HP OSF/Motif Style Guide

HP 9000 Series 300/800 Computers

HP Part Number 98794-90007

HEWLETT PACKARD

Hewlett-Packard Company
1000 NE Circle Blvd., Corvallis OR 97330

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MANUAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

Certification of conformance with the OSF/Motif user environment is pending.

Restricted Rights Legend

Use, duplication or disclosure by the Government is subject to restrictions as set forth in paragraph (b)(3)(B) of the Rights in Technical Data and Software clause in DAR 7-104.9(a).

Printing History

New editions of this manual will incorporate all material updated since the previous edition. Update packages may be issued between editions and contain replacement and additional pages to be merged into the manual by the user. Each updated page will be indicated by a revision date at the bottom of the page. A vertical bar in the margin indicates the changes on each page. Note that pages which are rearranged due to changes on a previous page are not considered revised.

The manual printing date and part number indicate its current edition. The printing date changes when a new edition is printed. (Minor corrections and updates which are incorporated at reprint do not cause the date to change.) The manual part number changes when extensive technical changes are incorporated.

September 1989........Edition 1

Contents

Preface P

This style guide contains information you will need to design your application, widget, or window manager so that it is consistent with Presentation Manager and the HP OSF/Motif user interface.

P.1 What's in this Style Guide

This style guide contains the following chapters:

1	Introduction	Discusses the concept of HP OSF/Motif design and its principles.
2	HP OSF/Motif Environment	Provides an overview of the elements of the HP OSF/Motif environment.
3	Consistent Behavior through Consistent Operational Models	Discusses the four basic models of HP OSF/Motif behavior: input focus, input devices, navigation, and, most importantly, object-action selection.
4	HP OSF/Motif Window Manager	Discusses the functional elements of an HP OSF/Motif-consistent window manager.
5	Client Area Design	Discusses the client area and its organization.
6	Controls	Discusses the types of controls and how to use them.
7	Menus	Discusses the different types of menus, their components, and how to use them.

8	Dialog Boxes	Discusses the characteristics of dialog boxes and dialog box actions, the graphical control elements of dialog boxes, the common dialog boxes, and how to design HP OSF/Motif-consistent dialog boxes.
9	International Implementation	Discusses the design of applications for international markets.
	Appendix A	Provides default keyboard and mouse button bindings.
	Glossary	Contains definitions of important terms used in the style guide.

P.2 Purpose

The *HP OSF/Motif Style Guide* provides a framework of behavior specifications to guide application developers, widget developers, and window manager developers in the design and implementation of new products consistent with Presentation Manager and the HP OSF/Motif user interface.

The style guide establishes a consistent behavior by drawing out the common elements from a variety of current behavioral models. The style guide anticipates the evolution of graphical user interfaces, and of behavior, as new technology becomes available and as the use of the HP OSF/Motif environment spreads. Thus research and the passage of time will enable additional elements of behavior to be added to the style guide.

In this light, the illustrations included in the style guide represent *reference implementations* and are offered as visual suggestions not appearance absolutes.

The specific details of coding the implementation into an application program, widget, or window manager are found in the other volumes of the HP OSF/Motif documentation set.

Clients using the X Window System must follow the guidelines set forth in the *Inter-Client Communication Conventions Manual* (ICCCM).

P.3 Other HP OSF/Motif Documents

The HP OSF/Motif documentation set includes the following manuals:

HP OSF/Motif Style Guide	This document.
HP OSF/Motif Programmer's Reference	Contains reference "man" pages for the HP OSF/Motif window manager, toolkit functions, and widgets.
HP OSF/Motif Programmer's Guide	Contains how-to information and programming examples for the toolkit, window manager, and user interface language.

Information on configuring and using the HP OSF/Motif Window Manager (MWM) can be found in *Using the X Window System*.

P.4 Conventions

The *HP OSF/Motif Style Guide* uses the following typographical conventions:

TABLE P-1. Typographical Conventions

Type Style	Description
italic text	A book title or emphasized text.
Key	A key corresponding to a key on the keyboard.
bold text	The definition of this term follows. Additionally, the term is defined in the glossary at the back of the style guide.
Helvetica text	A menu title or push button label.

