

# A Common Desktop Environment for Platforms Based on the UNIX<sup>®</sup> Operating System

User interface technologies from four companies have been combined to create a single UNIX desktop standard that provides a common look and feel for end users and a common set of tools for system administrators and application developers.

by **Brian E. Cripe, Jon A. Brewster, and Dana E. Laursen**

Until the early 1980s most users interacted with their computers via character-mode interfaces—they typed in commands. What happened to change all this was the arrival of proprietary window systems. HP's first window system was embedded in the Integral Personal Computer.<sup>1</sup> By 1988 the X Window System had been adopted as a standard for machines running the UNIX operating system. However, available user interface toolkits, such as HP's CXI widgets, were proprietary. These toolkits provided items such as scroll bars and pop-up menus, and they allowed software developers to create applications that had a consistent look and feel. By 1990 two stable toolkits had emerged, OSF/Motif and OpenLook.<sup>†</sup>

The stage was now set for proprietary user environments. A user environment is a collection of programs used by the end user to manage files, invoke applications, and perform routine tasks such as edit text files and send and receive email. HP delivered its first version of HP VUE (Visual User Environment)<sup>2</sup> in 1990 with subsequent upgrades continuing to this day.

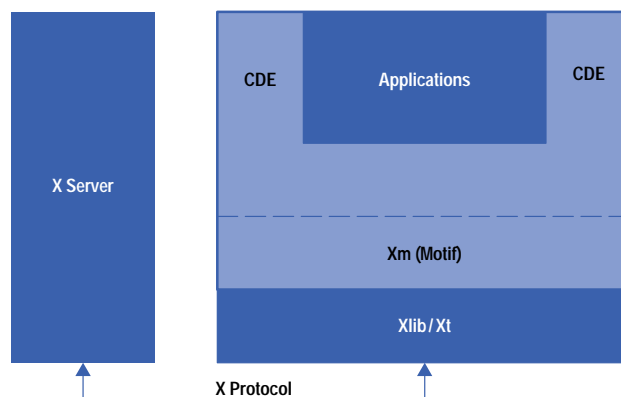
In March of 1993 representatives from Hewlett-Packard, IBM, Sun Microsystems, and Novell agreed to create a common user environment for UNIX platforms (see [Article 6](#)). This joint initiative resulted in the specification and development of the Common Desktop Environment (CDE). CDE accomplishes two things: first, it adopts OSF/Motif as the principal user interface toolkit for UNIX systems, and second, it establishes this rich new environment and framework as a standard user environment.

CDE is based on the X Window System from the X Consortium and the Motif graphical user interface from the Open Software Foundation. Fig. 1 shows how these technologies fit together.

The X Window System (X) components include:

- X server. This program writes directly to the user's display hardware.
- Xlib. This is a library of function calls for communicating with the X server. Xlib deals with low-level concepts such as rectangles, arcs, and fonts. It does not know about higher-level concepts such as menus and scroll bars (i.e., interface widgets).

**Fig. 1. CDE architecture.**



<sup>†</sup> OpenLook is the X Window System toolkit from Sun Microsystems.

- X protocol. This is the data stream that communicates between Xlib and the X server. This data stream can be passed across a network, which gives X the ability to run an application on one system and display the user interface to another system.
- Xt. This is the X toolkit, which provides a framework for defining and integrating user interface widgets.

The Motif component is Xm, which is the Motif library that provides a rich collection of user interface widgets such as dialog boxes, menus, buttons, scroll bars, and text editing panes.

## Different Views of CDE

The rest of this article provides an overview of CDE from three different perspectives: the end user who uses CDE but does not care about its internal design, the software developer who is writing applications that need to be integrated with CDE, and the system administrator who is responsible for managing systems that run CDE.

