwindows Motif are included with the New Desktop. By default, some of them are accessible from the Front Panel and all of them are accessible from Application Manager. The applications that are accessible from the Front Panel are:

- Calendar

- Mail

- DECterm

You can also access DECwindows applications from a DECterm. All optional DECwindows applications (those purchased separately) can also be accessed from a DECterm.

The access methods are described in Table 2–3.

Table 2–3 How to Access DECwindows Applications

Location	Method
Front Panel	*Single* click on the icon or on the selection from a subpanel.
Application Manager	*Single* click on the Application Manager icon on the Front Panel; then double click on the icon for its group; then double click on its icon.
DECterm	Run the application by its system-defined name.

2.3 Making Applications More Accessible

You can make an application not already located in Application Manager or the Front Panel more accessible. You can do this by dragging its default icon (a lightning bolt with its file name displayed below it) from the File Manager display and dropping it on any of the following locations:

- New application group in Application Manager

- Front Panel, on the Install Icon control of the Personal Applications subpanel (a new control will be created in the subpanel)

- Desktop (the background of a workspace)

After you move an application icon to one of these locations, you can invoke it from there.

Although moving an icon in this way makes the application more accessible, it does not provide the same level of integration that is available with the applications that are part of the New Desktop. For example, the application does not have a unique icon, and each time you invoke the application, a new DECterm is created. You can select a different icon from the set provided with the New Desktop or create a new one; you can eliminate the DECterm if it is not needed; and you can supply any arguments you want specified automatically if you integrate the application into the New Desktop.

For information about minimal integration, see Section 4.1. For information about creating new application groups, see Section 4.4.

2.4 CDE User Documentation

The CDE components of the New Desktop are documented in an extensive help system and in CDE manuals, which are available in print (separately orderable; see Section 1.8.1). The CDE manuals (and this manual) are also in HTML format on the DECwindows Motif Version 1.2–4 for OpenVMS CD–ROM. You can view these manuals with the Enhanced Mosaic web browser that is included on the same CD-ROM (see Section 1.8.2).

The following CDE user manuals are provided:

- *Common Desktop Environment: User's Guide*

- *Common Desktop Environment: Glossary*

- *Common Desktop Environment: Advanced User's and System Administrator's Guide*

3

Managing the New Desktop

This chapter provides information about the following topics:

- DECwindows and the New Desktop startup sequence (Section 3.1)
- Changing the default desktop (Section 3.2)
- New Desktop directory hierarchy (Section 3.3)
- Customizing the environment and applications (Section 3.4)
- Using the New Desktop error logs (Section 3.5)
- Using colors in the New Desktop environment (Section 3.6)
- Using the New Desktop fonts (Section 3.7)
- Using X terminals and other remote displays (Section 3.8)
- Using multiple screens (Section 3.9)
- Managing processes created within the New Desktop (Section 3.10)
- CDE system management documentation (Section 3.11)

3.1 DECwindows and the New Desktop Startup Sequence

SYS$MANAGER:DECW$STARTUP.COM, the DECwindows startup command procedure, can start either the traditional DECwindows desktop or the New Desktop. On OpenVMS Alpha, the New Desktop is started by default.

SYS$MANAGER:CDE$STARTUP.COM is executed during the application startup phase of the New Desktop to perform the following steps:

1. Define a series of logical names used by the New Desktop.

2. Install several executables (login) and libraries used by the New Desktop.

Once login starts, login starts Session Manager, which in turn starts Window Manager for the session. For a complete description of the DECwindows login process, see *Managing DECwindows Motif for OpenVMS Systems*.

3.2 Changing the Default Desktop

You can change the default desktop from the New Desktop to the DECwindows desktop by modifying the SYS$MANAGER:DECW$PRIVATE_APPS_ SETUP.COM file and restarting the desktop.

To change the default desktop to the DECwindows desktop, perform the following steps:

1. If you do not already have a SYS$MANAGER:DECW$PRIVATE_APPS_ SETUP.COM file, create one by copying SYS$MANAGER:DECW$PRIVATE_ APPS_SETUP.TEMPLATE to a new file, SYS$MANAGER:DECW$PRIVATE_ APPS_SETUP.COM.

2. Define the global symbol DECW$START_NEW_DESKTOP as FALSE in SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM:

   ```
   $ DECW$START_NEW_DESKTOP == "FALSE"
   ```

3. Restart the desktop with the following command:

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

If the DECwindows desktop is running and you want to change the default back to the New Desktop, perform the same steps but define DECW$START_ NEW_DESKTOP as TRUE instead of FALSE.

If you want to be able to select either desktop when you log in, do not modify the global symbol DECW$START_NEW_DESKTOP. Instead, select the desktop you want when you log in, using the Login Manager's Options menu, as described in Section 2.1.1.

3.3 New Desktop Directory Hierarchy

The New Desktop uses a fairly complex directory hierarchy for all of the files required for its environment. To traverse these directories and access these files, the New Desktop uses search paths to indicate the location and search order of these files for various operations. These search paths are logical names that contain a list of directory specifications and are defined when the New Desktop is started.

In general, three locations are searched for any data file. The locations and the order in which they are searched are shown in the following list:

1. *disk*$:[*user*.DT...]

   User-specific directory in the user's home directory.

2. CDE$USER_DEFAULTS:[*...] (customizable)

Systemwide directory that can be customized by the system manager.

3. CDE$SYSTEM_DEFAULTS:[*...] (noncustomizable)

Systemwide directory where the default copies of all of the files are placed at installation time.

This means that files found in a user-specific directory take precedence over files in a systemwide customizable directory, which take precedence over the files in the systemwide installed directory.

Note

The CDE$USER_DEFAULTS directory hierarchy is not created at installation. If you need to add any systemwide customizations, you must create the corresponding directories.

3.3.1 System Default Configuration Directories

CDE$SYSTEM_DEFAULTS:[*...] is the system directory hierarchy of the New Desktop. It contains read-only directories, which contain resource, application defaults, configuration, and application support files, as shown in Figure 3–1 and described in Table 3–1. These files should not be edited because future installations will overwrite the files. Furthermore, applications should never create files in these directories, although they can read and execute the files.

3.3.2 Directory Hierarchy With Multiple Locales

The [.C] directories shown in Figure 3–1 contain locale-specific files required for running the default C locale installed on your system. For each additional locale installed, a corresponding directory at the same level as each [.C] directory is present. For example, if the French locale is also installed on your system, the directory hierarchy of [.APP-DEFAULTS] will show both [.C] and [.FR_FR_ISO8859-1] directories. Files specific to the French locale are placed in the [.FR_FR_ISO8859-1] directories and are accessed before the files with the same name in the [.C] directories.

Figure 3–1   CDE$SYSTEM_DEFAULTS Directory Hierarchy


```
                          CDE$SYSTEM_DEFAULTS:[000000]

[.APPCONFIG]  [.APP DEFAULTS]  [.BACKDROPS]  [.BIN]  [.CONFIG]  [.EXAMPLES]  [.LIB]  [.MAN]  [.PALETTES]
                    |                                   |                      |
                   [.C]                          [.C]   [.XFONTS]           [.NLS]
                                                         |                    |
                                                        [.C]               [.MSG]
                                                         |                    |
                                                 [.75DPI]  [.100DPI]        [.C]

[.APPMANAGER]  [.HELP]  [.ICONS]  [.TYPES]
    |            |        |         |
   [.C]         [.C]     [.C]      [.C]

[.DECWINDOWS_APPS]   [.DECWINDOWS_UTILITIES]   [.DESKTOP.APPS]   [.DESKTOP.TOOLS]
```
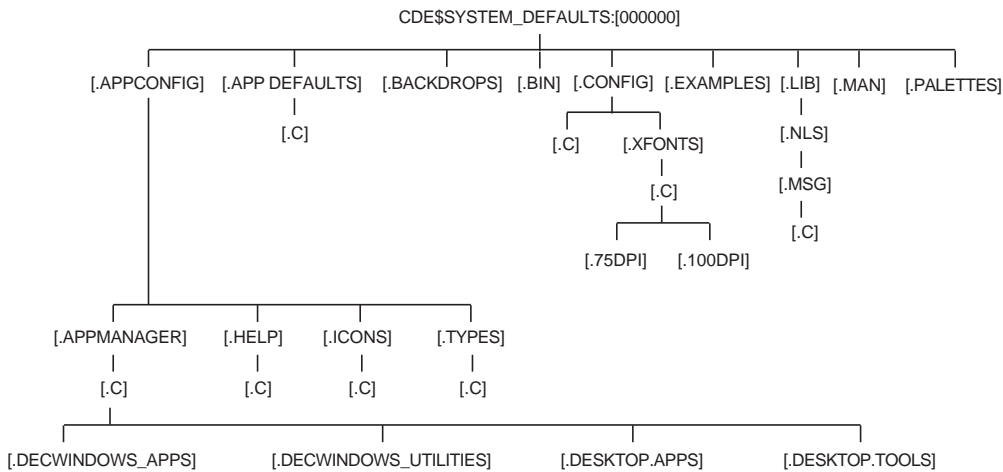
ZK-8651A-AI

Table 3–1   CDE$SYSTEM_DEFAULTS Directories

Directory	Contents
[.APPCONFIG]	Application management, help, example, icon, and datatype definition files, which include Front Panel configuration files. The [.APPMANAGER] directory contains several subdirectories, each corresponding to a different application group. Each subdirectory contains action (stub) files, one for each application in that group.
[.APP-DEFAULTS]	Application default (resource) files for applications provided by the New Desktop.
[.BACKDROPS]	Backdrops provided with the New Desktop.
[.BIN]	New Desktop executable (binary) files.
[.CONFIG]	System configuration files for global resource settings for various aspects of the New Desktop, such as Session Manager and Window Manager resources. Font alias files are contained within the [.XFONTS] subdirectory hierarchy.
[.EXAMPLES]	Example files that show how to use CDE programming resources.
[.LIB]	Localized message catalog files.
[.MAN]	Unmodified, preprocessed UNIX manpages for the New Desktop.
[.PALETTES]	Color palettes provided with the New Desktop.

3.3.3 Configuration Directories for Systemwide Customizations

The following directory hierarchy is the systemwide configuration directory for the New Desktop:

```
CDE$USER_DEFAULTS:[*...]
```

All directories listed in Table 3–1 can have a corresponding directory in the CDE$USER_DEFAULTS:[*...] hierarchy. Systemwide customizations should be made by first copying the file to be customized from its CDE$SYSTEM_ DEFAULTS location to the same directory in the CDE$USER_DEFAULTS hierarchy and then editing the file. You may need to create the destination directory.

The CDE$USER_DEFAULTS logical is a search path containing the logicals CDE$USER_SPECIFIC and CDE$USER_COMMON. Node-specific changes should be placed in the appropriate subdirectory of the CDE$USER_ SPECIFIC:[*...] hierarchy. Clusterwide changes should be placed in the appropriate subdirectory of the CDE$USER_COMMON:[*...] hierarchy.

Table 3–2 describes how to customize a system default for a node or a cluster.

Table 3–2 Customizing Defaults for a Node or Cluster

System Type	Directions
Node	Copy the file containing the default value you want to change from its directory under CDE$SYSTEM_DEFAULTS to the same directory under CDE$USER_SPECIFIC and edit it.
Cluster	Copy the file containing the default value you want to change from its directory under CDE$SYSTEM_DEFAULTS to the same directory under CDE$USER_COMMON and edit it.

When a user or an application reads from CDE$USER_DEFAULTS, CDE$USER_SPECIFIC:[*...] is searched before CDE$USER_COMMON:[*...].

When a user or an application writes a file to CDE$USER_DEFAULTS:[*...], it is placed in the CDE$USER_SPECIFIC:[*...] hierarchy. To write a file to CDE$USER_COMMON:[*...], you must specify CDE$USER_COMMON.

3.3.4 Configuration Directories for User Customizations

When you first log in to the New Desktop, a user-specific configuration directory hierarchy is created for you, as shown in Table 3–3. The following top-level directory of this hierarchy is placed in your home directory:

disk$:[*user*.DT]

This directory hierarchy contains information that is specific to an individual user. The information in this directory overrides or supplements any resources or customizations found in the system-specific configuration directories (CDE$USER_DEFAULTS:[*...]).

The *disk*$:[*user*.DT...] directory hierarchy contains an error log, user-defined configuration files for Application Manager, session state and save-and-restore information, user-customized backdrops, help configuration files, icons, and palettes.

Table 3–3   Private Directories Created When You First Log In

Directory	Contents
disk$:[*user*.DT.APPMANAGER]	Private application groups
disk$:[*user*.DT.DESKTOP]	Information to restore icons dropped on the desktop
disk$:[*user*.DT.HELP]	Private help files
disk$:[*user*.DT.ICONS]	Private icon files
disk$:[*user*.DT.PALETTES]	Private palette files
disk$:[*user*.DT.SESSIONS]	Session state and save-and-restore information work area
disk$:[*user*.DT.TMP]	Temporary files created by the New Desktop
disk$:[*user*.DT.TRASH]	Files moved to the Trash Can and information for processing them
disk$:[*user*.DT.TYPES]	Private action definitions

3.3.5 Order of Precedence of Data Files

The order of precedence for data files read by New Desktop applications is:

1. *disk*$:[*user*.DT...]

2. CDE$USER_SPECIFIC:[*...]

3. CDE$USER_COMMON:[*...]

4. CDE$SYSTEM_DEFAULTS:[*...]

3.4 Customizing the Environment and Applications

You can customize your New Desktop environment in several ways. Most customizations are made by modifying resource files. You can modify them directly by editing them or by using the xrdb utility, or you can modify them indirectly with Style Manager.

You can also customize your environment by adding or modifying actions. You can add new actions or modify existing actions with the Create Action application (see Section 4.1). Alternatively, you can add new actions or modify existing actions by creating or editing action definition files and their associated files.

3.4.1 Using Resource Files With the New Desktop

The New Desktop, like any X windows environment, uses resource files to define global and application-specific attributes. Global resources are read when you log in to the New Desktop. They apply to all applications. Application-specific resources are read when a specific application is started.

Most resources files currently in use with the DECwindows desktop will continue to work under the New Desktop. The logical name DECW$SYSTEM_ DEFAULTS for systemwide resources and the logical name DECW$USER_ DEFAULTS for user-specific resources are still searched. The global user defaults file DECW$USER_DEFAULTS:DECW$XDEFAULTS.DAT is still read at login time.

However, some differences exist with the New Desktop. Global systemwide resources are read from the SYS.RESOURCES and SYS.FONT files in the CDE$SYSTEM_DEFAULTS:[CONFIG.C] directory. User-specific global re-sources are read from the DECW$USER_DEFAULTS:DECW$XDEFAULTS.DAT file and from the user's DT.RESOURCES file, which resides in a subdirectory of *disk*$:[*user*.DT.SESSIONS]. The choice of the subdirectory depends on the session type (current or home) you are using.

The DT.RESOURCES file contains customizations specified by running the New Desktop Style Manager. These resource files serve basically the same purpose as the DECW$SMB_*.DAT files of the DECwindows desktop, which are created using Session Manager's Options pull-down menu. With few exceptions, anything that you can customize with the DECwindows Session Manager's Options pull-down menu is customizable with the New Desktop Style Manager.

Systemwide, application-specific resources for the CDE applications included in the New Desktop are placed in the CDE$SYSTEM_DEFAULTS:[APP-DEFAULTS.C] directory. This directory is searched if the resource file is not found in the DECW$SYSTEM_DEFAULTS: directory. The complete directory hierarchy for files provided with the New Desktop in described in Section 3.3.

Note

All resource identifiers, such as *hostList: and *keyboardDialect:, are case sensitive. The value that you supply after the colon (:) is usually not case sensitive, but this depends on how the applications interpret the value. File name resource values are not case sensitive.

3.4.2 Using Actions With the New Desktop

An action is an association of a name and, in many cases, an icon with an executable command. The File Manager and Front Panel use actions to present an iconic representation of operations that can be performed, in the case of the File Manager, on a particular file. Actions are defined in action definition files (file types .DT and .FP) that are placed in any of the directories designated by the action definition file search path DTDATABASESEARCHPATH. For more information about actions, see Section 4.1.