Another convention used throughout the style guide regards pointing devices. For the sake of simplicity, the style guide uses "mouse" to refer to any and all pointing devices including but not limited to track balls, graphics tablets, joy sticks, and special sets of graphics navigation keys.

Additionally, throughout the style guide certain words are used in a particular sense when referring to what constitutes consistent behavior. For example, "should," "may," and "can" denote a recommendation; "must" denotes a prescription.

Introduction 1

This style guide was written for three audiences:

Application developers Those who write application programs that use an interface consistent with the HP OSF/Motif user interface.

Widget developers Those who create new widgets or modify existing widgets to add substantial new functionality to a widget set that complies with the HP OSF/Motif user interface.

Window manager developers Those who create new window managers or modify the functionality of existing window managers to be consistent with the HP OSF/Motif user interface.

As an application developer, widget programmer, or window manager developer, you should design and implement your code using the principles suggested by this style guide. Your software product's behavior will be consistent with Presentation Manager and with the OSF/Motif user interface. It also facilitates your participation in the widest possible market for your product without the expense of creating multiple, machine-dependent versions.

1.1 Empowering the User

To perform work effectively, people must control the tools they use. A major goal of your software should be to *empower the people who use your software*, giving them both the tools to get the job done and an easily achieved control over those tools.

People are in control of your product when you design it with the following in mind:

- Consistency.
- Direct manipulation.
- Flexibility.
- Explicit destruction.

Because of the particular nature of your application program, widget, window manager, or perhaps the needs of your customers, you may not be able to apply all of these principles all of the time. Still, you should be very cautious about not implementing them. You should never entirely abandon them.

1.2 Consistency

Above all else, an application must be consistent. It must be consistent within itself; but to be truly successful in the marketplace, it must be consistent with other applications that share the same environment.

Consistency means the following:

- Similar controls operate similarly and have similar uses. Pull-down, pop-up, and cascading menus are similar controls, their operation is similar, and their use is similar.
- The same action always has the same result. Clicking the Select mouse button on the Window Menu button of the window frame always displays the window menu. Double-clicking always performs the default action.
- The location of the mouse pointer is determined by direct manipulation and is not positioned arbitrarily by the needs of the application.

Additionally, an application presents its capabilities in an orderly manner. Necessary and commonly used functions are presented first and in a logical order. For example, essential functions could be included in a menu bar at the top of the client area where they are always visible and ready for selection.

More sophisticated or less frequently used functions could be hidden from immediate view but could still be available. For example, dialog boxes provide a mechanism for hiding settings and functions that are infrequently used.

Decisions about the placement of functions are not easy to make. From the implementation standpoint, all functions are important. Often, however, a relatively small number of functions account for the majority of usage. These functions need to be prominently featured, but they can be prominent only if other functions are hidden.

Consistency among applications increases people's sense of mastery. Experience with one application can be readily applied to another application, creating a positive transfer of knowledge. The focus of a computer session becomes the task at hand, not "learning a new application." When applications work in a manner that is consistent with other applications, people enjoy a feeling of immediate confidence in their ability to master the new program. Also, they are pleasantly surprised when trying new functions because, although new, the functions seem familiar.

1.3 Direct Manipulation

Direct manipulation describes the interaction between a person and an object. Direct manipulation connects an action to an observable response from an object. In direct manipulation interfaces, people experience the immediate visible result of an action.

The immediacy of the visual response is *crucial* to the experience of direct manipulation. Performance problems caused by inefficient program design and implementation make it difficult for people to concentrate on the task at hand and can render an otherwise well-designed application unusable.

Direct manipulation simulates the "real" world where people employ tools to perform tasks on physical objects. People control their HP OSF/Motif-consistent environments by directly manipulating graphical controls similar to controls they have encountered in real life; for example, buttons "push" to start an action and the slider on a scale actually slides to select a setting.

Another feature of direct manipulation is that the output of the application is also available as input. For example, a list of files is not only the result of a command, but also a collection of screen objects that a person can act upon.

Direct Manipulation empowers people by enabling them to manipulate objects by "grabbing" them (or "pointing" and "selecting" them) rather than by typing a command on a command line. Empowering people through direct manipulation means reducing wherever possible the amount of information they must memorize.

1.4 Flexibility

Flexibility should be apparent in both operability and configurability.