### End-User's View

Putting together a user environment and application framework such as CDE always forces one to be precise about requirements. Many of the driving inputs for the CDE project turned out to be based on the following end-user requirements:

- Completeness. CDE needs to be a full-service environment covering everything from login to logout. This includes security and authentication, windowing, application launching, file management, email, and so on.
- Integration. CDE parts and services need to work together seamlessly. For example, it should be possible to mail a meeting notice with a calendar appointment in it, allowing the recipients to add the event easily to their calendars. Also, CDE utilities need to use CDE APIs (e.g., help, drag and drop, etc.) not only to be consistent with each other, but to be showcase components showing the results of proper use.
- Compatibility. Previous applications need to continue to run. This is true for OpenLook, Motif, and terminal-based software. For HP VUE users, we were very interested in ensuring that conversion tools could be created to move configuration information into CDE.
- Ease of use. The resulting environment needs to be guided by a standard set of usability principles and tested for usability defects. This work took place at the early stages and during every step of the CDE project.

Getting agreement on these fundamental end-user requirements was critical given the nature of our extended multicompany development team. Many of the more difficult project decisions required coming back to these basics. For example, the drag and drop architecture had to be reworked several times to accomplish the integration ambitions of the team.

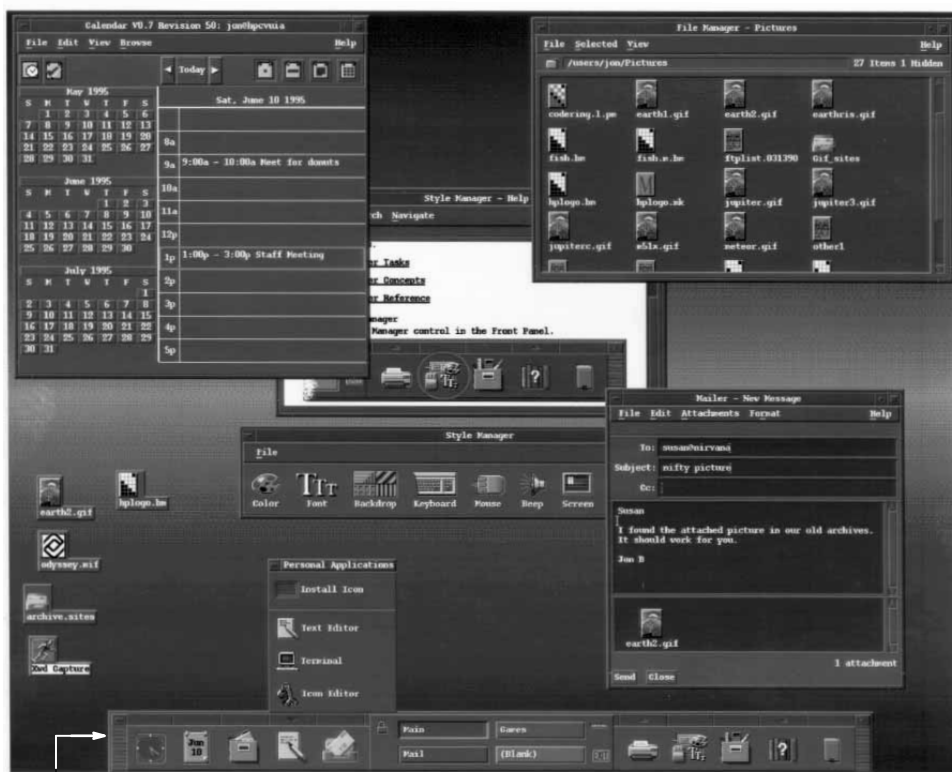
The cover of this issue and Fig. 2 show a typical CDE user interface, and Table I lists all the end-user components, some of which are shown in Fig. 1.

### Basic End-User Tasks

The first thing a user does when approaching CDE is log in. It is the CDE login service that authenticates and authorizes user access to the system. This gatekeeper for the UNIX system then invokes the basic session components such as the window manager and file manager. The user's previous session can also be automatically restored. This allows items such as running applications, color settings, and the graphical desktop arrangement to be retained from the logout of the previous session.