3.4.2.1 Data Typing

All files represented by the File Manager have one or more actions associated with them. The determination of which actions apply to which files is established by data typing. The most common form of data typing is the specification of a file name pattern. For example, a file with a .COM extension has a data type of COM_FILE, which in turn has actions of *Open* (edit) and *Run*. Run is the default action for command procedures, which means that double clicking on the icon for a file with the .COM extension will run it. You can customize the action database by using the Create Action application or by editing .DT files, which define the action database.

3.4.2.2 Action (Stub) Files

An action file is an empty file with the same name as an action. An action file must have execute permissions enabled.

An action file is used by File Manager to create a reference to the icon representing the action. The Create Action application automatically creates this file in your SYS$LOGIN directory when you select the Save option from the File pull-down menu. Action files can be moved to application groups or placed on the desktop window.

Note

Action names are case sensitive. They must be lowercase both to match action files and to be used by File Manager and Application Manager.

3.4.3 Security Settings

In the DECwindows desktop, security access is enabled via Session Manager's Options pull-down menu. Once access is specified, changes are saved in the DECW$USER_DEFAULTS:DECW$SMB_SECURITY.DAT file (and referenced by the resource sm.host_list:).

When you first log in to the New Desktop, Session Manager (dtsession) reads any existing DECW$USER_DEFAULTS:DECW$SMB_SECURITY.DAT file and transfers the resource (sm.host_list:) to an appropriate resource (*hostList:) in the New Desktop. The resource is saved in your DT.RESOURCES file when you save a home session or exit your session.

You can modify security access for your session by using the graphical user interface provided in Style Manager. From the Style Manager menu bar, select Security and then add or delete authorized users.

3.4.4 Keyboard Settings

If you have a DECW$USER_DEFAULTS:DECW$SMB_KEYBOARD.DAT file in the DECwindows desktop environment, the value of the keyboard dialect resource (keyboard_dialect) is transferred to your New Desktop session when you first log in. The new name for this resource is *keyboardDialect.

You can use Style Manager to modify the keyboard dialect and input method selection. When you select the keyboard icon, a list box containing all possible keyboard mappings is displayed. When you select a keyboard mapping, it takes effect immediately. The resource is saved in your DT.RESOURCES file when you save a home session or exit your session.

3.4.5 Printer Management

Print Dialog provides the New Desktop with access to the DECwindows print widget for customizing and submitting print jobs. By default, Print Dialog saves its application state in the file *disk*$:[*user*.DT.TMP]DTPRINTINFO when the user clicks on the OK button. If the *disk*$:[*user*.DT.TMP] directory does not exist, or if the user does not click on the OK button prior to submitting the job, the Print Dialog settings selected are not used in the current print job.

If Print Dialog does not appear, or if the print job is not submitted, try deleting the latest version of the application state file in [*username*.DT.TMP] and reissuing the command.

3.4.5.1 Adding Printers to the Personal Printers Subpanel

Printers are represented as icons on the desktop. You can change the default printer and you can add additional printer icons to the Personal Printers subpanel, provided you have an action definition file, an action (stub) file, and an icon for each additional printer. The easiest way to create these three files is with the Create Action application (see Section 4.1.4).

A template printer action definition file is supplied with the New Desktop in the following location:

```
CDE$SYSTEM_DEFAULTS:[APPCONFIG.TYPES.C]PRINTER.DT_TEMPLATE
```

Read this file for instructions on creating printer action definitions.

You can add printer icons to the Personal Printers subpanel using the following steps:

1. Display the printer icon using File Manager.

2. Display the Personal Printer subpanel.

3. Drag the printer icon from File Manager to the Personal Printer subpanel and drop it on the Install Icon item.

3.4.5.2 Creating a Printer Subpanel Control Definition File

Dropping a printer icon on the Install Icon control at the top of the Personal Printers subpanel automatically creates a printer subpanel control definition file and adds the new printer control to the bottom of the subpanel.

A template file for a printer subpanel control definition file is available in the following location:

```
CDE$SYSTEM_DEFAULTS:[APPCONFIG.TYPES.C]PRINTER.FP_TEMPLATE
```

Read this file for instructions on creating printer subpanel control definitions.

3.4.5.3 Changing the Default System Printer

To change the default system printer, add the following line to the SYS$MANAGER:SYSTARTUP_VMS.COM file:

```
$ DEFINE/SYSTEM/EXECUTIVE_MODE SYS$PRINT printer-name
```

3.4.5.4 Changing the Default Printer for Your Own Use

The default printer is the printer associated with the Printer icon on the Front Panel. By default, this printer is defined by the value of the SYS$PRINT logical.

To change the default printer for your own use, add the following line to your LOGIN.COM file:

```
$ DEFINE SYS$PRINT printer-name
```

Alternatively, you can change the default printer for your own use by selecting a different printer from the list of available printers in the Printer Dialog box, as follows:

1. Click on the printer icon on the Front Panel to display the Printer Dialog box.

2. From the list of printers on your system, click on the printer you want as a new default printer.

3. Click on OK.

Note that changing the default printer for Print Dialog does not change the default printer associated with the Print item on File Manager's Selected pop-up menu. You can change the default printer used by File Manager by following these steps:

1. Click on the icon of the file you want to print.

2. Click on the Selected item of the File Manager menu bar to display the Selected menu.

3. Click on Print... to display the Printer Dialog box.

4. From the list of printers on your system, click on the printer you want as a new default printer for File Manager.

5. Click on OK.

The default printer and other options that you set using Print Dialog remain in effect until you change them again.

3.4.5.5 Print Dialog Command Line Options

Print Dialog accepts the command line options shown in Table 3–4. You can use these in a print command that you supply to the Create Action application (in the Add Datatype dialog box). You can also use them for printing a file from a DECterm with the options that you specified in the New Desktop environment, if you do not have access to the file's icon. You might also want to use Print Dialog from a DECterm for debugging print actions for your data types.

Table 3–4 Print Dialog Command Line Options

Option	Description
-session *sessionfile*	Loads the session file and restores the client to its original state on the desktop. By default, the session file name is determined at logout by the desktop.
-state *statefile*	Loads the state file and restores the client to its original application state. By default, the state file is *disk*$:[*user*.DT.TMP]DTPRINTINFO.
-interface	Displays the Print Dialog interface. By default, this value is FALSE.
-unmanage	Unmanages the Print Dialog interface when the user clicks on the OK button. By default, this value is TRUE.
-printer *print-queue*	Specifies the name of the printer to receive the print job request. The printer name saved in the application state file takes precedence over this option. By default, this value is set to the value of the default printer.
-default *print-queue*	Specifies the name of the default printer to receive the print job request. The printer name saved in the application state file takes precedence over this option. By default, this value is set to the value of the SYS$PRINT logical of the system or to the printer name specified with the -printer option.
filenames	Path names of the files to be printed. If this argument is used with the -interface option, the files are printed when the user clicks on OK.
-usage	Displays the command line options.

3.4.5.6 Starting Print Dialog From a Command Line

To run Print Dialog from a DECterm window, define the following symbol:

```
$ PRINTDIALOG :== $CDE$PATH:PRINTDIALOG
```

To display Print Dialog, enter the following command:

```
$ PRINTDIALOG -interface
```

To print a file, enter the following command:

```
$ PRINTDIALOG filename
```

3.4.6 Changing the Default Editor

The default editor of the New Desktop is the CDE Text Editor. If you prefer to use another editor, you can make it the default editor by following these steps:

1. Copy the file CDE$SYSTEM_DEFAULTS:[APPCONFIG.TYPES.C]USER-PREFS.DT into your *disk*$:[*user*.DT.TYPES] subdirectory.

2. Within the new copy of the file, revise the specification for "ACTION TextEditor".

   To use the DECwindows TPU editor, change MAP_ACTION from dtpad to dxeve. To run the TPU editor in a DECterm window, change MAP_ACTION from dtpad to tpuedit.

Both dxeve and tpuedit are preconfigured actions. These preconfigured actions may not be adequate if you are accustomed to editing with a customized TPU section file or with command files. To edit with a customized section file or with command files, you need to explicitly specify your edit command in an action definition file (see Section 4.1) and then map the TextEditor action to the new action name.

3.4.7 Window Colors and Screen Background

Window colors and screen background settings in the DECwindows desktop resource files (DECW$SMB_WINDOW_COLOR.DAT, DECW$SMB_BACKGROUND_COLOR.DAT) are not read into the New Desktop resources.

You can use Style Manager's Color dialog box to customize window colors and screen backgrounds in the New Desktop. You can specify a different background for each workspace, which can be useful for identifying them. You can also use the Color dialog box to troubleshoot the color display, as described in Section 3.6.

3.4.8 Window Manager Customizations

In the DECwindows desktop, Window Manager configuration customizations are specified in two files: DECW$MWM.DAT and DECW$MWM_RC.DAT. The New Desktop Window Manager supports configuration customization in two similar files: DTWM.DAT and SYS.DTWMRC, respectively.

3.4.8.1 Window Manager Resource File (DTWM.DAT)

The Window Manager default resource file, DTWM.DAT, is located in the
CDE$SYSTEM_DEFAULTS:[APP-DEFAULTS.C] directory. Information stored
in this file affects many aspects of the Window Manager, including the pointer
focus, the icon box, the placement of the icon box, icon colors, use of the Front
Panel, window border shading, window menu appearance, and workspace
backdrop specifications. The following example shows part of the default
DTWM.DAT file:

```
Dtwm*Dtstyle*iconImage:      Fpstyle
Dtwm*Dthelpview*iconImage:   Fphelp
Dtwm*backdrop*image: SkyLight
Dtwm*ws0*backdrop*image: SkyLight
Dtwm*ws1*backdrop*image: WaterDrops
Dtwm*iconBoxGeometry:                    1x4+16+20
Dtwm*iconBoxSBDisplayPolicy:             horizontal
```

Most of the resources defined in the DECwindows Window Manager resource
file (DECW$MWM.DAT) can be used with the New Desktop's Window
Manager by adding the applicable resources to DTWM.DAT and changing
the application class name from Mwm to Dtwm. Note that customizations to
the DECwindows Window Manager resource file (DECW$MWM.DAT) are not
automatically carried forward to the New Desktop's Window Manager resource
file (DTWM.DAT). The user or system manager must make the appropriate
edits to the DTWM.DAT file.

As with all application resource files, user-specific customized versions of this
file should be placed in the DECW$USER_DEFAULTS directory (which is
typically the user's home directory). To put into effect any changes made to the
Window Manager resource file, you must restart the Window Manager.

For more information about the Window Manager resources, refer to the Dtwm
manpage.

3.4.8.2 Window Manager Resource Configuration File (SYS.DTWMRC)

The Window Manager resource configuration file for the New Desktop,
SYS.DTWMRC, is available in CDE$SYSTEM_DEFAULTS:[CONFIG.C]. It is
a supplementary resource file that is used with the Window Manager resource
file (DTWM.DAT) to control much of the behavior of the New Desktop Window
Manager.

In the DECwindows desktop, supplemental window manager configurations
are made via the DECW$MWM_RC.DAT file. Information stored in this
resource file affects the Window Manager Root Window Menu, customized key
binding descriptions, mouse button binding descriptions, and other attributes.
Customizations made to DECW$MWM_RC.DAT are not automatically

carried forward to the New Desktop. You can update the New Desktop's SYS.DTWMRC file with previous customizations made to DECW$MWM_RC.DAT by doing the following:

1. Copy CDE$SYSTEM_DEFAULTS:[CONFIG.C]SYS.DTWMRC to *disk*$:[*user*.DT]DTWMRC.DAT.

2. Make any edits that you would like to the file. You can incorporate changes previously made to DECW$MWM_RC.DAT and make other edits.

3. Restart the Window Manager to put these changes into effect.

Functional enhancements are available for DTWMRC.DAT, such as a series of new Workspace Manager functions (for example, f.create_workspace) to support the multiple-workspace environment. For more information about this new functionality, as well as more information about customizing DTWMRC.DAT, refer to the DTWMRC manpage.

3.4.9 Automatic Application Startup

The DECwindows desktop provides the Automatic Startup... option from Session Manager for specifying which applications should be started when you log in to a session. You can put additional customizations in the SYS$LOGIN:DECW$LOGIN.COM file.

The New Desktop provides a different technique for starting applications when you log in to a session. There are two options, both based on starting a default set of applications you want and then saving the state of your session: Resume Current Session or Return to Home Session.

Style Manager's Startup menu provides these options, as described in Section 2.1.11.

Applications are responsible for informing the New Desktop how to restart them. The CDE applications of the New Desktop give detailed information, including which windows are displayed, where they are positioned, and current values. The DECwindows applications, in general, provide only enough information for them to be restarted; no state information is preserved.

Optional DECwindows applications, such as DECwindows Notes, may not provide any information and therefore are not restarted. For such cases, there is a command procedure called *disk*$:[*user*.DT]SESSIONETC.COM, that you can use to start any applications that cannot be restarted automatically. This procedure is analogous to the DECW$LOGIN.COM procedure in the traditional DECwindows environment.

3.5 Using the New Desktop Error Logs

The New Desktop provides two log files by default—one for session startup and one for the session itself—and a third log file for the login process, if you define the symbol for it. These log files are described in the following list:

- Login log file

  To create a log file for the login process, define the DECW$LOGINLOG symbol in your SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM file as the name of the log file you want to create. (This is the same way you create a log file for the login process for the traditional DECwindows desktop.) For example, the following definition will write login log file messages to the file SYS$MANAGER:LOGIN.LOG:

  ```
  $ DECW$LOGINLOG == "SYS$MANAGER:LOGIN.LOG"
  ```

- Session startup log file

  After login but before Session Manager takes control, the log file SYS$LOGIN:DECW$SM.LOG is used to display information about session startup. Additional information pertaining to exiting your session is added to the file. This file is analogous to the file of the same name in the traditional DECwindows environment.

  If you do not want this log file, you can define the symbol DECW$SESSIONLOG to be "FALSE" in your SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM file, as shown in the following example:

  ```
  $ DECW$SESSIONLOG == "FALSE"
  ```

- Error log file for the session

  Once Session Manager takes control, all errors are written to the session error log file *disk*$:[*user*.DT]ERRORLOG. All New Desktop applications running in the current session write their status and error information to this file with a timestamp on each entry. In addition, application output (data sent to SYS$OUTPUT) is captured in the error log file, except in the following cases:

  - If the application is run with its own DECterm for displaying output.

  - If the output is explicitly sent somewhere other than SYS$OUTPUT.

  - If the account of the user running the application differs from the account of the user who is logged in to the New Desktop.

  - If the application is runing on a system other than the one on which the New Desktop Session Manager is running.

A new ERRORLOG. file is created each time you log in. The previous version is renamed to ERRORLOG.OLD, and the version before that is renamed to ERRORLOG.OLDER. There is only one version of each of these files. Only the three latest error log files are saved; previous error logs are automatically deleted.

The Watch Errors action, provided in the Desktop Tools application group, enables easy viewing of the ERRORLOG. file.

3.6 Using Colors in the New Desktop Environment

In the New Desktop, color allocation is centralized. A set of colormap entries is preallocated when a session starts. The default values for widget color resources are mapped to these colormap entries. All applications that do not explicitly set widget color resources will use the shared colormap entries. By using this approach you can dynamically change colors for all applications by changing the pixel values of the shared colormap entries.

In the New Desktop, the Session Manager acts as the color server for the desktop. It reserves and controls a portion of the X server's colormap, dividing it into a series of color sets. A color set consists of a color for each of the resources, as described in Table 3–5.

Table 3–5 Color Set Resources

Resource	Description
Background	An application's background color
Foreground	An application's foreground color
topShadowColor	The color for the top and left borders of application controls and the window frame
bottomShadowColor	The color for the right and bottom borders of application controls and the window frame
selectColor	The color used to highlight selected controls in the application

Style Manager combines up to eight color sets to create a color palette. The widgets to which these color sets apply are determined by color set ID resources. A color set can be applied to all widgets, only to popup dialog boxes, or only to text widgets based on these resources.