Providing multiple ways for people to access application functions and accomplish their tasks increases their sense of control. For example, a function could be accessed through a pull-down menu, a mnemonic key press, or a keyboard accelerator. Empowering people through flexibility enables them to select the best method for accessing a function based on criteria *they* choose: experience level, personal preference, unique situation, or simply habit.

Allowing people to configure settings and select personal preferences enhances their sense of control and encourages them to take an active role in understanding your product and how it works. To be effective, the configurability of your application must be easily accessible.

1.5 Explicit Destruction

Explicit destruction means that, if an act has irreversible negative consequences, it should require people to make an explicit action to perform it.

For example, while a worksheet could be saved by clicking the Select mouse button or typing the [Select] key on a Save push button, to delete the worksheet should require clicking the Select mouse button or typing [Select] on an Erase push button *and* answering some type of an "Are you sure you want to erase this worksheet?" question with another selection action.

Warnings protect people from inadvertent destructive operations yet allow them to remain in control of the application. Operations that may cause a serious or unrecoverable loss of work should warn people of the consequences and request explicit confirmation.

Anticipating errors enables you to support recovery attempts and to provide messages informing people of the proper corrective action. Part of this support could be context-sensitive help and the ability to undo actions. One technique for avoiding excessive errors is to make controls temporarily unavailable when it would be inappropriate to use them. Unavailable controls should provide a visual cue that they are not currently operable.

Context-sensitive help aids understanding, reduces errors, and eases recovery efforts. Help information text should be clear, concise, and written everyday language. Help information must be readily accessible and just as readily removable.

Many people are most comfortable with learning when they use a natural, trial-and-error method. An "undo" function supports learning by trial and error by minimizing the cost of errors. The undo function allows people to retract previous actions, facilitating the spirit of exploration and experimentation that is essential for learning.

1.6 What is the HP OSF/Motif Environment

The HP OSF/Motif environment is a graphical user interface combining a toolkit, presentation description language, window manager, and style guide.

Toolkit

The HP OSF/Motif toolkit is a rich and varied collection of widgets and gadgets for building HP OSF/Motif-consistent applications. The toolkit provides a common graphical interface upon which the window manager is based. The behavior of the toolkit is consistent with Microsoft's Presentation Manager (PM), ensuring an easy transition between PC and workstation environments.

Toolkit widgets provide a 3-D reference appearance that gives people real-world, visual cues to the effects of their actions.

Presentation Description Language

The HP OSF/Motif presentation description language allows application developers and interface designers to create simple text files which describe the visual properties and initial states of interface components. Changes to components are made in the text file, eliminating the need to change application code when tuning an interface.

Window Manager

The window manager works with the toolkit to manage the operation of windows on the screen. The window manager provides functions for moving and resizing windows, reducing windows to icons, restoring windows from icons, and arranging windows on the workspace. The HP OSF/Motif window manager provides compatibility with PM behavior. An additional HP OSF/Motif window manager feature is the icon box. The icon box contains icons for all windows operating under the window manager.

Style Guide

This style guide describes a consistent behavior for window managers and toolkits. It is a guide to usage, providing application writers with guidelines for using toolkit widgets, widget writers with guidelines for designing new widgets, and window manager writers with guidelines for designing new or customized window managers.

Together, these four elements provide a consistent user interface behavior for applications.

The HP OSF/Motif Environment 2

This chapter provides a general description of the following components of the HP OSF/Motif environment:

- The elements of the HP OSF/Motif graphical environment.
- The input devices available in the HP OSF/Motif environment.

The chapter also presents an overview of how people, typical knowledge workers busy in the lab, at the drafting table, at the office, or elsewhere, might view and interact with the HP OSF/Motif environment.

2.1 The Elements of the HP OSF/Motif Graphical Environment

At the highest level, the HP OSF/Motif environment is composed of the following three elements:

- An input selection model.
- A window manager.
- Application programs.

2.1.1 The Input Selection Model

The HP OSF/Motif environment is based on an object-action input selection model. The selection model defines the actions that people must perform to control the window manager and applications in the HP OSF/Motif environment.

The selection model follows a point-and-click paradigm. In a point-and-click, object-action selection model, people first point at and select an object with which to work and then point at and select an action to perform on that object.

The HP OSF/Motif selection model is discussed in chapter 3.