Once the user is logged in, the file manager is used to find and organize data files. Files can be dragged out of the file manager and dropped in many interesting places such as the printer. Files can be moved and copied by dragging them about. File icons can be placed on the main screen as if the screen were a desktop. A file's type (document, spreadsheet, image, etc.) can be determined by its icon, and various type-specific actions can be invoked via a pop-up menu that is made available for each icon. For example, editing a graphics image and viewing that same image might require two different applications and thus two different actions in the pop-up actions menu. The file manager also allows different views of the file directories, such as a tree view (containment hierarchy) and a full file properties view (size, security, etc.). Files can be found by manually browsing through the directories or via a finding service that works with name or content matching. A found file can then be automatically placed on the desktop.

**Fig. 2.** A typical CDE user interface.



Front Panel

**Table I**  
End-User Components

Component	Function
Login Manager	Graphical login and authentication
Session Manager	Logout to login session save and restore
Window Manager	X11 Windows compliant window manager
Workspace Manager	Window grouping and task switching
Front Panel	Ubiquitous services access and workspace switching
Style Manager	Customization and configuration
File Manager	File manipulation services
Application Manager	Application discovery and launch services
Text Editor	ASCII text file editor
Icon Editor	File type icon creator and editor
Mailer	Full MIME compliant multipart email facility
Calendar	Personal and group time manager
Calculator	Financial, logical, and scientific modes
Terminal Emulator	VT220 type character mode support
Action Creator	Linking behavior to specific icons
Print Manager	Multiprinter-aware print submission tool

Applications can be launched in a number of ways. Simply double-clicking a data file icon and allowing CDE to launch the correct application with that data file as its argument is usually the most convenient approach. However, sometimes running an application directly is more appropriate. The applications themselves also show up as icons that can be moved about, organized, and double-clicked for invocation. These application icons are initially found in the application manager, which is really just a file manager window onto the applications directory. A user can drag the application icons into any other directory or even onto the desktop. For important and commonly used applications, the icon can be directly installed into the front panel for easy access.

**Article 2** describes the data structures involved in linking applications, icons, and actions together.

Task management in CDE is much like any other windowing environment in that it has various ways to control the windowing behavior of the running applications. However, there is a major enhancement in CDE called *workspaces*. The multiple-workspace paradigm supported by CDE allows a user to switch between multiple screens full of application windows. This allows each collection of windows to be treated as a group (the default is four workspaces). End users can use workspaces to collect together groups of windows for specific tasks. For example, one workspace could contain all the windows associated with a remote system, while another workspace could contain windows for the user's desktop publishing tools.

Workspace switching is very fast, typically less than a few tenths of a second, and is accomplished by pressing a single button within the front panel. Each workspace has its own backdrop, which is usually color-coded to match its button. The workspaces are also named and can be created and destroyed at any time by the user.

The front panel is usually located at the bottom of the screen (see Fig. 2). It is available in all workspaces and provides access to important and frequently used services. These services include:

- Clock and calendar functions
- Personal file space (file manager access)
- General application space (application manager access)
- High-use application and data file access
- Mail facilities
- Workspace management controls
- Printer access
- CDE configuration controls
- Help facilities
- Trash can for deleting files
- Screen locking and logout services.

The front panel is a unifying tool for collecting important services that are otherwise left scattered and hard to find. It is also a distinctive visual feature known to the development team as a *signature visual* in that it distinguishes the machine running CDE from other machines.

## End-User CDE Utilities

CDE has a number of utility programs that support specific end-user tasks. The following are some of these utilities.

**Text Editor.** This is a simple ASCII text editor that is neatly integrated into CDE so that drag and drop, help, and general Motif text behavior are seamlessly available to the end user. Text formatting, spell checking, and cut and paste services are also available.

**Mailer.** This is a highly integrated set of services that has full drag and drop behavior with respect to folders, messages, and attachments. The following scenario illustrates the services integrated with the mailer.