The colors in each color set are based on the background color choice. The background color for each color set is displayed in Style Manager's Color dialog as a colored button. Depending on the setting of the Number of Colors option,

two to eight buttons will be displayed. The Number of Colors option appears
on Style Manager's Color menu, shown in Figure 3–2.

Figure 3–2   Style Manager's Color Menu
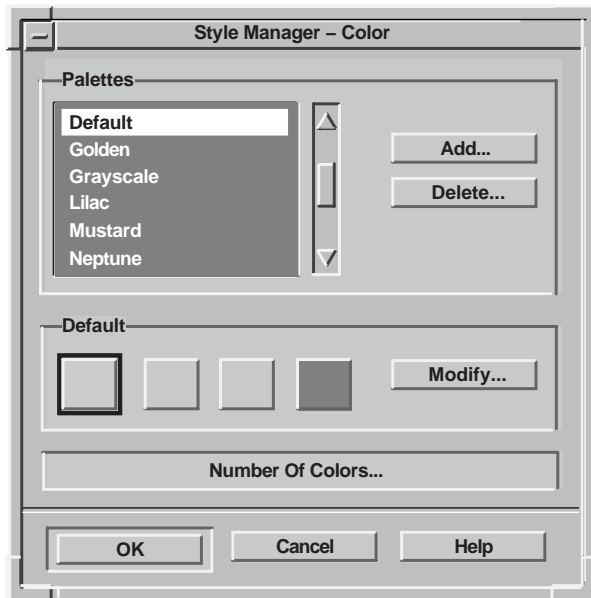
ZK–8685A–GE

You can set the Desktop's colors by choosing from the list of color palettes and
modifying the color sets displayed in the buttons with the Modify... dialog.

When you select a color button and move the sliders in the Modify dialog,
the application's colors displayed in that color will change if the application
supports dynamic colors. If the color does not change when you move the
sliders, you can assume that the application's colors are not under the control
of the New Desktop but are under the control of the application. You can
change an application's color resources to change its colors.

3.6.1 Changing the Number of Available Colors

It is possible that the colors needed by an application combined with the colors needed by the New Desktop could exceed the maximum number of colors available on the display (even if they are the same colors). If an application cannot set all the colors it needs, it can reveal the problem by:

- Generating errors such as "Unable to allocate colormap entry...".

- Not generating errors but displaying in different colors or changing the desktop's colors when they are started.

If an application exhibits any of these symptoms, use the Number of Colors... dialog, shown in Figure 3–3, to give as many colors as possible to the application. The number of color sets used by the desktop depends on the selection in the Number of Colors... dialog. The fewer color sets used for the desktop, the more colors available for an application. The Number of Colors options are described in Table 3–6.

Figure 3–3 Number of Colors Menu
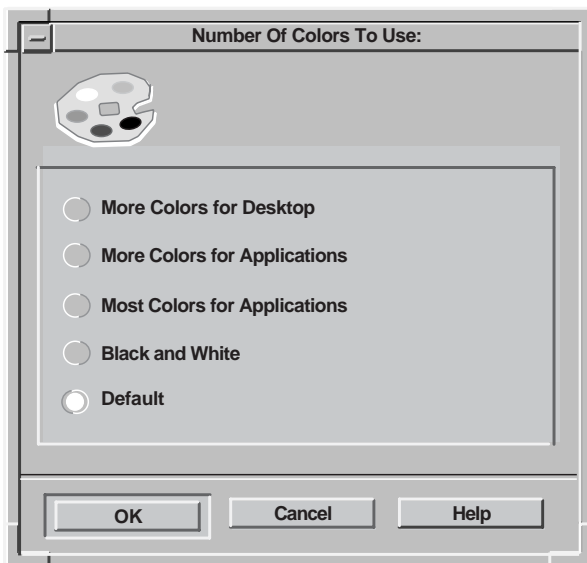
ZK–8686A–GE

Table 3–6   Number of Colors Options

Options	Description
More Colors for Desktop	Specifies a high color display. The color palettes use 8 color sets and have a maximum of 42 colors (including black and white).
More Colors for Applications	Specifies a medium color display for the desktop. The color palettes use 4 color sets and have a maximum of 22 colors (including black and white).
Most Colors for Applications	Specifies a low color display for the desktop. The color palettes use 2 color sets and have a maximum of 12 colors (including black and white).
Black and White	Specifies a black-and-white display. The color palettes use only two colors. In this nondynamic color configuration, the available palettes are BlackWhite and WhiteBlack.
Default	Allows the color server to select the color configuration based on the number of colors on the display. The default is usually More Colors for Applications.

_____ **Note** _____

The Color Customizer example program uses a similar technique for providing a dynamic color environment. It allocates an additional set of colors for the color resources it manages, which further limits the number of colors available for applications. This can cause some undesired interactions with the dynamic color environment provided with the New Desktop.

_____

For more information about how the New Desktop manages the colormap, use the Man Page Viewer to view the dtsession reference page.

3.6.2   Avoiding Color Conflicts

It is usually best if applications use the default colors and allow the desktop to manage the colors consistently for all applications. However, this isn't always possible. If an application is not an OSF/Motif application running against the new version of the Motif library, it will not be able to take advantage of the new dynamic color model. Also, if color resources are explicitly defined, then shared dynamic colors will not be used. These applications should be able to coexist, but some undesired effects could occur if a palette is selected that conflicts with predefined colors in an application.

An application may exhibit "antisocial" color behavior—that is, it modifies colormap entries that it has not allocated for its own use. If these are desktop dynamic color entries, it will affect all applications. You may need to reselect the current palette in Style Manager to reset the colors properly.

Conflicting colors can make text hard to read or even invisible. They can also make it appear as if a control has disappeared from the application interface. Such effects are caused by the foreground color of an application control being set to a color similar to its background color, such as a white foreground on a white background.

Session Manager provides a number of resources that can be used to affect the foreground and background colors of the desktop. Also, many applications provide resources for setting the foreground and background colors. However, the simplest way to resolve these problems may be to select a different palette in Style Manager (or to use the Modify... dialog box to change the current palette).

To change any Session Manager resource, copy its application defaults file to the DECW$USER_DEFAULTS directory, as shown in the following example:

```
$ COPY CDE$SYSTEM_DEFAULTS:[APP-DEFAULTS.C]DTSESSION.DAT -
_$ DECW$USER_DEFAULTS:DTSESSION.DAT
```

Add or modify the resource strings in this file, exit the New Desktop, and log in again to restart Session Manager. By default, Session Manager loads the foreground and background resources each time the user logs in. This is due to the following default resource of Session Manager:

```
dtsession*writeXrdbColors: true
```

The foreground and background colors are determined dynamically from the colors in the user's selected palette. This is due to the following two default resources of Session Manager:

```
dtsession*dynamicColor: true
dtsession*foregroundColor: "DYNAMIC"
```

The *dynamicColor resource controls whether colors can be changed dynamically when you change palettes. The *foregroundColor resource controls whether the *foreground resource is set explicitly to either "BLACK" or "WHITE" or whether it is calculated based on the current palette. If "DYNAMIC" is used, the following resource controls the threshold above which the *foreground resource is set to white:

```
*foregroundThreshold: 70
```

which is specified in the file:

```
CDE$SYSTEM_DEFAULTS:[APP-DEFAULTS.C]DT.DAT
```

Note

Changing the default value of the *foregroundThreshold resource to set
the foreground color of any particular palette may cause unpredictable
results in the foreground color of other palettes. The default value of
70 is used because it works correctly with the installed set of default
palettes.

You can change the foreground color in an application from black or white to
another color by adding a foreground resource specification to its application
resource file. Since the default foreground resource specification is defined in
its most general form, any application-specific foreground resource specification
will override it. For example:

```
DECW$TERMINAL.main.terminal.foreground: goldenrod
```

overrides

```
*foreground: #FFFFFFFFFFFF (White)
```

In this example, the color goldenrod is applied to the foreground of all
DECterms created after this change. The foreground of all other applications
remains white.

3.7 Using the New Desktop Fonts

Fonts used by applications within the New Desktop are referenced by generic
font names. These font names are not the names of actual fonts but rather
are the names of font aliases. These aliases are mapped to real font names by
means of font alias files. When the New Desktop login process starts, it selects
the appropriate font alias files based on the current language selected and the
resolution of the display. The New Desktop instructs the X server to which
it is connected to reset its font path to include the new font directories. This
operation is possible only if the login process is running on the same system as
the X server. The login process determines whether this is the case by looking
at the display node name and the current transport type.

_____ **Note** _____

The font alias directories are added to the font path only if the LOCAL
transport (the default) is used or if a display node name of 0 is used.

_____

For remote displays, New Desktop fonts are used only if the New Desktop font
aliases are defined for the remote display.

3.7.1 Changing Application Font Size

With Style Manager's Font option, you can select the size of fonts used for all
CDE applications. Your selection affects every CDE application that you start
after you make the change, but it does not affect any CDE applications that
are already running. Furthermore, the font size option does not affect any
existing DECwindows applications.

3.8 Using X Terminals and Other Remote Displays

When using the New Desktop with remote displays such as X terminals, you
need to give special consideration to certain components, including window
managers and font aliases.

3.8.1 Selecting the New Desktop Window Manager

Most X terminals provide some local capabilities, including a local window
manager that runs on the X terminal itself. The Workspace (Window) Manager
provided with the New Desktop is an integral part of the New Desktop and
must be used in order for the New Desktop to operate properly. Most local
window managers for X terminals can yield to a remote window manager when
it starts up. However, this feature is sometimes not the default option.

For example, the Digital VXT 2000 X Terminal provides the Allow Remote
Window Manager option, but it is not the default. To use the New Desktop
Window Manager, you must enable the Allow Remote Window Manager
option, which is under the Customize option in the Terminal Manager... dialog
box. For an equivalent option on other X terminals, consult your X terminal
documentation.

3.8.2 Accessing Font Aliases

For X terminals and in any environment where the X server is not running on the local system, the font path is not modified to include the New Desktop font alias directories during the login process. As a result, the X server does not recognize the font names used with any of the New Desktop applications. Most applications use the same fallback font, and you will *not* be able to modify its size through the use of Style Manager's font option. Applications will run properly, but the size and font style of many of the applications will not be what you expect.

To avoid this problem, you need to make the New Desktop font aliases available to the X server. One way to do this is to use a font server running on a UNIX system. If the UNIX system supports the Common Desktop Environment (CDE), then the correct font aliases should be in place. If not, the font alias files can be copied from the OpenVMS Alpha system to the UNIX system. For the default C locale, the font alias files are:

```
CDE$SYSTEM_DEFAULTS:[CONFIG.XFONTS.C.100DPI]DECW$FONT_ALIAS.DAT
CDE$SYSTEM_DEFAULTS:[CONFIG.XFONTS.C.75DPI]DECW$FONT_ALIAS.DAT
CDE$SYSTEM_DEFAULTS:[CONFIG.XFONTS.C]DECW$FONT_ALIAS.DAT
```

Each of these files should be copied to appropriate font alias directories on the UNIX system and renamed to fonts.alias. See your UNIX X server and font server documentation for details on how modify font paths if necessary. Once the font server recognizes the New Desktop font aliases, add the font server to the X terminal or remote system's default font path.

For other languages, the font alias files are:

```
CDE$SYSTEM_DEFAULTS:[CONFIG.XFONTS.language.100DPI]DECW$FONT_ALIAS.DAT
CDE$SYSTEM_DEFAULTS:[CONFIG.XFONTS.language.75DPI]DECW$FONT_ALIAS.DAT
```

For non-C languages, always include the C locale font aliases after the language-specific font aliases.

Note

At the time of this printing, Digital does not provide an OpenVMS font server. However, font servers for OpenVMS Alpha are available from other vendors.

If you do not have access to a font server, you may be able to include the font alias files directly in the font path of the X terminal or remote X server. You do this by copying the necessary font alias files to a location accessible by the

remote system's X server. If the remote X server is running on an OpenVMS Alpha system, the correct locations for font alias files are:

```
SYS$COMMON:[SYSFONT.DECW.USER_100DPI]
SYS$COMMON:[SYSFONT.DECW.USER_75DPI]
SYS$COMMON:[SYSFONT.DECW.USER_COMMON]
```

After copying the font alias files, you need to reset the font path. You can do this by simply logging out of your session and then logging back in.

Note

Digital X terminals cannot read font alias files using the OpenVMS LAT font daemon. You must use NFS or TFTP to access font alias files on an OpenVMS Alpha system.

3.8.3 Switching Languages

With remote displays, the font path is not dynamically changed at login time to include the font alias files for the selected language. Each time you change languages, you must edit the font path of the X terminal or remote display.

3.9 Using Multiple Screens

The New Desktop supports a single X display with multiple screens. This configuration is sometimes called a dual-headed display. For information about creating and managing multiple screens, see Appendix B.

3.10 Managing Processes Created Within the New Desktop

The New Desktop consists of many applications. Most of these applications are started implicitly on your behalf as detached processes. Most process state information, such as process logical names and global symbols, is not passed to detached processes. Usually only the DECW$DISPLAY and LANG logicals are passed. You can pass additional process logical names with a special symbol, or you can specify that subprocesses be created instead of detached processes.

3.10.1 Passing Process Logical Names to Detached Processes

If you require that additional process logical names be passed to detached processes, you can use a special mechanism within the New Desktop. This mechanism works using the symbol CDE$DETACHED_LOGICALS, which you define in your private applications setup file, SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM. For example, the

following definition causes the logical name MYLOGICAL to be passed to all created detached processes:

```
$ CDE$DETACHED_LOGICALS == "MYLOGICAL"
```

3.10.2 Specifying the Creation of Subprocesses

You can specify that all processes be created as spawned subprocesses instead of detached processes by defining the symbol CDE$SPAWN_PROCESSES in SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM. The following definition causes all processes to be created as subprocesses:

```
$ CDE$SPAWN_PROCESSES == "TRUE"
```

Certain process quotas are shared by a process and all of its subprocesses. Specifying the creation of subprocesses may require that you increase some of your process quotas using the Authorize utility. In particular, PRCLM, PGFLQUOTA, and BYTLM may need to be increased.

3.10.3 Debugging Processes

You may want to display the actual DCL commands used to invoke each process in the New Desktop for debugging purposes. By defining the following symbol, you can capture this information in your error log file:

```
$ CDE$LOG_PROCESSES = "TRUE"
```

By default, this information is not captured in your error log file. Note that this information may significantly increase the size of your error log file if you are iteratively creating processes, for example, if you have selected multiple screen savers with a short time per background.

3.10.4 Identifying Processes By Name

Both the New Desktop and the DECwindows desktop have a well-defined naming strategy for the processes they create. You can see a list of processes on your system by using either the SHOW SYSTEM command or the SHOW USERS/FULL command.

The processes created within and specific to the New Desktop are shown in Table 3–7.

Table 3–7   Process Names of the New Desktop

Name[1]	Purpose
DTLOGIN	Login Manager
DTGREET	Login user-interface application
DTHELLO_*n*	Welcome screen displayed at login
DTSESSION	Session Manager
DTWM	Workspace (Window) Manager
user$CDE*n*	Detached process created within the New Desktop

[1]In these names, *n* represents one or more digits and *user* represents the user name.

The processes created within and specific to the DECwindows desktop are shown in Table 3–8.

Table 3–8   Process Names of the DECwindows Desktop

Name[1]	Purpose
DECW$LOGINOUT	Login Manager
WAITFORSM_*n*	Holds X connection during Session Manager startup
DECW$SESSION	Session Manager
DECW$MWM	Window Manager
VUE$*user_n*	Spawned process

[1]In these names, *n* represents one or more digits and *user* represents the user name.

Some processes share the same name under both desktops, as shown in Table 3–9.

Table 3–9   Process Names Shared by Both Desktops

Name[1]	Purpose
DECW$TE_*n*	DECterm controller
_FTA*n*	DECterm user process

[1]In these names, *n* represents one or more digits and *user* represents the user name.

(continued on next page)

Table 3–9 (Cont.)   Process Names Shared by Both Desktops

Name[1]	Purpose
_WSA*n*	Placeholder process prior to login or during restart
user_n	Spawned process

[1]In these names, *n* represents one or more digits and *user* represents the user name.