2.1.2 The Window Manager

A window manager provides people with a way to manipulate the windows that display in their HP OSF/Motif environment. Typical manipulations are to move, resize, minimize, and maximize windows, arranging them as required on the workspace.

The HP OSF/Motif Window Manager (MWM) frames application windows with an eight-segment border that can be stretched to resize the window. A title area supplied by the window manager displays a title for the window and can be used to move the window. Graphical buttons embedded in the window manager frame provide a window management menu and other window controls. Additionally, the HP OSF/Motif window manager has a three-dimensional appearance so that the control buttons, when "pressed" by the mouse pointer, actually look like they have been pressed.

The window manager helps provide for consistent behavior from one application to the next and is discussed in chapter 4.

2.1.3 Application Programs

Application programs fill the space inside the window frame and are specific to the work that people are trying to accomplish. By following the selection model and the guidelines set down in the rest of this style guide, your application will have a behavior that is consistent with the behavior of all other applications in the HP OSF/Motif environment.

Designing applications with consistent behavior and the proper use of controls are discussed in chapters 5 through 8.

2.2 Input Devices

To be productive using a computer, people must be able to put data into the computer and guide the computer's processing of that data. To do this, they need input devices to speed communication and economize control.

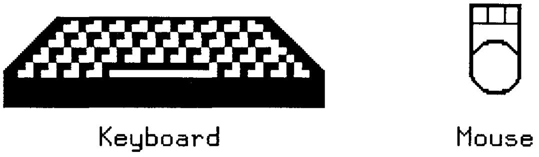


Figure 2-1. Two Popular Computer Input Devices

Figure 2-1 illustrates two popular input devices, the traditional typewriter-style keyboard and the mouse pointer device. The typical HP OSF/Motif environment uses both a keyboard and a pointer device (usually a mouse).

2.2.1 The Keyboard

The HP OSF/Motif environment doesn't require people to be keyboard experts to control their computer environment or to be productive. HP OSF/Motif application developers design their programs under the premise that most people aren't interested in becoming expert typists or in learning the arcane syntax of command line instructions. HP OSF/Motif application developers rely on the natural inquisitiveness of people. People who know what they want to do only have to point to an object and select an action from a list of possible actions to get their work done.

For those people who have become skilled keyboard typists, the HP OSF/Motif environment offers shortcuts to the normal point-and-click selection style. Single-key selection mnemonics and special key combinations enable keyboard users to speed the selection process without recourse to a pointing device.

The HP OSF/Motif user interface recognizes the variety of keyboards available, each with a slightly different set of keys and with keys in slightly different locations. The HP OSF/Motif style recommends the use of certain keys and suggests substitutions for those cases when keys aren't available on a keyboard.

2.2.2 The Mouse Pointing Device

In the HP OSF/Motif environment, the control over the computer and application programs that traditionally relied on a person's ability to type has been replaced by the use of pointing devices such as the mouse.

A mouse enables people to control most operations using just three actions:

Pointing — Pointing the mouse pointer at an object or action signals a possible interest in that object or action.

Clicking — Clicking a mouse button selects the object or performs the action.

Dragging — Dragging the mouse pointer enables a person to move objects around in the HP OSF/Motif environment, to change the size of windows, or to browse a menu (depending on the context of the situation).

The mouse pointer changes shape to indicate the current operation being performed by an application. The HP OSF/Motif toolkit includes many pointer shapes: pointing, moving, resizing, selecting, drawing, and waiting to name a few. Figure 2-2 illustrates these shapes.

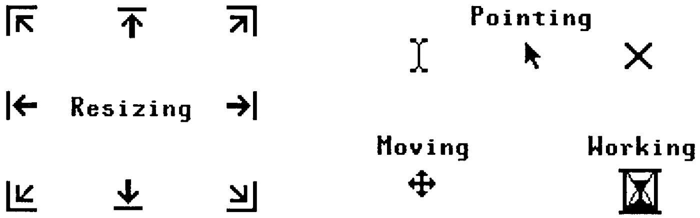


Figure 2-2. Pointer Shapes Provide Visual Cues to Activity.

Each shape is visually descriptive of the operation being performed and provides a visual cue to the state of the application.