The front panel's mail icon is pressed, bringing up the user's in-box. A message is dragged from the in-box to the front panel's printer icon. Another message is dragged to a mail folder icon within a file manager for saving. A file is dragged from the file manager to the mail icon on the front panel for sending. Finally, a message being read has a graphics attachment that is double-clicked to invoke the graphics viewer.

Thus, from the CDE mailer the user can print a message, save a message to a file, view the graphics in a message, and compose and send a message.

**Calendar.** The calendar facility is a personal tool for managing time and to-do items. It is "group aware" so that the user can examine and search for meeting opportunities with colleagues. Individuals can control the privacy of their own schedules. Meetings can be emailed via the drag and drop mechanism, and the calendar view can be flipped between day, week, month, and six-month views.

**Terminal.** This tool supports character-mode applications (some of which predate window systems). The CDE terminal emulator behaves like a DEC VT220 terminal with minor changes consistent with ANSI and ISO standards. Full cut and paste behavior with the rest of the desktop is built in. The core feature of this emulator traces its ancestry back to HP's

Integral Personal Computer, which had HP's first windowing system and thus HP's first terminal emulator for the UNIX operating system.

## Software Developer's View of CDE

To the software developer, CDE is composed of two distinct components: the X/Open<sup>®</sup> standard and CDE product implementations. The X/Open standard defines the components that must be present on any system that claims to be CDE-compliant. HP's CDE product, like CDE products from other vendors, must support the interfaces defined by the X/Open CDE standard, but may contain additional functionality. For example, many vendors have enhanced CDE to provide backward compatibility with previous proprietary products. Software developers should be cautious when using features of a CDE product that are not part of the X/Open standard because they may not be portable to all CDE systems.

The major benefits that CDE provides to the developer are:

- A single GUI toolkit (OSF/Motif) that is supported by all major UNIX vendors
- Tools and libraries to help build an application
- Mechanisms to integrate an application with the desktop environment
- Mechanisms to integrate applications with each other.

Table II lists the components available in CDE that enable developers to integrate their applications into CDE. **Appendix A**, contains a complete list of all the CDE APIs, which enable developers to build applications based on CDE.

Table II Developer Components	
Component	Purpose
Help System	Viewer, API, and administration tools
Motif Toolkit	OSF/Motif version 1.2.3 plus some 1.2.4 repairs
Custom Widgets	SpinButton and ComboBox (taken from Motif 2.0)
Terminal Widget	For adding terminal emulation to an application
Dtksh	A GUI dialoging and scripting facility
Data Interchange	Drag and drop conventions, protocols, and APIs
ToolTalk <sup>®</sup>	A standard general-purpose message passing service that enables tight integration between separate applications and CDE components
Data Typing	API for access to the desktop engine typing services
Actions	API for access to the desktop invocation services
Font Guidelines	Conventions for standard font interactions
Internationalization Guidelines	Overview and reconciliation of relevant standards and style guides
Client/Server Guidelines	Network execution and distribution model

CDE defines three levels of application integration from which the developer can choose: basic, recommended, and optional. Basic integration consists of the minimal integration steps that allow a user to access an application from the desktop environment instead of a command prompt in a terminal window. Recommended integration requires more effort by the developer but allows the application to be fully consistent with CDE and other CDE applications. The final level, optional integration, is not necessary for most applications but is useful for applications that need to perform specialized tasks.

## Basic Integration

Basic integration allows an application and its data files to be managed by CDE. This management includes:

- Finding the application and invoking it using an icon in the application manager
- Identifying the application's data files with a unique icon
- Loading a data file into the application by dragging and dropping the file icon on the application icon
- Invoking the application to print a data file by dragging and dropping the file icon on a printer icon
- Using the style manager to specify colors and fonts for the application

- Locating information about the application in the help manager.