3.11 CDE System Management Documentation

For more detailed information, see the related online help and the online CDE manual *Common Desktop Environment: Advanced User's and System Administrator's Guide.* For instructions on how to access this manual on line or obtain a printed copy, see Table 1–2.

4

Integrating Applications Into the New Desktop

This chapter provides information about the following topics:

- Three levels of desktop integration, defined by the CDE documentation:
  - Minimal (Section 4.1)
  - Recommended (Section 4.2)
  - Optional (Section 4.3)
- Creating new application groups (Section 4.4)
- Registering a third-party application (Section 4.5)
- Porting UNIX reference pages (manpages) to the New Desktop (Section 4.6)
- CDE documentation for integrating applications (Section 4.7)

---------- **Note** ----------

The integration of applications is not required. You can run any existing DECwindows application from the New Desktop without making any changes to it.

4.1 Minimal Integration

Minimal integration enables an application to be launched from the CDE desktop with one or more associated actions. To enable this to happen, certain application configuration files are required. You can achieve minimal integration for most applications by using the Create Action application to create the following configuration files for each application:

- Icon file

- Action and data type definition file

- Action (stub) file

For some applications, you cannot complete minimal integration with Create
Action. For example, you cannot use Create Action to create the action for
an application if the command line requires a nonfile argument, such as a
parameter. Nor can you use Create Action for an application if the data type
must have actions associated with it other than Open and Print. In general,
you can use Create Action to create the files you need for minimal integration
and then edit the files to supply additional information that you cannot supply
with Create Action. In some cases, you must create new files with your editor.

4.1.1 CDE Action Terminology

To understand the process of creating configuration files for integrating
applications, it is important to understand how the following terms are used in
the CDE interface and documentation:

- Action

  Action refers to what happens to an application when you double click on
  its icon in Application Manager or File Manager. The association of the
  application's configuration files with the application files enables this action
  to happen. The Action Name field of Create Action is the name that will
  appear on the icon you select for the application.

- Action definition file

  This file contains the definitions of actions and, optionally, definitions of
  any data types that you want to associate with an application. When this
  file contains data type definitions, it is sometimes referred to as an action
  and data type definition file.

  The action definition file for the Man Page Viewer, DTMANPAGEVIEW.DT,
  follows:

```
ACTION dtmanpageview
{
LABEL           Man Page Viewer
ARG_COUNT       0
ICON            Dthover
TYPE            COMMAND
WINDOW_TYPE     NO_STDIO
EXEC_STRING     cde$system_defaults:[bin]dthelpview -man \
                -xrm "Dthelpview*manBox*columns: 100"
DESCRIPTION     The Man Page Viewer (Dtmanpageview) action \
                displays a man page in a Quick Help viewer window.
}
```

- Action (stub) file

An action (stub) file is a file that the CDE software requires for every action definition file. An action file associates an action definition file with its icon file, making it possible for the icon to be visible within File Manager and Application Manager. An action file has the same name as the action definition file but lacks the DT extension. For example, DTMANPAGEVIEW. is the action file for DTMANPAGEVIEW.DT.

4.1.2 Creating Action Definition Files on OpenVMS Alpha

When creating action definition files for integrating an application, either with the Create Action application or directly with an editor, you must observe the following requirements on OpenVMS Alpha:

- Icon names

  If the action specified for your application will be displayed as an icon in an Application Manager group, the action name must contain only valid OpenVMS file name characters and must be all lowercase. Create Action automatically lowercases the action name for you.

- Command line for running an application

  The Command When Action Is Opened field of the Create Action application produces the EXEC_STRING in the action definition file. This field must contain a valid OpenVMS command line. If the command line begins with $, RUN, or MCR followed by a space, or if it begins with @, the command is processed as a DCL command without interpretation. If the command line begins with a valid OpenVMS or UNIX file specification, a foreign command is automatically generated. In addition, all of the command line options are implicitly quoted in order to preserve their case.

  If the command line begins with only a file name (no device or directory), then the following default execution directories are assumed:

      CDE$USER_DEFAULTS:[BIN]
      CDE$SYSTEM_DEFAULTS:[BIN]

  The logical name CDE$PATH is defined as a search list that points to these directories. If the file does not exist, the command is processed as an uninterpreted DCL command.

- Arguments in the command line

  File arguments in the command line can be specified with Create Action. When you enter arguments using Create Action, you specify with the syntax $$n$$, with $1 representing argument 1, $2 representing argument 2, and so forth in the Command When Action is Opened field. You represent all remaining arguments with $*. (For more information about specifying

arguments, see Chapters 9 and 10 in the *Common Desktop Environment: Advanced User's and System Administrator's Guide*.)

Create Action has the following limitations with regard to specifying arguments:

- It does not support prompt strings without an associated argument.

- It provides a way to enter a prompt string for only the first argument using the Advanced option When Action Opens, Ask User For field.

- It does not provide a way to specify the data type for an argument.

- You cannot include the CWD keyword in your action to set a default directory.

When you specify arguments, you must be sure to use the correct argument type. By default, prompted arguments (syntax: %"*prompt string*"%) are of type String, and explicit arguments (syntax: %Arg_*n*["*optional prompt string*"]%) are of type File. If you do not want the default argument type, you can specify a different type by using either of the following expressions:

%(File)"*prompt string*"%
%(String)Arg_1%

The %Args% keyword is similar to the %Arg_*n*% keyword except that the former is replaced with a list of arguments, separated by spaces, that is passed to the action. Whether or not this is appropriate depends on the specific DCL command.

4.1.3 Create Action Overview

As described in Chapter 9 of the *Common Desktop Environment: Advanced User's and System Administrator's Guide*, Create Action does the following:

- Creates an action definition that runs a command.

- Creates a file *disk*$:([*user*.DT.TYPES]*action-name*.DT. This file stores the action and data type definitions created for the application.

- Creates an action file in the user's home directory. The action file is an executable file with the same name as the action.

  The action file's representation in File Manager is called an application icon. Double clicking on the application icon starts the application.

  When you create the action, you can make the action icon a drop zone by specifying data types that you can drop on the icon (optional).

- Creates one or more data types for the application's data files (optional).

- Creates an Open action for each data type.

- Creates a Print action for each data type (optional).

- Reloads the database of actions and data types. This makes the actions and data types take effect immediately.

4.1.4 Integrating Applications Using Create Action

You can access the Create Action application from the Desktop Tools application group in Application Manager. Figure 4–1 shows the main window of the Create Action application.

Figure 4–1 Create Action Main Window

ZK–8654A–GE
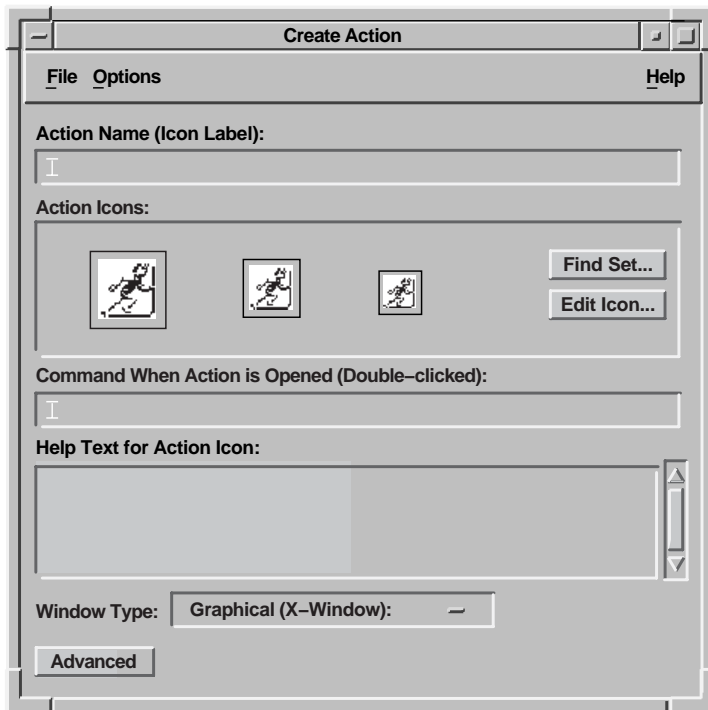
Additional fields are displayed if you select the Advanced option, which is located in the lower left corner of the Create Action main window. These additional fields are shown in Figure 4–2. Table 4–1 describes each field in the Create Action main window and in the expanded (Advanced option) window.

Table 4–1 Create Action Fields

Field Name	Purpose
Initial Fields	
Action Name (Icon Label):	This label is displayed next to the icon for the application.
Action icon:	This icon represents the application. With Find Set..., you can select an icon from the collection that is included with the New Desktop. With Edit Icon..., you can modify an existing icon or create a new one.
Command When Action Is Opened (Double-clicked):	The name of the application you want to access. This name can specify one or more file arguments as operands. The arguments must be specified in the format $filename. For example: DIFF $1 $2.
Help Text for Action Icon:	This help becomes the On Item help for the Action Icon. The text automatically wraps in the text field. You can explicitly add a continuation line by ending a line with a space, a backslash (\), and Return. If the backslash is not present, subsequent lines will not be displayed.
Window Type:	Select the windowing support option required by the application. You have a choice of three types of windows or no window.
Advanced	Select this field if your application uses data files and you want to create one or more data types for them. The Create Action window expands to display additional, advanced fields.
Advanced Option Fields	
When Action Opens, Ask Users for:	Supply a prompt if the application's command line has a required or optional file argument. Otherwise, leave it blank.
Datatypes That Use This Action:	If the action can accept any data type, select All Datatypes. If you want to modify any existing data types or create new ones, select Only Above List and then select Add. (Initially, the Datatypes That Use This Action list is empty.) When you select Add, a dialog box called Add Datatypes is displayed.

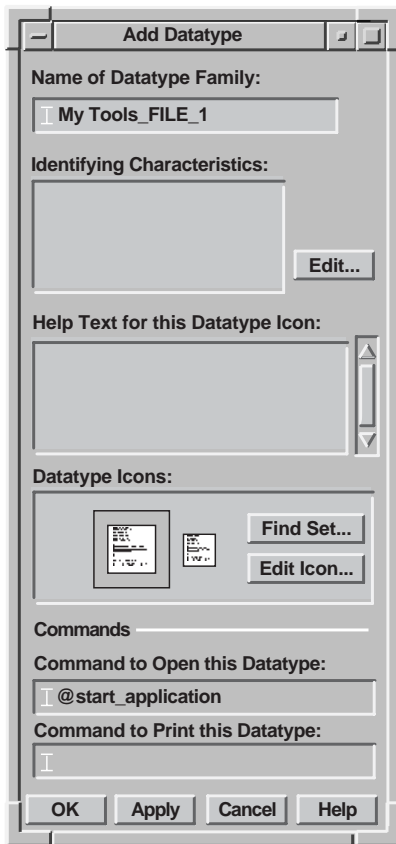**Figure 4–2 Create Action Main Window With Advanced Option Fields**
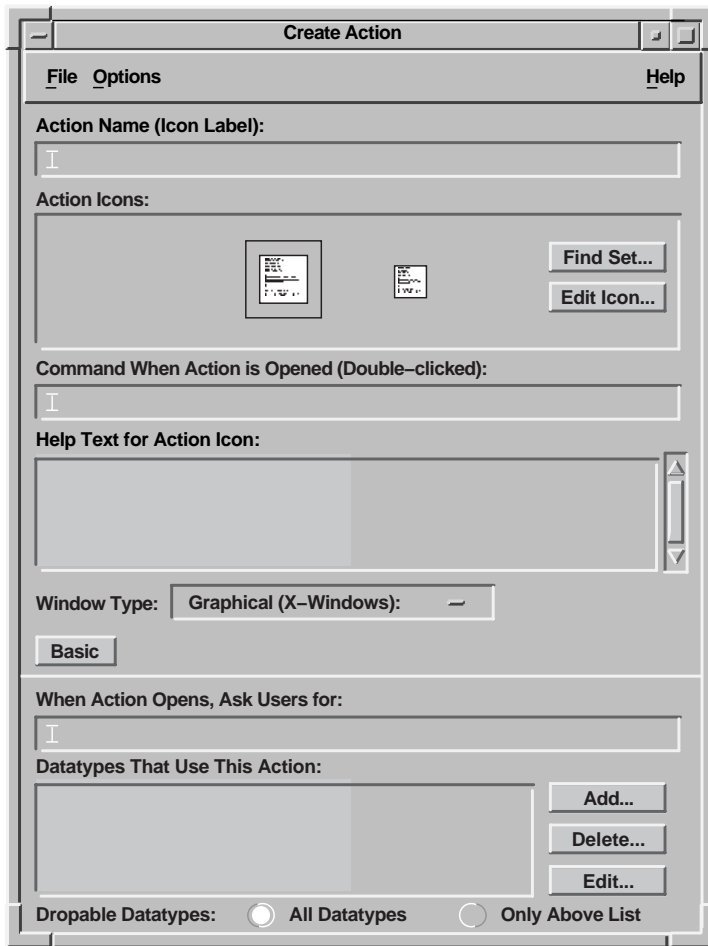
ZK–8655A–GE

4.1.5 Creating or Modifying Data Types Using Create Action

The Create Action application provides the Add Datatype dialog box for
modifying an existing data type or creating a new one. The Add Datatype
dialog box is shown in Figure 4–3, and the fields are described in Table 4–2.

Figure 4–3 Add Datatype Dialog Box

ZK–8656A–GE

Table 4–2 Create Action: Add Datatype Dialog Box

Field	Purpose
Name of Datatype Family:	If you want to modify an existing data type and retain the same name, leave this field blank. If you want to create a new data type, supply a name. (The name cannot include spaces. The name is not visible to application users; it is used in the actions/data types database to identify the data type definition.)
Identifying Characteristics:	To display the Identifying Characteristics dialog box, click on the Edit button.
Help Text for this Datatype Icon:	This help becomes the On Item help for the Action Icon.
Datatype Icons:	With Find Set..., you can select an icon from the collection that is included with the New Desktop. With Edit Icon..., you can create a new icon or modify an existing one.
Command to Open this Datatype:	This command is the same that you supplied for Command When Action is Opened (Double-clicked): and will appear in this field.
Command to Print this Datatype:	Enter the Print Dialog command line to suit the application data type.

After you respond to the second prompt, Identifying Characteristics, by clicking on the Edit... button, the Identifying Characteristics dialog box is displayed, as shown in Figure 4–4. The fields of this dialog box are described in Table 4–3.

The characteristics are displayed in the Identifying Characteristics field of the Add Datatype dialog box, using the following codes:

- d — A directory.
- r — The file has read permission.
- w — The file has write permission.
- ! — Logical operator NOT.
- & — Logical operator AND.

Figure 4–4 Identifying Characteristics Dialog Box
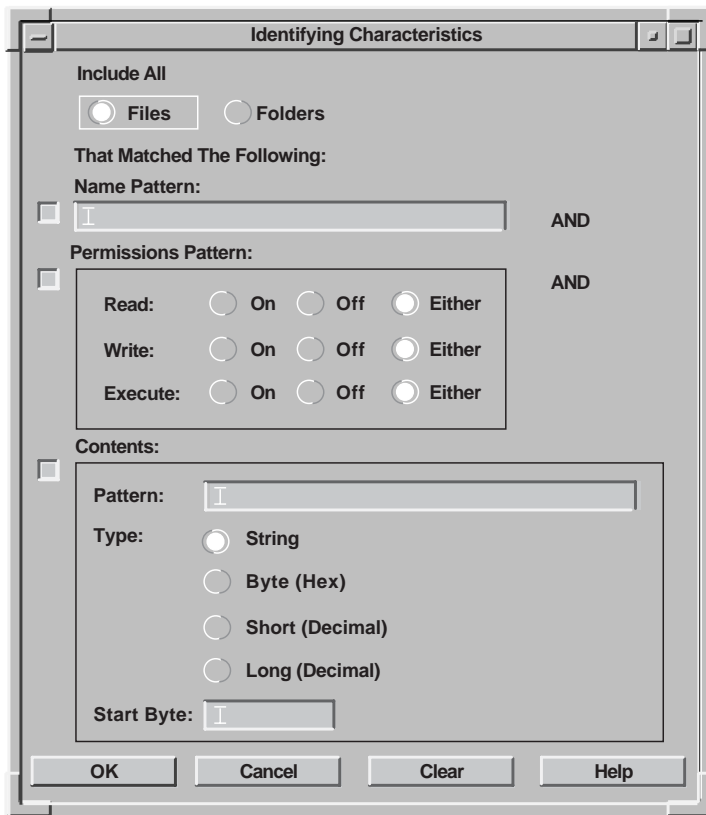
ZK–8657A–GE

Table 4–3 Create Action: Identifying Characteristics Dialog Box

Field	Purpose
Files, Folders	The data type applies only to files or only to folders.
Name Pattern:	Data typing based on the file name.
Permissions Pattern:	Read, write, execute, or delete permission. The Either option for these permissions means that the permission does not matter. The permissions testing does not recognize OpenVMS access control lists (ACLs).
Contents:	A pattern to search for, the type of contents, and an optional starting point in the file. Note that this feature can carry a substantial performance penalty when changing or updating views.