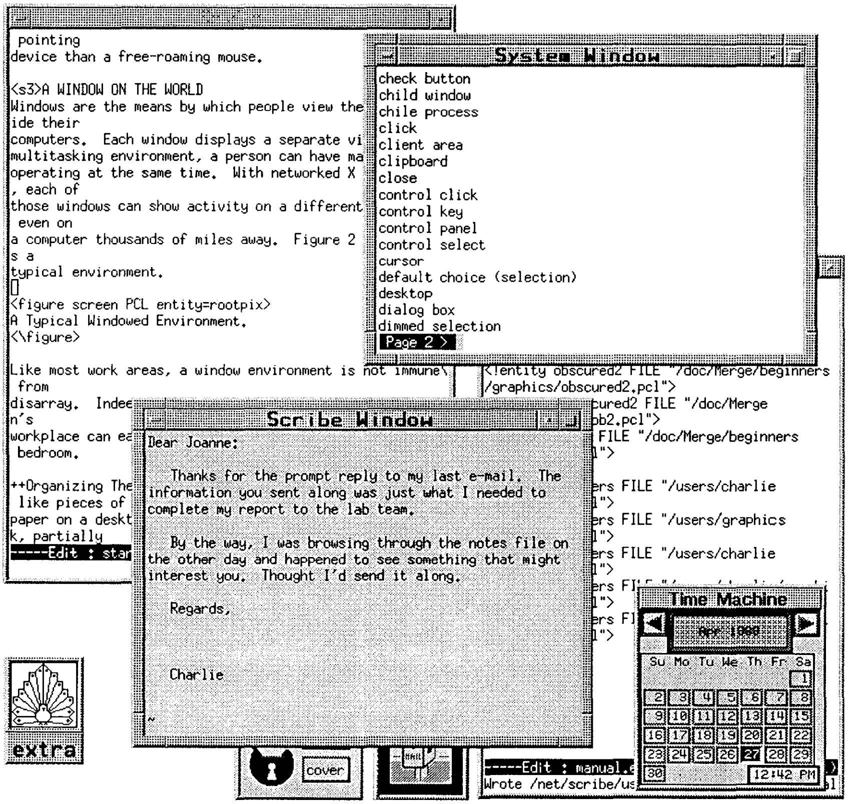
2.2.3 Special Tools for All Occasions

The traditional style keyboard and the mouse are by no means the only "hand tools" available in the HP OSF/Motif environment. OSF/Motif supports the use of many special tools for just about all occasions. Which tools a person uses to be most productive depends on the person, the situation and the application, not on the HP OSF/Motif environment.

A mouse might not be appropriate as a pointing device in all cases. If the application is computer-aided design (CAD), perhaps a graphics tablet or light pen would be a better choice. If the situation is such that only a minimum of deskspace exists, perhaps a track ball would be a better choice of pointing device than a free-roaming mouse.

2.3 Viewing the World through Windows

People using the HP OSF/Motif user interface view the world inside their computers through windows. Each window displays a separate view. In a multi-tasking environment such as the HP OSF/Motif environment, a person can have many windows operating at the same time. Figure 2-3 illustrates a typical HP OSF/Motif environment.



```
 pointing
device than a free-roaming mouse.

<s3>A WINDOW ON THE WORLD
Windows are the means by which people view the
ide their
computers.  Each window displays a separate vi
multitasking environment, a person can have ma
operating at the same time.  With networked X
, each of
those windows can show activity on a different
 even on
a computer thousands of miles away.  Figure 2
s a
typical environment.

<figure screen PCL entity=rootpix>
A Typical Windowed Environment.
<\figure>

Like most work areas, a window environment is not immune\
 from
disarray.  Indee
n's
workplace can ea
 bedroom.

++Organizing The
 like pieces of
paper on a deskt
k, partially
-----Edit : star
```

```
check button
child window
chile process
click
client area
clipboard
close
control click
control key
control panel
control select
cursor
default choice (selection)
desktop
dialog box
dimmed selection
Page 2 >
```

```
Dear Joanne:

   Thanks for the prompt reply to my last e-mail.  The
information you sent along was just what I needed to
complete my report to the lab team.

   By the way, I was browsing through the notes file on
the other day and happened to see something that might
interest you.  Thought I'd send it along.

   Regards,


   Charlie

~
```

```
<!entity obscured2 FILE "/doc/Merge/beginners
/graphics/obscured2.pcl">
cured2 FILE "/doc/Merge
b2.pcl">
 FILE "/doc/Merge/beginners
l">

ers FILE "/users/charlie
l">
ers FILE "/users/graphics
l">
ers FILE "/users/charlie
l">
ers FI
l">
ers FI
l">
-----Edit : manual.
Wrote /net/scribe/us
```

```
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30             12:42 PM
```

Figure 2-3. A Typical HP OSF/Motif User Environment.