Basic integration can be accomplished without any modifications to the application's source or executable files. The steps for performing basic integration include:

- Defining an application group for the application manager that will hold the application
- Defining the application's icons and giving them the correct double click and drag and drop behavior by creating new CDE actions and data types
- Removing color and font specifications from the application's defaults file so that it will use the default colors and fonts specified by the CDE style manager
- Installing configuration files for the application in the standard file system locations and registering it with the desktop using the `dtappintegrate` command (This command is typically invoked by the application's installation script.)
- Creating and installing any appropriate application information as desktop help files.

## Recommended Integration

Recommended integration includes additional steps that are necessary to make the application fully integrated into CDE and consistent with other applications. This level of integration requires modifications to the application's source code, but in return it provides the following benefits:

- The user can access context-sensitive online help from within the application using the CDE help system. To achieve this the application must use the help system API.
- The application can achieve tight integration with other applications using ToolTalk messages. For example, an editor application that supports the media exchange message suite can be used by other applications for editing data objects. To achieve this the application must use the ToolTalk messaging API and support one or more of the standard message suites.
- The user can log out with the application running and upon login the application will be automatically restarted and restored to its previous state. To achieve this the application must be able to read and write session files that define the application's state, and it must use the session management API for controlling this behavior.
- The user can move data into or out of a running application using the drag and drop facility. To achieve this the application must use the drag and drop API.
- The application user interface can be translated into other languages without modifying the source code for the application. To achieve this the application must follow the internationalization guidelines.
- The application can be displayed on a remote workstation or an X terminal and be assured of getting the expected fonts. To achieve this the application must access its fonts using the standard font names defined in the font guidelines.

## Optional Integration

Applications with unique needs may choose to use the optional CDE integration facilities. Applications are not required or expected to use any of these facilities, but some applications will find them useful. These facilities include:

- Additional Motif widgets such as SpinBox and ComboBox
- Data typing functions that enable an application to determine the type and attributes of files and other data items in a manner consistent with CDE and other applications
- Action invocation functions that enable an application to invoke other applications
- Monitor and control functions for the placement of applications in the user's workspaces
- A terminal emulator widget that can be used to add a conventional UNIX command window to the application
- A text editor widget that allows adding a text editing window to the application, which is more powerful than the standard Motif text editor widget
- An API to access calendar and scheduling capabilities, which is an implementation of the X.400 association calendaring and scheduling API 1.0
- An enhanced version of Korn shell which provides access to CDE APIs from an interpreted script language.

More information about these integration techniques can be found in references [3](#) and [4](#).

## System Administrator's View of CDE

CDE greatly simplifies the burden of a UNIX system administrator because it provides a consistent set of capabilities and configuration mechanisms across virtually all UNIX systems. Tasks that an administrator of a CDE system might perform include configuring the behavior of CDE, administering CDE in a networked environment, and administering applications.

**Configuring CDE.** CDE is a highly configurable environment. Many of the customizations that a user can choose to do to configure a personal environment can also be done by a system administrator for all users. Some examples of possible configuration changes include the ability to:

- Customize the appearance of the login screen
- Modify the set of applications that get automatically started when a user first logs in
- Add or remove printer icons from the print manager
- Customize the contents of the front panel
- Lock all or portions of the front panel so that they cannot be modified by the user
- Customize the set of controls embedded in window frames and modify their behavior
- Modify the menus available from the root window of the display
- Modify the keyboard bindings and accelerator keys used by applications
- Customize the default fonts, colors, and backdrops used by CDE and applications.

For more information on any of these tasks see reference 4.

**Administering CDE in a Networked Environment.** CDE is designed to work well in a highly networked environment. The architecture of the desktop lets system administrators distribute computing resources throughout the network, including applications, data files for applications, desktop session services (desktop applications such as the login manager and file manager), and help services. Help data files can be put on a central help server.