4.1.6 Editing Icons From the Create Action Application

If you invoke the Icon Editor from the Create Action application and edit an icon file, the edits that you make are not automatically shown in the displayed icon when you exit from Icon Editor. After exiting from Icon Editor, you must choose Find Set... and select the newly edited icon to change the display and associate the edited icon with the action.

If you want to edit more than one size of the icon, make all the edits before you select Find Set. If you do not have a complete set of edited icons when you select Find Set..., the missing icons will not be displayed and you will not be able to create them from within the Create Action application.

4.2 Recommended Integration

Recommended integration means that an application is integrated with other CDE components, making it more consistent with the CDE desktop. Recommended integration includes making the following enhancements to your application:

- Providing convenience functions in an application, such as the drag-and-drop function and the save/restore function (see Chapter 5)

- Providing for the internationalization of an application

- Displaying CDE style error messages

- Using the fonts provided with CDE (see Chapter 5)

- Incorporating CDE style help (see Chapter 5)

Recommended integration will likely require code changes. This section briefly describes the resources for providing user interfaces in different languages and CDE style error messages. Chapter 5 describes the programming resources for providing the other enhancements.

4.2.1 Internationalization

The New Desktop provides user interfaces in different languages. The language and other culture-specific attributes, such as the symbols and rules for formatting monetary numeric information and dates, are defined in a file called a locale. Each locale is supported by a message catalog, which contains all the text an application will output for a particular language, and by resource files. The DEC C XPG4 localization utilities, briefly described in Table 4–4, are used to create locale files and message catalogs.

Users can specify the language of the user interface when they log in. From the login display, select Options, then select Language. The languages installed on your system, as defined in the locale files, are displayed. You can then select the language for the user interface on your system.

Note

The default locale is the C locale. The notation *.lang* is used in file specifications in this manual to indicate the locale variable. In the CDE documentation, %L and <lang> are used to indicate the locale variable.

Table 4–4   DEC C XPG4 Localization Utilities

Utility	Purpose
GENCAT	Merges one or more message-text source files into a message catalog file. The message-text source files use a default file type of .MSG. The message catalog file uses a default file type of .CAT. The GENCAT syntax is: `GENCAT msgfile [,...] catfile`
ICONV	Depending on the option that you specify, ICONV either compiles conversion table files (ICONV COMPILE) or converts characters from one codeset to another (ICONV CONVERT). A conversion table file defines how characters in one codeset are converted to characters in another codeset.
LOCALE	Depends on the option that you specify: • LOCALE COMPILE compiles a binary locale file. • LOCALE LOAD loads a locale name into system memory as shared, read-only global data. • LOCALE SHOW displays details of locales on your system. • LOCALE UNLOAD unloads a locale name from system memory.

The locale files are in the directory defined by the logical name SYS$I18N_LOCALE. For more information about the DEC C XPG4 localization utilities, see the *OpenVMS Version 6.2 New Features Manual* or the *OpenVMS Version 7.0 New Features Manual*.

4.2.2 Displaying Error Messages From Applications

New Desktop applications follow a common model for presenting error messages and warnings. Users expect messages to be displayed in message footers, error dialog boxes, or warning dialog boxes, with further explanations in online help, when appropriate. For details about displaying error messages in applications and linking message dialogs to online help, see the *Common Desktop Environment: Programmer's Guide*.

4.3 Optional Integration

Optional integration enables you to leverage CDE services to perform specialized tasks, such as providing certain graphical user interface functions and managing multiple workspaces. Custom widgets provide the specialized graphical user interface functions. The Workspace Manager API provides the functions for managing multiple workspaces. For more information about these CDE services, see Chapter 5 and Appendix B of this guide, the *Common Desktop Environment: Programmer's Overview*, and the *Common Desktop Environment: Programmer's Guide*.

4.4 Creating New Application Groups

The New Desktop includes four application groups: DECwindows Apps, DECwindows Utilities, Desktop Apps, and Desktop Tools, as shown in Figure 4–5. Desktop Apps refers to the CDE applications and Desktop Tools refers to the CDE tools.

An application group is simply a directory containing action (stub) files. For Application Manager to recognize that a directory is an application group, the directory must reside in one of the application group root directories defined by the logical name DTAPPSEARCHPATH. The directory *disk*$:[*user*.DT.APPMANAGER] is one of the application group root directories.

You can create additional application groups to be managed by Application Manager in the following way:

1. Create a directory with the name of a new application group, for example, *disk*$:[*user*.DT.APPMANAGER.MY_TOOLS].

2. The new application group will initially appear using the folder icon and the default name (in lowercase, with the underscore), as shown in Figure 4–6. To change the folder icon and the default name, copy the file NEWGROUP.DT_TEMPLATE from CDE$SYSTEM_DEFAULTS:[APPCONFIG.TYPES.C] to *disk*$:[*user*.DT.TYPES]*newgroup*.DT,

Figure 4–5 Application Manager

ZK–8652A–GE
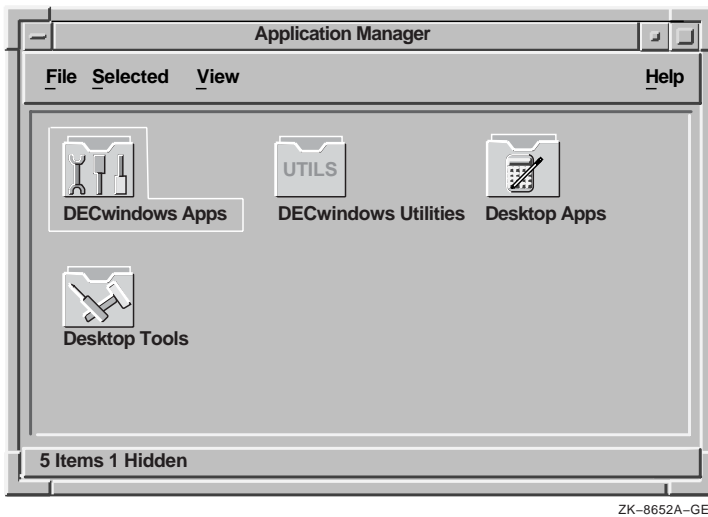
if one doesn't exist already. Modify this file following the instructions contained within the file.

3. Run the Create Action application located in the Desktop Apps group to create:

   a. Action (stub) file, for example, SYS$LOGIN:CALCULATOR

   b. Action and data type definition file, for example, *disk*$:[*user*.DT.TYPES]CALCULATOR.DT

   You must fill in a minimum of two fields: Action Name (Icon Label) and Command When Action Is Opened (Double-clicked). For this example, the action name is CALCULATOR.DT and the command when action is opened is RUN SYS$SYSTEM:DECW$CALCULATOR.

4. Provide an icon for your application in one of the following ways:

   • Accept the default icon (the runner) by not selecting either Find Set... or Edit Icon....

   • Select an icon that is included with the New Desktop by choosing Find Set... and responding to the prompts.

   • Revise an existing icon or create a new one by choosing Edit Icon... and responding to the prompts.

   For this example, the default icon is used.

5. Move the action (stub) file from SYS$LOGIN to
    disk$:[*user*.DT.APPMANAGER.MY_TOOLS].

6. Enable the new application group within Create Action with the Save
    option on the File pull-down menu. Alternatively, you can exit from Create
    Action and enable the new application group with Reload Actions. From
    Application Manager, select the Desktop Tools group, then double click on
    Reload Actions.

7. Select Application Manager.

    You will see a new file folder icon labeled my_tools, as shown in Figure 4–6.
    Double click on this icon and you will see the action icon (a runner) with
    the name Calculator. Double click to start Calculator.

8. Repeat steps 3 through 6 for every application you want to add to a group.

Every time you use this procedure, a new .DT file is added to your
disk$:[*user*.DT.TYPES] directory. You might want to create one file, such
as MY_TOOLS.DT, and merge all the actions into this file. A single file can
simplify the management of an application group.

Figure 4–6 Application Manager With a New Application Group

ZK–8653A–GE

4.5 Registering a Third-Party Application

Registering an application is the phrase used in CDE documentation for integrating and installing an application into the New Desktop. The files required for integrating an application are called a registration package. Registering an application is described in Chapter 4 of the *Common Desktop Environment: Advanced User's and System Administrator's Guide*.

The process on UNIX systems differs from the process on OpenVMS Alpha systems. The most significant difference is that the CDE application dtappintegrate is not included with the New Desktop. However, the rest of the process on the New Desktop is similar.

An application is considered fully registered into the New Desktop when it has the following components:

- Its own application group at the top level of Application Manager.

- An action that starts the application. The action is represented by an icon in the application group.

- Optionally, data types for its data files.

The registration package consists of the following components:

- Files that are created for integrating an application into the New Desktop. The number and types of files will vary, depending on the type of application and the level of integration you choose (minimal, recommended, or optional).

- A new directory file and associated files for a new application group to be placed in the CDE$USER_DEFAULTS:[APPCONFIG.APPMANAGER.C] directory, as shown in Figure 4–7.

The installation procedure for a registration package must perform the following tasks:

- Test to determine that the appropriate directories are present in the CDE$USER_DEFAULTS:[*...] hierarchy. Files of the following file types should be placed in this hierarchy (see Figure 4–7):
  - *.DT
  - *.FP
  - *.SDL
  - *.HV
  - *.PM

- *.BM

- Action files

- Readme files

  If the directories are not present, then the installation procedure must create them.

- Install the files for the new application group that you created for one or more applications that you are installing into that group.

- Install the application integration files that you created for each application into the appropriate directories in the CDE$USER_DEFAULTS hierarchy.

- Install the application executable files, shareable images, and resource files into their respective locations: SYS$SYSTEM, SYS$SHARE, and CDE$USER_DEFAULTS.

Figure 4–7   CDE$USER_DEFAULTS Directory Hierarchy

```
                       CDE$USER_DEFAULTS:[000000]
                                 |
                           [.APPCONFIG]
                                 |
      ┌───────────────┬──────────────────┬──────────────────────────┐
   [.TYPES]        [.HELP]            [.ICONS]                  [.APPMANAGER]
   ┌────┴────┐     ┌────┴────┐        ┌────┴────┐               ┌────┴────┐
 [.C]    [.LANG]  [.C]   [.LANG]    [.C]    [.LANG]           [.C]    [.LANG]
  |               |                  |                          |
 *.DT            *.SDL              *.*PM                  [.appsgroupname]
 *.FP            *.HV               *.*BM                        |
                                                  ┌─────────────┼──────────────┐
                                         action_stub_file.  action_stub_file.  action_stub_file.
```
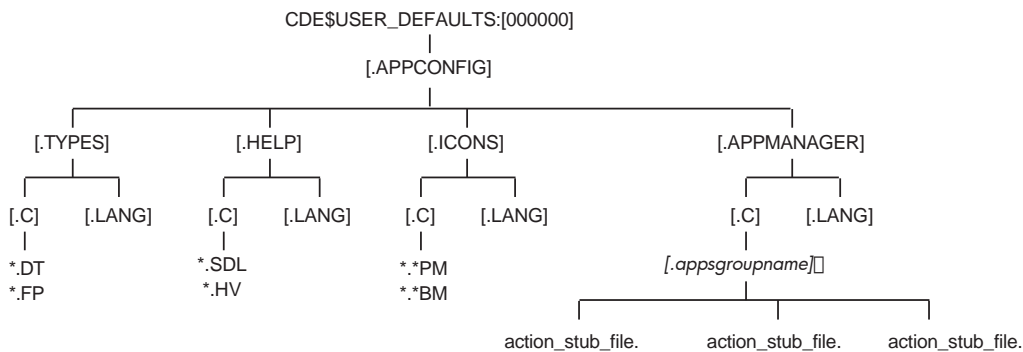
ZK-8658A-AI

After your installation has completed and the session has restarted, you can then access the new application group and the applications in it with the following steps:

1. Single click on the icon representing Application Manager.

2. Double click on the icon representing the new application group that you installed. An icon for each application in the group will be displayed.

3.   Double click on the application you want to start.

4.6 Porting UNIX Reference Pages to the New Desktop

If you have created reference pages (manpages) on a UNIX system that you would like to add to the reference pages on the New Desktop, you can do so in the following way:

- On the UNIX system, format the reference page and capture the output as follows:

```
% man dtwmrc > dtwmrc.1
```

- Copy the formatted file to the OpenVMS system. Use dcp, ftp, or another copy command.

  The following example shows copying with dcp:

```
% dcp dtwmrc.1 node"user password"::'cde$system_defaults:[man]'
```

  The following example shows copying with ftp:

```
ftp node
ftp> cd cde$system_defaults:[man]
ftp> put dtwmrc.1
ftp> bye
```

After you have copied the reference page, you can view it by selecting the Man Page Viewer from the Desktop Apps group of Application Manager.

4.7 CDE Documentation for Integrating Applications

For more detailed information about integrating applications, see the related online help, especially the Create Action help, and the following CDE manuals:

- *Common Desktop Environment: Programmer's Overview*

- *Common Desktop Environment: Advanced User's and System Administrator's Guide.*

For instructions on how to access either manual on line or obtain printed copies, see Table 1–2.

5

New Desktop Programming Resources

DECwindows Motif for OpenVMS Alpha provides the same programming
resources as DECwindows Motif for OpenVMS VAX, as well as programming
resources for the New Desktop. These include a subset of CDE APIs and
integration services and additional CDE resources.

This chapter provides information about the following topics:

- Standards supported by the New Desktop (Section 5.1)

- APIs and application integration services (Section 5.2)

- Save and restore support (Section 5.3)

- Create Action application (Section 5.4)

- File type support (Section 5.5)

- CDE standard fonts (Section 5.6)

- Help programming (Section 5.7)

- Header files (Section 5.9)

- CDE example programs (Section 5.10)

- CDE programming documentation (Section 5.11)

5.1 Standards Supported by the New Desktop

The New Desktop supports the following standards:

- CDE Motif 1.0 toolkit (OSF/Motif Release 1.2.5)

  The New Desktop provides run-time support for the CDE Motif 1.0,
  OSF/Motif Release 1.1.3, and XUI toolkits. DECwindows applications built
  with these toolkits will run under the New Desktop.

- X11 R5

- UIL compiler

Although the UIL compiler is supported, CDE applications do not use it. Therefore, no .uid files are associated with any of the CDE applications.

- Motif Resource Manager (Mrm)

- Inter-Client Communication Conventions Manual (ICCCM)

- CDE programming libraries (see Section 5.2)

5.2 APIs and Application Integration Services

Table 5–1 lists the subset of CDE application programming interfaces (APIs) and the CDE application integration services available with the New Desktop. They are in the following shareable libraries (pointed to by the logical SYS$LIBRARY):

- CDE$LIBDTSVC.EXE

- CDE$LIBDTHELP.EXE

- CDE$LIBDTWIDGET.EXE

The corresponding header files are also shown in Table 5–1. For a list of all CDE API routines, see Appendix C. Most of the CDE API routines are supported on the New Desktop, but a small subset are not. Those not supported are noted.

In addition to these new APIs and application integration services, all the DECwindows Motif APIs and application integration services are available in the New Desktop.