Like most work areas, the HP OSF/Motif environment is not immune from disarray. Indeed, with remarkably little effort a person's workplace can very easily become cluttered.

2.3.1 Getting Things Straight

Fortunately, people using the HP OSF/Motif environment have a number of ways to visually reorganize their work area.

In the HP OSF/Motif environment windows overlap each other, stacking on the screen like pieces of paper on a desk. New windows open on top of the stack, partially obscuring older windows lower in the stack.

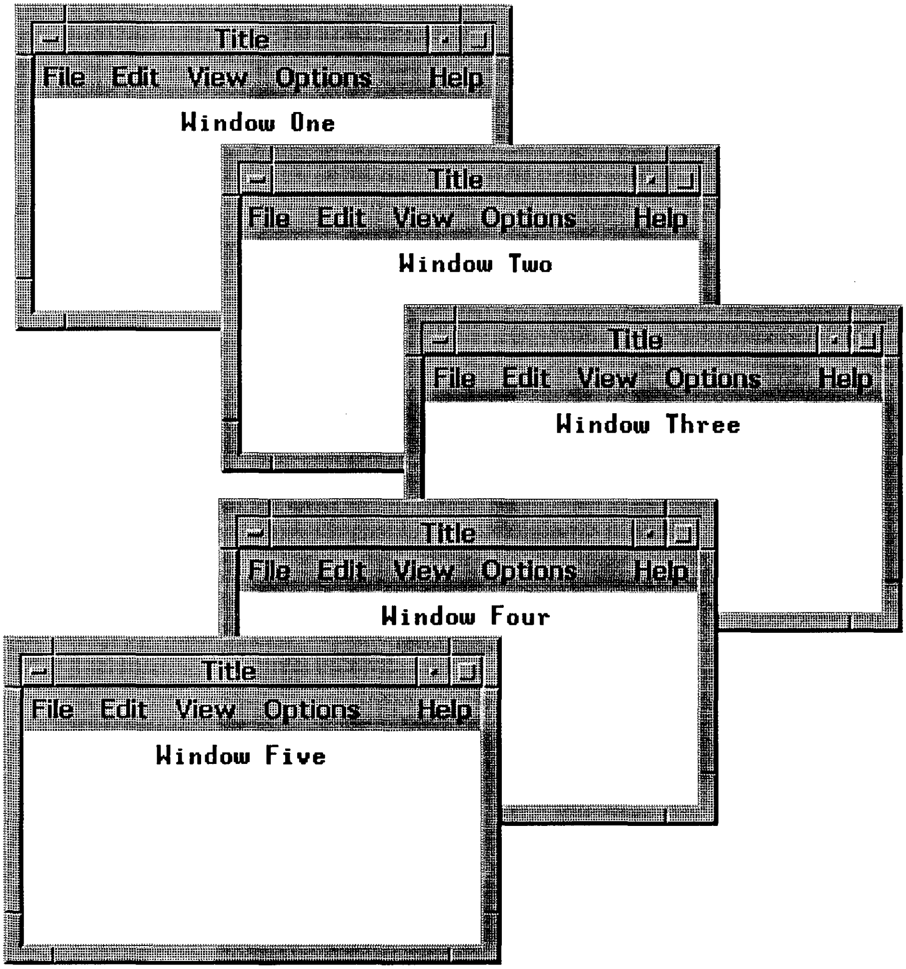

Title
File Edit View Options Help
Window One
Title
File Edit View Options Help
Window Two
Title
File Edit View Options Help
Window Three
Title
File Edit View Options Help
Window Four
Title
File Edit View Options Help
Window Five

Figure 2-4. Windows Appear Stacked on the Work Area.