Typical tasks performed by the administrator of a network running CDE include:

- Installing the operating system and CDE on a network of systems, some of which might perform specialized tasks such as act as an application server
- Configuring the login manager so that workstations or X terminals have login access to the appropriate set of systems
- Configuring the distributed file system so that all systems have access to the necessary set of data files
- Installing and configuring devices such as printers so that they are accessible from the desktop environment
- Configuring application servers that run applications on behalf of other systems in the network
- Configuring other servers such as database servers or help servers.

CDE includes a number of daemons. System administrators often do not need to be aware of these daemons because they are installed and configured automatically when CDE is installed. However, in some situations system administrators may need to use the information in the manuals and man pages to create some nontypical configurations.

These daemons include:

- dtlogin. The login manager, which provides login services to the workstation or X terminal
- dtspcd. The subprocess control daemon, which provides remote command invocation
- rpc.ttdbserver. The ToolTalk database server, which is used by the ToolTalk messaging system and performs filename mapping
- ttssession. The ToolTalk message server, which provides message passing
- rpc.cmsd. The calendar daemon, which manages the calendar databases.

More information about these daemons can be found in reference 3.

**Administering Applications.** The networking capabilities of the HP-UX\* operating system, the X Window System, and CDE can be used to create many different application execution configurations. The simplest configuration is local application execution in which applications are installed on the local disk of a workstation and executed locally.

A variation of this configuration is to install applications on a disk on a central file server and then mount that disk on other workstations. Each workstation accesses the application's executable and configuration files across the network, but executes them locally. This configuration reduces the total amount of required disk space because multiple workstations are sharing a single copy of the application files.

Another approach is to use centralized application servers. This configuration uses the client/server capabilities of the X Window System to execute the application on one system and display its user interface on another workstation or X terminal.

Application servers are a good solution to the problem of giving users access to applications that have special run-time requirements. For example, if users need access to an application that only runs on HP-UX 8.0, an HP-UX 8.0 application server can be created and accessed from workstations running HP-UX 9.0.

CDE makes these distributed application environments simple to use by representing all applications as icons in the application manager. The user does not need to know or care whether the application is installed locally or on a remote application server.

CDE also makes these distributed configurations easy to create and administer. Applications are installed the same way whether they will be used locally or accessed remotely. When a workstation is configured to access another system as an application server, all of the applications on that system that have been registered with CDE automatically become available. **Article 2** provides a more detailed discussion about CDE application administration tools.

## Summary

The HP VUE user will find much to appreciate in CDE. CDE retains the best end-user features of HP VUE, such as workspaces and the iconic desktop behavior. CDE adds many new end-user services, such as an integrated mailer and a calendar system. The system administrator gets a rich and new standard set of configuration options that also shares much of the HP VUE approach. A software developer has optional access to a new programming framework to take advantage of deep environment integration. Other than the help facility, these programming services were not available as part of HP VUE.

---

---

## References

1. *Hewlett-Packard Journal*, Vol. 36, no. 10, October 1985, pp. 4-35.
2. C. Fernandez, "A Graphical User Interface for a Multimedia Environment," *Hewlett-Packard Journal*, Vol. 45, no. 2, April 1994, pp. 20-22.
3. *CDE Programmer's Overview*, Hewlett-Packard, Part Number B1171-90105, January 1996.
4. *CDE Advanced User's and System Administrator's Guide*, Hewlett-Packard, Part Number B1171-90102, January 1996.

HP-UX 9.\* and 10.0 for HP 9000 Series 700 and 800 computers are X/Open Company UNIX 93 branded products.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

X/Open is a registered trademark and the X device is a trademark of X/Open Company Limited in the UK and other countries.

OSF, Motif, and Open Software Foundation are trademarks of the Open Software Foundation in the U.S.A. and other countries.

ToolTalk is a trademark or registered trademark of Sun Microsystems, Inc. in the U.S.A. and certain other countries.

- 
- 
- ▶ [Go to Article 2](#)
  - ▶ [Go to Table of Contents](#)
  - ▶ [Go to HP Journal Home Page](#)