Table 5–1 CDE APIs and Application Integration Services of the New Desktop

Contents	Description
CDE$LIBDTSVC.EXE	**Desktop Services**
Drag-and-drop convenience API	Convenience functions layered on top of the OSF/Motif R1.2 drag-and-drop services. These functions simplify the use of the OSF/Motif R1.2 drag-and-drop services, which have not changed.
Action invocation API	Used to initialize and load the database of action and data type definitions, query the action database, invoke an action, and register a callback to receive action status and return arguments.

Table 5–1 (Cont.) CDE APIs and Application Integration Services of the New Desktop

Contents	Description
CDE$LIBDTSVC.EXE	**Desktop Services**
Session Manager API (dtsession)	Services include save and restore.
Workspace Manager API (dtwm)[1]	Gets workspace information, sets the current workspace, and so on.
Header files[2]	Action.h, Dnd.h, Dts.h, Dt.h, Saver.h, Session.h, and Wsm.h.

For more information, see the *Common Desktop Environment: Programmer's Guide.*

Contents	Description
CDE$LIBDTHELP.EXE	**Help Widget Support**
DtHelpDialog	General help dialogs that include a menu bar, topic tree, and a help topic display area.
DtHelpQuickDialog	Quick help dialogs that include a topic display area and one or more dialog buttons at the bottom.
Header files[2]	Help.h, HelpQuickD.h, and HelpDialog.h.

For more information, see the *Common Desktop Environment: Help System Author's and Programmer's Guide.*

Contents	Description
CDE$LIBDTWIDGET.EXE	**Custom Widgets**
DtSpinBox	Combines a text field and arrow buttons in a control that can increment or decrement numeric or text values.
DtComboBox	Combines a text field and a list box in a control that displays one or more valid choices for the text field.
DtMenuButton	Provides menu cascading functionality outside the menu bar.
DtEditor	Incorporates the functionality of a simple text editor.
Header files[2]	SpinBox.h, ComboBox.h, MenuButton.h, and Editor.h.

For more information, see the *Common Desktop Environment: Programmer's Guide.*

[1]Some functions are not supported.

[2]Found in DECW$INCLUDE:.

Table 5–2 lists the CDE APIs, application integration services, and tools that are not included in the New Desktop.

Table 5–2 CDE APIs and Application Integration Services Not Included

CDE Component	Name
Application Builder	dtappbuilder
Calendar Manager API	libcsa
Messaging services (ToolTalk)	libtt
Multipurpose Internet Mail Extensions (MIME) support	libdtmail
Terminal emulator API	libdtterm
Windowing Korn shell	dtksh

For examples of how these routines are used in applications, see the example programs, as described in Section 5.10.

5.3 Save and Restore Support

The New Desktop provides session save and restore utility functions. If these are integrated into your application, the application state is saved upon exiting Session Manager and is restored when the session is restarted.

Use the XmAddWMProtocol() function to set the WM_SAVE_YOURSELF atom on the WM_PROTOCOLS property for the top level of the application. You can view window properties by using the xprop utility program located in the DECW$UTILS directory.

Use the XmAddWMProtocolCallback() function to declare a callback procedure for when the application receives a WM_SAVE_YOURSELF client message. Session Manager sends a WM_SAVE_YOURSELF client message to your application, and the callback function is called.

The application should not interact with the user during save or restore. For example, a dialog box should not be displayed during save or restore.

An example program, DTSESSIONEXAMPLE.C, is provided as a guide. The example program creates a main window with a toggle button and saves the state of the button during execution of the save/restore routine.

The example program is available in the following location:

```
CDE$SYSTEM_DEFAULTS:[EXAMPLES.DTSESSION]
```

For general information about the example directories, see Section 5.10.

5.4 Create Action Application

To enable an application to be launched from the New Desktop, certain application configuration files must be created and installed. You can create and install the configuration files for most applications by using the Create Action application, which automates the process of creating these files. You can access the Create Action application from the Desktop Tools application group in Application Manager.

For more information about Create Action, see Section 4.1.

5.5 File Type Support

The New Desktop contains a set of file types with preconfigured icons and associated actions. The icons and actions can be modified with Create Action, the action database and execution management tool.

Examples of file types are executable files (with the file extension .EXE) and data files (with the file extension .TXT). Examples of actions include *running* a file if it is an executable file, and *printing* and *opening* (in order to edit) a file if it is a data file.

5.6 CDE Standard Fonts

CDE defines two types of standard fonts: application fonts and interface fonts. Use the application fonts for application data. Use the interface fonts with the Motif widgets.

The CDE standard application fonts supplied with the New Desktop are names of generic fonts that are available on all CDE platforms. They are mapped to different fonts on different CDE platforms, typically using the X font alias mechanism. You specify the fonts for these resources in your APP-DEFAULTS file.

The CDE standard interface fonts ensure that, for any application development that you do with Motif widgets, the application's windows look like other CDE desktop client windows, and users can change the size of these fonts using Style Manager. If you override the supplied fonts by changing the Motif font list resource specifications, then you must provide additional functionality if you want users to be able to customize the fonts in your application.

By observing these guidelines, your application will find the appropriate fonts on all CDE platforms. For more information, see the *Common Desktop Environment: Programmer's Overview* and the *Common Desktop Environment: Programmer's Guide*.

5.7 Help Programming

The New Desktop help system is a complete system for developing and displaying online help for application software. This section briefly describes the process and tools that are provided for developing help. The aspects that are unique to the OpenVMS implementation are fully documented here. For more information about the process, the SGML markup language, the writing tasks, and the programming tasks that comprise this process, see the *Common Desktop Environment: Help System Author's and Programmer's Guide*.

The following New Desktop programming components are provided for developing help:

- Help compiler interface (DTHELPTAG.COM)
- Help utility functions (CDE$LIBDTHELP.EXE)
- HelpTag examples

5.7.1 Help Development Process

Online help is written in text files whose file extension is .HTG. The SGML markup language named HelpTag is used to specify how elements of the text are formatted.

HelpTag consists of a set of tags that specify the formatting of elements such as headings, paragraphs, and lists. After composing the help text and labeling the elements of the text with the appropriate tags, you process these tagged files with the DTHELPTAG.COM command procedure. DTHELPTAG.COM produces run-time help files (with the extension .SDL) that you can view with the Help Viewer.

Two styles of SGML can be used with the HelpTag markup language: shorthand SGML or fully compliant SGML. The shorthand style is simpler and quicker to code.

5.7.2 Help Compiler Interface

The Help compiler interface consists of three executable files that are invoked from DTHELPTAG.COM. The executable files, described in Table 5–3, are in the CDE$SYSTEM_DEFAULTS:[BIN] directory, which is referenced by the CDE$PATH logical name.

Table 5–3 DTHELPTAG.COM Executable Files

File	Purpose
DTHELP_CTAG1.EXE	Parser for fully compliant SGML-based help topics.
DTHELP_HTAG1.EXE	Parser for shorthand SGML-based help topics.
DTHELP_HTAG2.EXE	Runs after either parser for enhancement of distribution file.

The main function of DTHELPTAG.COM is to compile a help source file. DTHELPTAG.COM can also be used for the following tasks:

- Removing all intermediate files generated by DTHELPTAG.COM

- Listing all the files associated with a help volume

DTHELPTAG.COM was ported from the UNIX version of the dthelptag shell script. DTHELPTAG.COM supports all the functions found in the UNIX version of this utility except file compression and decompression.

5.7.3 DTHELPTAG.COM Format and Options

The format for DTHELPTAG.COM is:

@CDE$PATH:DTHELPTAG [*options*] *filename*

where *filename* specifies the name of the input help source file.

The DTHELPTAG.COM options, which require a leading hyphen (-), are described in Table 5–4.

Table 5–4 DTHELPTAG.COM Options

Option	Purpose
-formal	Specifies the parser for fully compliant SGML-based help topics. If you do not specify -formal, the parser for the shorthand SGML-based help topics is used.
-nooptimize	Eliminates certain optimizations that normally take place during translation of HelpTag markup to the run-time format. Using this option speeds the translation process.
-memo	Includes a writer's memos entered with the memo tag (<memo>) in the help file.
-debug	Saves the intermediate files; otherwise, they are deleted.

Table 5–4 (Cont.)   DTHELPTAG.COM Options

Option	Purpose
-clean	Removes all files generated by DTHELPTAG.COM for a particular help volume.
-files	Lists all the files associated with for a particular help file. The files include intermediate files generated during the DTHELPTAG.COM command procedure and graphics files included in the help file.
-help	Invokes command line help for DTHELPTAG.COM.

5.7.4 Help Utility Functions

For application programmers, the New Desktop provides a help API that allows the addition of help dialog boxes to any Motif application. The New Desktop help library (CDE$LIBDTHELP.EXE) provides the following two types of help dialog boxes:

- General help dialog boxes (DtCreateHelpDialog())

  General help dialog boxes have a menu bar, a topic tree, and a help topic display area.

- Quick help dialog boxes (DtCreateQuickHelpDialog())

  Quick help dialog boxes have only a topic display area and one or more dialog buttons located at the bottom of the display area.

For more information on the New Desktop help API, refer to the *Common Desktop Environment: Help System Author's and Programmer's Guide*.

5.7.5 HelpTag Examples

HelpTag example files are in the following directory:

```
CDE$SYSTEM_DEFAULTS:[EXAMPLES.DTHELP]
```

The following example uses an example file to illustrate the -files HelpTag option:

```
$ @CDE$PATH:DTHELPTAG -files HELPDEMO.HTG
```

The output from this command follows:

```
Workfile is --> HELPDEMO.HTG
MARKUP PARSER (with non-canonical SGML enhancements)
(c) Copyright 1993, 1994 Hewlett-Packard Company
(c) Copyright 1993, 1994 International Business Machines Corp.
(c) Copyright 1993, 1994 Sun Microsystems, Inc.
(c) Copyright 1993, 1994 Unix System Labs, Inc., a subsidiary of Novell, Inc.
CDE HelpTag Formatting System. - Version B.00.00  (DTD Version A.01.23)
Interface generated from help.if on Thu Dec 28 09:01:10 1995
helpdemo.htg
Warning: could not open locale translation database.
[.graphics]AppWithHelp.xwd
[.graphics]cauticon.pm
[.graphics]clock.xwd
[.graphics]clouds.xpm
[.graphics]bee.xwd
[.graphics]deadjim.xwd
[.graphics]GeneralHelp.xwd
[.graphics]head-down.xwd
[.graphics]helpchar.ent
[.graphics]helpicon.ent
[.graphics]helplang.ent
[.graphics]helpShelf.pm
[.graphics]integral.bm
[.graphics]noteicon.pm
[.graphics]QuickHelp.xwd
[.graphics]rooster.xpm
[.graphics]Snapshot.bm
[.graphics]Snapshot.xwd
[.graphics]speaker.pm
[.graphics]shuttle2.xwd
[.graphics]sunset.xwd
[.graphics]tribe.xwd
[.graphics]warnicon.pm
[.graphics]xload.xwd
Exiting HelpTag Utility...
$
```

5.8 Screen Savers

You can create additional screen savers for the New Desktop. An example
screen saver is provided in CDE$SYSTEM_DEFAULTS:[EXAMPLES.DTSCREEN].

Creating a screen saver involves the following steps:

1. Create a screen saver application.

2. Create an action to invoke the screen saver application.

3. Add the new screen saver action to the list of available screen savers. This is done by adding the following line to your SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM file:

```
DTSCREENSAVERLIST == "SampleScreenSaver"
```

   Use spaces to separate screen saver names in the list.

4. Restart your session.

After completing these steps, the new screen saver appears in the list of available screen savers in the Screen option of Style Manager.

For more information about creating new screen savers, see the README. file in the examples directory.

5.9 Header Files

Header files for the CDE components, shown in Table 5–5, are supplied with the New Desktop. They enable application developers to take advantage of the Desktop APIs (drag and drop, save and restore, Workspace Manager, and so forth), Help services, and custom widgets.

The header files are in the DECW$INCLUDE directory. They are referenced in the source files of the examples directory with the following syntax:

#include <DT/*filename*.H>

where DT is a logical name that is defined as DECW$INCLUDE.

_____ **Note** _____

Include must be specified in lowercase, and the brackets surrounding the include phrase must be angle brackets.

_____

Table 5–5 CDE Header Files

File	Function
Desktop Services APIs	
DT/ACTION.H	Action-related structures and functions
DT/DND.H	Drag-and-drop functions

(continued on next page)

Table 5–5 (Cont.)   CDE Header Files

File	Function
Desktop Services APIs	
DT/DT.H	CDE version information, DtInitialize, and DtAppInitialize functions
DT/DTS.H	Data typing constants and functions
DT/SAVER.H	Screen saver API functions
DT/SESSION.H	Session Manager API (save and restore) functions
DT/WSM.H	Workspace Manager API related data and functions
Help Services	
DT/HELP.H	Define functions for dtfile help
DT/HELPQUICKD.H	Quick Help dialog resources and functions
Custom Widgets (available in libdtwidget.exe)	
DT/COMBOBOX.H	For combo box widget
DT/EDITOR.H	For editor widget
DT/MENUBUTTON.H	For menu button widget
DT/SPINBOX.H	For spin box widget

5.10 CDE Example Programs

Several CDE example programs are included in the New Desktop and are
described in Table 5–6. The example programs show how to use the various
CDE APIs and other programming resources provided with the New Desktop.
Each example directory contains a readme file (README.) that describes the
example programs and a command file (*nnnn.com*) that can be used to build
the example programs for that directory.

The top-level example directory can be referenced by using the
CDE$EXAMPLES logical, as shown in the following example:

```
$ DIR CDE$EXAMPLES

 Directory CDE$SYSTEM_DEFAULTS:[EXAMPLES]

 DTACTION.DIR;1      DTDTS.DIR;1         DTHELP.DIR;1
 DTSCREEN.DIR;1      DTSESSION.DIR;1     DTWIDGET.DIR;1
 DTWSM.DIR;1
```

The header files for these example programs are included in the form:

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

where DT is a logical that is defined as DECW$INCLUDE.

Table 5–6 CDE Example Programs

Name	Purpose
DTACTION	Shows the use of the action API to invoke actions on a file. The application displays two text entry fields. Enter an action name in the first field and the name of the file to which the action is to be applied in the second field; then press Return.
DTDND	Demonstrates how to use the drag-and-drop functions so that an application can be moved by dragging and dropping its icon. The example program consists of a row of three different sources for text, file name, and application-named data drags. It also has a type-in field that can accept either text or file name drops and a data area that accepts file-name or data drops.
DTDTS	Demonstrates the use of the Dts data typing API. This program displays the data type, icon name, and supported actions for each file passed to it. The dtaction client can be used to execute a supported action on the file.
DTHELP	Illustrates the use of the CDE help system's HelpTag language and the building of a CDE help file. You can invoke the Help Viewer to display the compiled help file (HELPDEMO.SDL).
DTSCREEN	Contains an example program (SCREENSAVER.EXE) of the screen saver API. It is an example of a simple screen saver that uses the screen saver API and certain techniques to make the screen saver available to all desktop users or your own session. (Alternatively, you can select a screen saver from the set provided with the New Desktop for use in desktop sessions. You can preview and select these screen savers from the Style Manager Screen dialog box.)
DTSESSION	Demonstrates the session mechanism and API. The program SESSION.EXE is an example of an application that supports the Dt session management protocol using the DtSession API. The application saves its current state (the value of a toggle button) when the session is terminated. When the session is restarted, the state of the toggle button is restored. (For more information about save and restore, see Section 5.3.)

Table 5–6 (Cont.)   CDE Example Programs

Name	Purpose
DTWIDGET	Contains demonstrations of the widget library. The CONTROLS.EXE program demonstrates DtSpinBox, DtComboBox and DtMenuButton controls. The EDITOR.EXE program demonstrates the DtEditor widget.
DTWSM	Contains demonstrations of the Workspace Manager API. The OCCUPY.EXE program demonstrates how to set and query an application's presence in CDE workspaces. The WSINFO.EXE program demonstrates how to query for information about the attributes of an application's current workspace.

5.11 CDE Programming Documentation

The CDE programming documentation includes the CDE help system with context-sensitive help, online CDE manuals, and online reference pages (also known as manpages). Printed CDE manuals are also available.