However, people can raise a window to the top of the window stack as easily as they can lift a sheet of paper on their desk to the top of the paper stack. People can also move windows around on the screen in the same manner they push papers out of the way on their desks.

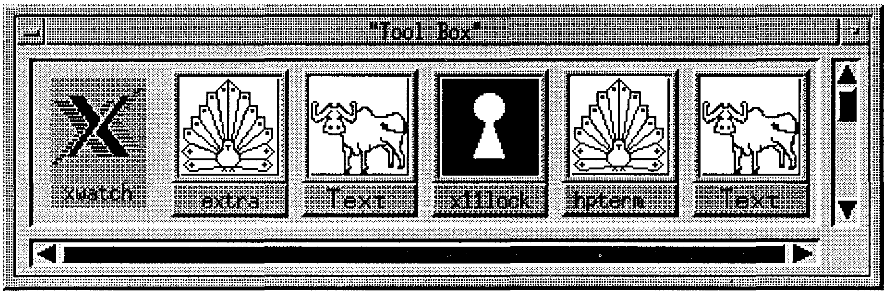


Figure 2-5. Windows Can Be Changed into Icons to Save Space.

And, better than a desk, people using the HP OSF/Motif environment can change the size of windows (without scissors), enlarging or shrinking them to fit the situation. People also have the option of minimizing the window, turning it into a small graphical icon. Minimizing saves space on the work area without halting the application that was running in the window. This is analogous to a person's putting the phone in the bottom desk drawer: The phone still works, it's just out of the way.

2.3.2 Controlling the View

The HP OSF/Motif environment is designed around the concept of direct manipulation so that all the graphical "objects" that appear in a person's work area can be acted upon directly. In other words, when a people want to move a window, for instance, they latch onto the window's title area and drag the window to the new location. As they drag, the window actually appears to move across the screen.

The HP OSF/Motif environment provides a number of ways for people to control the windows in their work areas – the idea being that no one way will be correct for everyone, so, by building flexibility into the environment, people can pick a way to manage windows that is the most natural for them or that best fits the situation.

Every window has a window menu that can be displayed either by pressing the window menu button with the mouse or by typing Shift + ESC. The window menu contains a complete list of all the controls people have over the window itself.