5.11.1 CDE Programming Manuals

The following CDE programming manuals are available on line:

- *Common Desktop Environment: Programmer's Overview*

- *Common Desktop Environment: Programmer's Guide*

- *Common Desktop Environment: Help System Author's and Programmer's Guide*

- *Common Desktop Environment: Internationalization Programmer's Guide*

- *Common Desktop Environment: Style Guide and Certification Checklist*

For instructions on how to access these manuals or obtain printed copies, see Table 1–2.

5.11.2 Reference Pages

CDE reference pages (manpages) are provided on the kit as an installation option. Some of the commands described in the reference pages are not implemented in the New Desktop.

The reference pages are divided into sections. On OpenVMS Alpha, the file extension is used to indicate the section type, as shown in Table 5–7.

Table 5–7 Reference Page Sections

Section	Section Type	Section Number[1]
1	Applications	*filename*.1
3	Libraries/programming	*filename*.3
4	Programming	*filename*.4
5	Include file formats	*filename*.5

[1]A reference page section of .2, which applies to CDE system calls, does not exist in the New Desktop.

For instructions on how to access the reference pages, see Table 1–2.

A

New Desktop and CDE Differences

This appendix describes the most significant differences between the New Desktop on OpenVMS Alpha and CDE on UNIX systems. The New Desktop provides an environment that looks similar to any CDE environment, except for several components that are not included on OpenVMS Alpha (see Table 5–1).

The differences between the common components are primarily due to differences in the operating systems. For example, the file name syntax is different, and logical names are used by the New Desktop in place of environment variables.

A.1 Overall Differences

File names are always displayed and accepted in OpenVMS syntax. However, UNIX path specifications can be used as input to any New Desktop application. For example, /sys$manager/login.com is equivalent to SYS$MANAGER:LOGIN.COM.

Some CDE files are expected to appear in multiple places on UNIX systems through the use of symbolic links. With the New Desktop, files only appear in a single location. There is no dtappintegrate application to create symbolic links in the system directories for application-specific files residing in application-specific directory trees. New application files must be placed in the CDE$USER_DEFAULTS:[*...] system directory.

File names on OpenVMS are case insensitive. When a file name is used in other contexts, such as in the name of an action used to match an action (stub) file or in the description resource names for palettes and backdrops, the reference to the file name must be lowercase.

A.2 Login Manager Differences

Although the user interface for the login process is basically the same on the New Desktop as on CDE, the implementation of the Login Manager dtlogin on OpenVMS Alpha is different. The main differences are:

- Login Manager on OpenVMS can manage only a single X display. Login Manager does not create a separate subprocess to manage its display, and the Login Manager process exits once the user logs in.

- The New Desktop does not provide a chooser application to select alternate systems on which to run your session.

- The New Desktop does not provide a Login Manager image. Instead, the shareable image CDE$SYSTEM_DEFAULTS:[BIN]DECW$LOGINOUT.EXE is dynamically activated by SYS$SYSTEM:DECW$LOGINOUT.EXE, the OpenVMS login process. A DECwindows or New Desktop login is initiated by the command RUN SYS$SYSTEM:DECW$STARTLOGIN.

- Some of the scripts used with CDE, including Xsetup, Xstartup, Xreset, and dtprofile, are not supported in the New Desktop. Instead, custom configuration information is defined in the SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM and SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.COM procedures.

- In the New Desktop, most systemwide environment variables are supported as logical names. However, the user variants of the CDE environment variables (XMUSER* and DTUSER*) are not supported.

- The login transitional greeting application, dthello, which displays "Starting the New Desktop for OpenVMS" when you log in, is started directly from dtlogin, not from the Xsession script. Command line arguments can not be passed to dthello.

- In the New Desktop, the login log file is optionally enabled using the SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM file (see Section 3.5). It does not use the Dtlogin.errorLogFile resource in the /usr/dt/config resource file.

- The session startup log file is SYS$LOGIN:DECW$SM.LOG in the New Desktop, not $HOME/.dt/startlog.

The New Desktop login process does use an XSESSION.COM file to start the dtsession process, and the New Desktop does support an CDE$SYSTEM_DEFAULTS:[BIN.XSESSION_D] directory for processing specialized startup command files. In addition, the New Desktop supports the standard set of resource files, including their localized versions.

A.3 File Manager Differences

The OpenVMS implementation of File Manager differs from the UNIX implementation in the following areas:

- File versions

  File versions do not exist on UNIX systems. On OpenVMS Alpha systems, you can choose to see all versions of your files or only the highest version. You can purge files by first selecting them and then choosing Purge from the Selected menu. Regardless of which version of a file name you selected, all versions of the file will be purged.

- Protection dialog boxes

  Additional protection codes are available on OpenVMS systems.

- Updating views of directories

  On UNIX systems, directories that are modified are automatically updated. On OpenVMS Alpha systems, a view of a directory is automatically updated if the directory is revised by a File Manager action, such as moving, deleting, or copying a file. To see whether directory changes were made by another process, you can select the Update option from the View menu.

- On OpenVMS Alpha systems, there is nothing equivalent to a root directory.

  You cannot traverse from one disk or rooted logical to another.

- Minimal use of extra processes

  UNIX implementations of File Manager make extensive use of additional (forked) processes. The OpenVMS implementation uses additional processes only for certain file system operations. Because of this, certain operations, such as directory updates, require the user to wait for completion rather than perform another action.

A.4 Printing Differences

The New Desktop printing implementation differs from the CDE printing implementation. The current implementation of CDE printing relies heavily on the UNIX line printer command, lp(1), and the line printer daemon, lpd(8), for its underlying functionality. Because the printing environment for CDE has not been standardized, it was not practical to port this type of operating system specific implementation to OpenVMS Alpha.

On OpenVMS Alpha, the New Desktop printing environment consists of a new application called Print Dialog. It uses the User Interface Language (UIL) and the DECwindows Print Widget (part of the DECWindows Extensions to Motif library) to provide access to printing on OpenVMS Alpha.

Although the New Desktop implementation differs, an attempt was made to integrate OpenVMS Alpha printing with the CDE printing paradigm so that printing would "feel" similar to printing on UNIX systems running CDE. Printers are still managed as objects on the desktop, with print jobs initiated either by dragging file icons and dropping them on printer icons or by selecting them and using the Print Dialog dialog box.

The Print Dialog executable file is located with the other New Desktop executables in the following directory:

```
CDE$SYSTEM_DEFAULTS:[BIN]PRINTDIALOG.EXE
```

A.5 Messaging Differences

Messaging capabilities of the New Desktop are more limited than those of CDE. Some activities that appear synchronous with CDE on UNIX are not synchronous in the New Desktop or may require user intervention. This limitation is noticeable in the following cases:

- The timing of the busy light on the Front Panel and the hourglass cursor may be different with the New Desktop.

- As described in Section A.3, file updates may not occur automatically.

- When you invoke the Icon Editor from the Create Action application and edit an icon file, the edits that you make are not automatically shown in the displayed icon when you exit from Icon Editor, as they are with CDE on UNIX (see Section 4.1.6).

- On the New Desktop, the Text Editor dtpad runs only in standalone mode.

A.6 Differences in Invoking Processes

Most processes invoked within the New Desktop are started using actions with associated execution strings (EXEC_STRINGs). With the New Desktop, these EXEC_STRINGs must be a single valid DCL command. Strings that begin with file names are automatically converted to foreign command lines.

A.7 Directory Hierarchies of the New Desktop and CDE

This section describes the differences between the directory hierarchies of the New Desktop and those of CDE.

CDE was developed for UNIX systems. The CDE documentation, which is provided on line with the New Desktop, uses UNIX path specifications.

A.7.1 System Default Configuration Directories

Table A–1 shows the CDE path specifications that correspond to the New Desktop system default configuration directory names. For more information about the CDE file system or directory tree structure, see the dtfilsys.5 reference page, which you can access with the Man Page Viewer.

The Man Page Viewer is located in the Desktop Apps group in Application Manager. To view dtfilsys.5, access the Man Page Viewer and enter dtfilsys.5 at the Man Page prompt.

Table A–1 System Default Configuration Directories

OpenVMS Directory Name	UNIX Path Specification
CDE$SYSTEM_DEFAULTS:[000000]	/usr/dt
CDE$SYSTEM_DEFAULTS:[APP-DEFAULTS.*lang*[1]][2]	/usr/dt/app-defaults/<lang>
CDE$SYSTEM_DEFAULTS:[APPCONFIG]	/usr/dt/appconfig
CDE$SYSTEM_DEFAULTS: - [APPCONFIG.APPMANAGER.*lang*[1]]	/usr/dt/appconfig/appmanager
CDE$SYSTEM_DEFAULTS:[APPCONFIG.HELP.*lang*[1]]	/usr/dt/appconfig/help
CDE$SYSTEM_DEFAULTS:[APPCONFIG.ICONS.*lang*[1]][3]	/usr/dt/appconfig/icons
CDE$SYSTEM_DEFAULTS:[APPCONFIG.TYPES.*lang*[1]]	/usr/dt/appconfig/types

[1]For the value of *lang* (or %L) in a directory or file name on OpenVMS, any dot (.) or at sign (@) in the locale name is converted to an underscore (_). For example, fr_FR.ISO8859-1 on UNIX is FR-FR_ISO8859-1 on OpenVMS.

[2]Application default files on UNIX have no file extension. On OpenVMS there is a .DAT file extension on each application default file in the [APP-DEFAULTS] directory. For example, /usr/dt/app-defaults/C/Dtpad is CDE$SYSTEM_DEFAULTS:[APP-DEFAULTS.C]DTPAD.DAT in the New Desktop. (This does not apply to resource files located in the configuration directory hierarchy.)

[3]Icon pixmap and bitmap files that contain multiple dots in their file names have been converted to an OpenVMS file name syntax. The second dot in the file name becomes an underscore in the New Desktop. For example, sysinfo.l.pm becomes SYSINFO.L_PM.

(continued on next page)

Table A–1 (Cont.)   System Default Configuration Directories

OpenVMS Directory Name	UNIX Path Specification
CDE$SYSTEM_DEFAULTS:[BACKDROPS]	/usr/dt/backdrops[4]
CDE$SYSTEM_DEFAULTS:[BIN]	/usr/dt/bin
CDE$SYSTEM_DEFAULTS:[CONFIG]	/usr/dt/config
CDE$SYSTEM_DEFAULTS:[CONFIG.*lang*[1]]	/usr/dt/config/<lang>
—	/usr/dt/dthelp[5]
CDE$SYSTEM_DEFAULTS:[EXAMPLES]	/usr/dt/examples[4]
—	/usr/dt/include[5]
DECW$INCLUDE:, DT:	/usr/dt/include/Dt[4]
SYS$MANAGER:CDE$STARTUP.COM	/usr/dt/install/dec/start.cde.dec
CDE$SYSTEM_DEFAULTS:[LIB]	/usr/dt/lib
CDE$SYSTEM_DEFAULTS:[MAN]	/usr/dt/man[4]
CDE$SYSTEM_DEFAULTS:[PALETTES]	/usr/dt/palettes[4]
—	/usr/dt/share[5]

[1]For the value of *lang* (or %L) in a directory or file name on OpenVMS, any dot (.) or at sign (@) in the locale name is converted to an underscore (_). For example, fr_FR.ISO8859-1 on UNIX is FR-FR_ISO8859-1 on OpenVMS.

[4]On a UNIX system, the file path specifications for the backdrops, examples, include, man, and palettes directories are links to a share directory. On an OpenVMS Alpha system, the directories for these same files contain the actual files.

[5]There is no corresponding directory in the New Desktop.

A.7.2 Node and Cluster Configuration Directories

Table A–2 shows the directories that a system manager creates for any resource, application default, or configuration files that have been customized for systemwide use. The corresponding UNIX path specifications are also shown.

Table A–2 Node and Cluster Configuration Directories

OpenVMS Directory Name	UNIX Path Specification
CDE$USER_DEFAULTS:[000000]	/etc/dt
CDE$USER_DEFAULTS:[APPCONFIG]	/etc/dt/appconfig
CDE$USER_DEFAULTS: -[APPCONFIG.APPMANAGER.*lang*]	/etc/dt/appconfig/appmanager/<lang>
CDE$USER_DEFAULTS:[APPCONFIG.HELP.*lang*]	/etc/dt/appconfig/help/<lang>
CDE$USER_DEFAULTS:[APPCONFIG.ICONS.*lang*]	/etc/dt/appconfig/icons/<lang>
CDE$USER_DEFAULTS:[APPCONFIG.TYPES.*lang*]	/etc/dt/appconfig/types/<lang>
CDE$USER_DEFAULTS:[BACKDROPS]	/etc/dt/backdrops
CDE$USER_DEFAULTS:[CONFIG]	/etc/dt/config
CDE$USER_DEFAULTS:[CONFIG.*lang*]	/etc/dt/config/<lang>
CDE$USER_DEFAULTS:[PALETTES]	/etc/dt/palettes

A.7.3 User Configuration Directories

Table A–3 shows the directories that are used for individual users' customizations.

Table A–3 User Configuration Directories

OpenVMS Directory Name	UNIX Path Specification
disk$:[*user*.DT]	$HOME/.dt
disk$:[*user*.DT.APPMANAGER]	$HOME/.dt/appmanager
disk$:[*user*.DT.HELP]	$HOME/.dt/help
disk$:[*user*.DT.ICONS]	$HOME/.dt/icons
disk$:[*user*.DT.PALETTES]	$HOME/.dt/palettes
disk$:[*user*.DT.SESSIONS]	$HOME/.dt/sessions
disk$:[*user*.DT.TMP]	$HOME/.dt/tmp
disk$:[*user*.DT.TYPES]	$HOME/.dt/types

A.7.4 Font Directories

The font directories of the New Desktop and the corresponding UNIX path specifications are shown in Table A–4.

Table A–4   Font Directories

OpenVMS Directory Name	UNIX Path Specification
CDE$SYSTEM_DEFAULTS:[CONFIG.XFONTS.C.100DPI]	/usr/dt/config/xfonts/C/100dpi
CDE$SYSTEM_DEFAULTS:[CONFIG.XFONTS.C.75DPI]	/usr/dt/config/xfonts/C/75dpi

B

Multiple-Screen Support

This appendix describes the New Desktop support for multiple screens and
what you need to do to enable this support.

B.1 Overview of Multiple Screens With a Single Display

The New Desktop supports a single X display with multiple screens, as shown
in Figure B–1. This configuration is sometimes called a dual-headed display.

Figure B–1 Single X Display With Multiple Screens
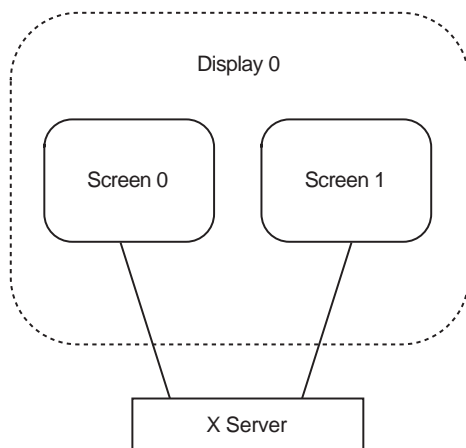
ZK-8679A-AI

With this configuration, each screen served by the X server is a desktop. The
Front Panel appears on only one desktop (usually screen 0 of display 0). You
can move the pointer from screen to screen by moving it beyond the left and
right sides of the desktop. You can place icons on a screen's desktop only if File

Manager or Application Manager (dtfile) has been started on that screen. Icons cannot be dragged from one screen and dropped on another.

You can add certain functions to the Window Manager configuration file that make configuring and manipulating multiple screens easier and quicker. These functions (described in Section B.2) enable you to create and change workspaces on any screen.

In a multiple-screen configuration, one screen is the default or current screen. You establish the default screen with the Set Default Screen application in Application Manager's Desktop Tools group. All applications that are subsequently started from either the Front Panel or File Manager will be started on the default screen.

Note

File Manager, Application Manager, and Trash Can are actually windows from the same dtfile application. Only one dtfile application will be active in your session. The first invocation of File Manager, Application Manager, or Trash Can defines the screen used for all future invocations of these tools. They will always be on the same screen.

B.2 Window Manager Functions for Multiple Screens

The Window Manager functions that are required for using multiple screens are described in Table B–1.

Table B–1   Window Manager Functions for Multiple Screens

Function	Description
f.screen [next | prev | back | *screen_number*]	Causes the pointer to move to a specific screen number or to the next, previous, or last-visited (back) screen. The arguments to this function are mutually exclusive. The *screen_number* argument indicates the screen to which the pointer is to go. Screen numbers start with screen 0. Next or prev causes the pointer to move to the next or previous managed screen (skipping over any unmanaged screens). Specifying back causes the pointer to move to the last-visited screen.
f.create_workspace	Creates a new workspace on the screen. The new workspace name is generated automatically and takes the form ws_*n*, where *n* is an integer.
f.delete_workspace	Deletes the current workspace. Windows that reside only in this workspace are moved to the next workspace. If the last workspace is being deleted, then windows are moved to the first workspace.
f.goto_workspace *workspace-name*	Causes Workspace Manager to switch to the named workspace. If no workspace exists by the specified name, then no action occurs. Note that adding and deleting workspaces dynamically affects this function.
f.next_workspace	Causes the workspace manager to switch to the next workspace on the screen. If the last workspace is currently active, then this function switches to the first workspace.
f.prev_workspace	Causes the Workspace Manager to switch to the previous workspace on the screen. If the first workspace is currently active, then this function switches to the last workspace.
f.action	Causes the specified action to be invoked on the screen.

B.3 Configuring Window Manager for Managing Multiple Screens

You can edit the Window Manager configuration file for managing multiple screens for systemwide use or for use by a single user. To set up a customized workspace menu (also known as a root window menu) for all users on a multiple-screen system, copy the system's default Window Manager

configuration file to the user's default area and modify the file there, as
shown in the following example:

```
$ COPY CDE$SYSTEM_DEFAULTS:[CONFIG.C]SYS.DTWMRC  -
_$ CDE$USER_DEFAULTS:[CONFIG.C]SYS.DTWMRC
$ EDIT CDE$USER_DEFAULTS:[CONFIG.C]SYS.DTWMRC
```

To set up a customized workspace menu for one user, copy the user's default
Window Manager configuration file to *disk*$:[*user*.DT]DTWMRC.DAT and
modify the file there, as shown in the following example:

```
$ COPY CDE$USER_DEFAULTS:[CONFIG.C]SYS.DTWMRC -
_$ disk$:[user.DT]DTWMRC.DAT
$ EDIT disk$:[user.DT]DTWMRC.DAT
```

By default, the workspace menu is defined in SYS.DTWMRC, as shown in the
following source code:

```
###
# Root Menu Description
###

Menu DtRootMenu
{
    "Workspace Menu"                f.title
    "Shuffle Up"                    f.circle_up
    "Shuffle Down"                  f.circle_down
    "Refresh"                       f.refresh
    "Minimize/Restore Front Panel"  f.toggle_frontpanel
    no-label                        f.separator
    "Restart Workspace Manager..."  f.restart
    no-label                        f.separator
    "Log out..."                    f.action ExitSession
}
```

The following example shows what the workspace menu definition might look
like after it has been customized for use in a multiple-screen configuration:

```
###
# Root Menu Description
###

Menu DtRootMenu
{
    "Workspace Menu"                f.title
    "Shuffle Up"                    f.circle_up
    "Shuffle Down"                  f.circle_down
    "Refresh"                       f.refresh
    "Minimize/Restore Front Panel"  f.toggle_frontpanel
    no-label                        f.separator
    "Next Screen"                   f.screen next
    "Previous Screen"               f.screen prev
    "Create Workspace"              f.create_workspace
    "Next Workspace"                f.next_workspace
    "Previous Workspace"            f.prev_workspace
    "Delete Workspace"              f.delete_workspace
    no-label                        f.separator
    "Style Manager"                 f.action dtstyle
    "File Manager"                  f.action DtfileHome
    no-label                        f.separator
    "Restart Workspace Manager..."  f.restart
    no-label                        f.separator
    "Log out..."                    f.action ExitSession
}
```

B.4 Using Style Manager on Multiple Screens

Style Manager provides controls for customizing several aspects of your
desktop. When using Style Manager on multiple screens, two controls—color
and backdrop—must be set for each workspace on each screen; the remaining
controls apply to all workspaces on all screens.

Only one instance of Style Manager can be running at any time. Changes will
apply to the workspace and screen on which Style Manager is running.

Initially, all screens except the default screen display a gray patterned
background with the default X server colormap installed. You can then choose
from any of the color palettes in Style Manager's Color dialog box to set
the colors for the initial workspace of the screen. If you create additional
workspaces on the screen, you must select a color palette from the Style
Manager for each one. Backdrops function similarly.

B.5 Customizing Applications for a Specific Screen

In addition to the screen-specific attributes that you can customize with Style Manager, you can customize applications on a screen-specific basis by using command line options and screen-specific resources.

B.5.1 Using Command Line Options

The Window Manager supports the following command line options for managing multiple screens:

- -multiscreen

  Multiple-screen management is enabled by default. You can use this option to disable it.

- -screen [*screen-name* ...]

  This option enables users to specify a list of screen names to be used in place of the default screen numbers. You can then specify the screen names, separated by spaces, when setting screen-specific resources. By default, the screen names are numbers, with zero (0) representing the default screen.

The following example names the second screen "screen1" and displays the icon box on it, using the name in a screen-specific resource.

```
$ MCR CDE$PATH:DTWM -screens "screen0" "screen1" -xrm -
_$ "*screen1*useIconBox: true"
```

These options are not required for managing multiple screens with Window Manager.

All the CDE applications included in the New Desktop support the following command line options:

- -display [node:display*x*[.screen*x*]]

  This option specifies the screen on which to display the application. This is done by specifying the number of the screen in the option value. By default, screen numbers start at zero (0).

- -xrm "*resource_id:resource_value*"

  This option, the X Resource Manager option, allows the user to set screen specific customizations for the application from the command line. This is done using the screen number (or name) in the resource specification.

B.5.2 Specifying Resources for a Specific Screen or Workspace

Screen-specific resources are Window Manager resources that apply only to
the clients that appear on a particular screen. Workspace-specific resources
are Window Manager resources that apply only to the clients that appear on a
particular workspace.

The syntax for screen-specific resources is:

*screenname*resource_id: resource_value

By default, the screen number is used as the screen name.

In the following example, Style Manager specifies the color palette resource for
all clients on screen 0.

*0*ColorPalette: palette_name

The syntax for workspace-specific resources is:

Dtwm[*screen_name]*workspace_name*resource_id: resource_value

The following example specifies the workspace title for workspace wk1. This
title is the value that appears in the workspace button representing workspace
wk1 on the Front Panel.

Dtwm*0*ws1*title: MYWORKSPACE

B.5.3 Specifying Applications for a Specific Screen

You can specify an application for a specific screen by editing its action
definition file to include a specific screen number. The following example
is a screen-specific action definition for the X Window Dump (xwd)
application. When this action is invoked, the application prompts the
user for a screen number and dumps only that screen's image to the file
DISK$WORK:SCREENDUMP.XWD. The -display command line option in this
example causes the prompt to be displayed.

```
ACTION screendump
{
    LABEL       Screen Dump
    ICON        Dtactn
    TYPE        COMMAND
    ARG_COUNT   0
    EXEC_STRING decw$utils:xwd \
                -display :0.%"Screen Number: "% \
                -root -screen -out DISK$WORK:SCREENDUMP.XWD
    WINDOW_TYPE NO_STDIO
    DESCRIPTION Dump the image on the Screen to DISK$WORK:SCREENDUMP.XWD
}
```

C

CDE API Routines

This appendix lists all the CDE API routines. They are grouped by shareable libraries (pointed to by the logical SYS$LIBRARY) and by the header files within those libraries. The routines are defined in their respective header files. Each library of routines is shown in a separate table.

A few CDE API routines are not included in the New Desktop. They are indicated in each table by a footnote.

Detailed information describing the parameters and use of these routines is available from their respective manpages. You can either select the Man Page Viewer from the Desktop Apps application group or run it from the command line, as shown in the following example:

```
$ DTHELPVIEW == "$CDE$PATH:DTHELPVIEW"
$ DTHELPVIEW "-manPage" DtInitialize
```

C.1 CDE Desktop Services API Routines

The CDE Desktop Services API routines in CDE$LIBDTSVC.EXE are shown in Table C–1.

Table C–1   CDE Desktop Services API Routines in CDE$LIBDTSVC.EXE

API	Header File	Routines
API initialization	Dt.h	DtInitialize
		DtAppInitialize
Action processing	Action.h	DtActionExists
		DtActionLabel
		DtActionDescription
		DtDbReloadNotify[1]
		DtDbLoad
		DtActionInvoke
		DtActionIcon
Data typing	Dts.h	DtDtsLoadDataTypes
		DtDtsRelease
		DtDtsDataToDataType
		DtDtsFileToDataType
		DtDtsFileToAttributeValue
		DtDtsFileToAttributeList
		DtDtsBufferToDataType[1]
		DtDtsBufferToAttributeValue[1]
		DtDtsBufferToAttributeList[1]
		DtDtsDataTypeToAttributeValue
		DtDtsDataTypeToAttributeList
		DtDtsFreeDataTypeNames
		DtDtsFreeAttributeList
		DtDtsFreeAttributeValue
		DtDtsFreeDataType
		DtDtsDataTypeNames
		DtDtsFindAttribute
		DtDtsSetDataType
		DtDtsDataTypeIsAction

[1]Not included in the New Desktop.

Table C–1 (Cont.)   CDE Desktop Services API Routines in CDE$LIBDTSVC.EXE

API	Header File	Routines
Drag-and-drop convenience	Dnd.h	DtDndCreateSourceIcon
		DtDndDragStart
		DtDndVaDragStart
		DtDndDropRegister
		DtDndVaDropRegister
		DtDndDropUnregister
Workspace Manager	Wsm.h	DtWsmAddWorkspaceFunctions
		DtWsmRemoveWorkspaceFunctions
		DtWsmGetWorkspaceInfo
		DtWsmFreeWorkspaceInfo
		DtWsmGetWorkspaceList
		DtWsmGetCurrentWorkspace
		DtWsmAddCurrentWorkspaceCallback[1]
		DtWsmRemoveWorkspaceCallback[1]
		DtWsmSetCurrentWorkspace[1]
		DtWsmGetWorkspacesOccupied
		DtWsmSetWorkspacesOccupied
		DtWsmOccupyAllWorkspaces
		DtWsmAddWorkspaceModifiedCallback[1]
		DtWsmGetCurrentBackdropWindow
Session save/restore support	Session.h	DtSessionSavePath
		DtSessionRestorePath
Screen saver	Saver.h	DtSaverGetWindows

[1]Not included in the New Desktop.

C.2 CDE Help Widget Support Routines

The CDE Help Widget support routines in CDE$LIBDTHELP are shown in
Table C–2.

Table C–2   CDE Help Widget Routines in CDE$LIBDTHELP.EXE

API	Header File	Routines
Help utility routines	Help.h	DtHelpSetCatalogName[1]
		DtHelpReturnSelectedWidgetId
General help dialog support	HelpDialog.h	DtCreateHelpDialog
Quick help dialog support	HelpQuickD.h	DtCreateHelpQuickDialog
		DtHelpQuickDialogGetChild

[1]Not included in the New Desktop.

C.3 CDE Custom Widget Support Routines

The CDE custom widget support routines in CDE$LIBDTWIDGET are shown
in Table C–3.

Table C–3   CDE Custom Widget Routines in CDE$LIBDTWIDGET.EXE

API	Header File	Routines
DtSpinBox widget support	SpinBox.h	DtCreateSpinBox
		DtSpinBoxAddItem
		DtSpinBoxDeletePos
		DtSpinBoxSetItem
DtComboBox widget support	ComboBox.h	DtCreateComboBox
		DtComboBoxAddItem
		DtComboBoxDeletePos
		DtComboBoxSetItem
		DtComboBoxSelectItem
DtMenuButton widget support	MenuButton.h	DtCreateMenuButton

Table C–3 (Cont.)   CDE Custom Widget Routines in CDE$LIBDTWIDGET.EXE

API	Header File	Routines
DtEditor Text Editor widget support	Editor.h	DtCreateEditor
		DtEditorAppend
		DtEditorAppendFromFile
		DtEditorChange
		DtEditorCheckForUnsavedChanges
		DtEditorClearSelection
		DtEditorCopyToClipboard
		DtEditorCutToClipboard
		DtEditorClearSelection
		DtEditorCopyToClipboard
		DtEditorCutToClipboard
		DtEditorDeleteSelection
		DtEditorDeselect
		DtEditorDisableRedisplay
		DtEditorEnableRedisplay
		DtEditorFind
		DtEditorFormat
		DtEditorGetContents
		DtEditorGetInsertionPosition
		DtEditorGetLastPosition
		DtEditorGetMessageTextFieldID
		DtEditorGetSizeHints
		DtEditorGoToLine
		DtEditorInsert
		DtEditorInsertFromFile
		DtEditorInvokeFindChangeDialog
		DtEditorInvokeFormatDialog
		DtEditorInvokeSpellDialog[1]

[1]Not included in the New Desktop.

Table C–3 (Cont.)   CDE Custom Widget Routines in CDE$LIBDTWIDGET.EXE

API	Header File	Routines
		DtEditorPasteFromClipboard
		DtEditorReplace
		DtEditorReplaceFromFile
		DtEditorReset
		DtEditorSaveContentsToFile
		DtEditorSelectAll
		DtEditorSetContents
		DtEditorSetContentsFromFile
		DtEditorSetInsertionPosition
		DtEditorTraverseToEditor
		DtEditorUndoEdit

Index

C

Calculator, 1–6, 2–6
Calendar, 1–6, 2–6, 2–10
Calendar Manager, 5–4
Cardfiler, 2–6
Catalogs
   message, 3–4, 4–11
CDE, 1–1
   differences between OpenVMS and UNIX
      implementations, A–1
CDE$DETACHED_LOGICALS symbol,
   3–25
CDE$LIBDTHELP.EXE shareable library,
   5–2
CDE$LIBDTSVC.EXE shareable library,
   5–2
CDE$LIBDTWIDGET.EXE shareable library,
   5–2
CDE$LOG_PROCESSES symbol, 3–26
CDE$PATH logical name, 4–3
CDE$SPAWN_PROCESSES symbol, 3–26
CDE$STARTUP.COM command procedure,
   3–1
CDE$SYSTEM_DEFAULTS logical name,
   3–2, 3–3
CDE$USER_COMMON logical name, 3–5
CDE$USER_DEFAULTS logical name, 3–5
   directory hierarchy, 4–17
CDE$USER_SPECIFIC logical name, 3–5
Clock, 1–6
   icon, 2–3
Colormap
   X server's, 3–17
Colors
   See also Palettes
   avoiding conflicts, 3–20
   changing number, 3–19
   dynamic, 3–17
   New Desktop usage, 3–17
Color sets
   resources, 3–17

Command procedures
   CDE$STARTUP.COM, 3–1
   DECW$PRIVATE_APPS_SETUP.COM,
      3–2, 3–16, 3–25
   DECW$STARTUP.COM, 3–1
   DTHELPTAG.COM, 5–7
   XSESSION.COM, A–2
Common Desktop Environment
   See CDE
Compilers
   help, 5–6
   UIL, 5–1
Create Action application, 1–6, 4–1, 4–3
   overview, 4–4
   specifying arguments, 4–4
Customizations
   applications, 3–7
   Automatic Startup
      New Desktop equivalent, 3–15
   default editor, 3–13
   environment, 3–7
   keyboard settings, 3–9
   printers, 3–9
      private, 3–11
      systemwide, 3–10
   screen background, 3–13
   security settings, 3–9
   systemwide, 3–5
   user configuration, 3–6, A–7
   window colors, 3–13
   Window Manager, 3–13

D

Data files
   order of precedence, 3–6
Data types, 3–8
   adding, 4–6
   creating, 4–7
   definition files, 4–2
   family, 4–9
   modifying, 4–7
Data typing, 3–8
   files, 4–10
   folders, 4–10