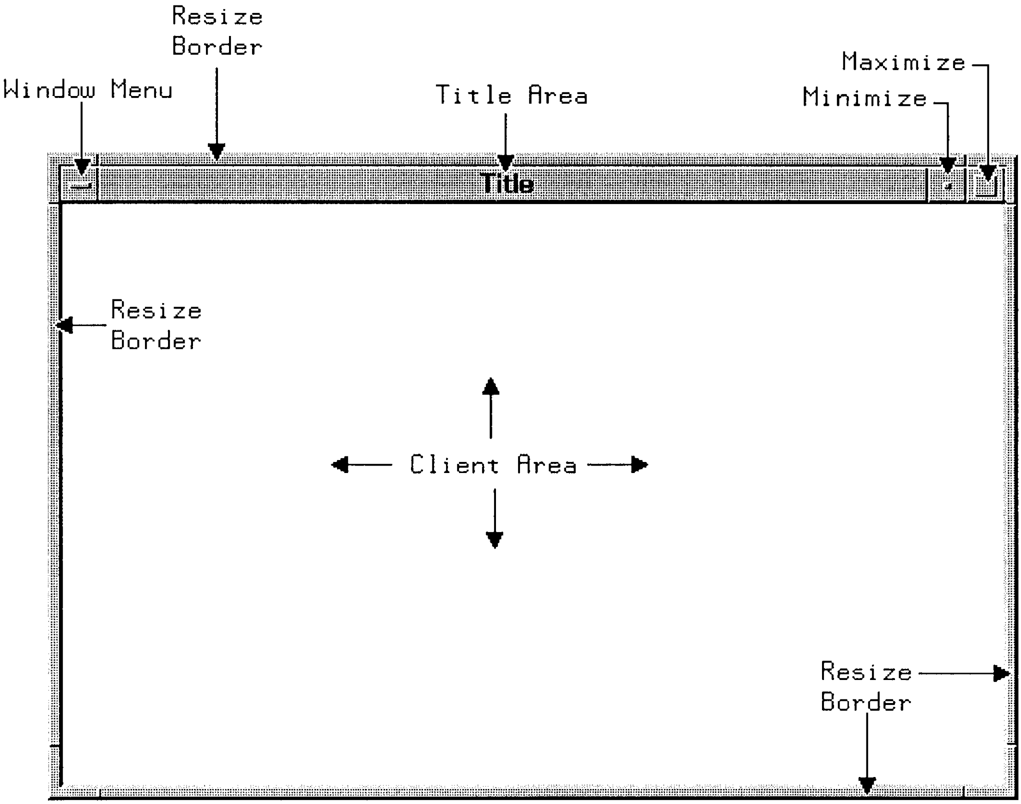


Figure 2-6. The Anatomy of An HP OSF/Motif Window Frame.

Because the HP OSF/Motif environment is designed for utility, the most frequently used window controls are accessible directly from the window frame. Thus the window frame itself is not just a border; when grabbed by the mouse, the eight border segments stretch or shrink, resizing the window. When the title area is grabbed, the entire window can be moved to a new location without affecting either the window size or its contents. Additionally, window control buttons (mounted on the frame and similar in operation to the window menu button) provide shortcuts to important control functions.

2.4 Commanding the Application in the Window

In addition to the controls listed in the window menu and embedded in the window frame itself (general window management controls), each application program has controls particular to the tasks it performs. These application-specific controls are accessible either through the menu bar at the top of the client area of the window, or through control panels, pop-up menus, or dialog boxes. See Figure 2-7.

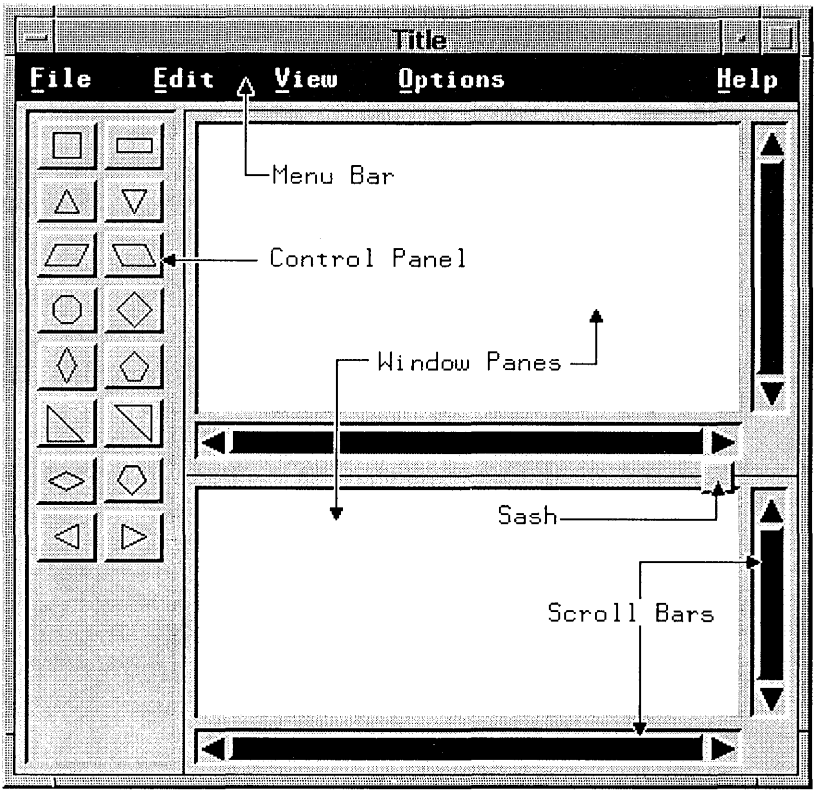


Figure 2-7. The Anatomy of A Client Area.

The menu bar lists the titles of available menus. To display a menu, a person clicks on the menu's title with the mouse or uses the underlined mnemonic letter associated with that title (for example, F in the case of the File menu).

Since many applications open, save, and close files, and perform other common actions, the HP OSF/Motif user interface provides several menus so that common actions are accessed and performed in the same way.

2.4.1 Common Menus for Common Actions

The common actions identified by the HP OSF/Motif user interface include file actions, edit actions, optional actions, and help actions. Each of these sets of actions has its own menu in the menu bar.

File — Contains file actions like opening, creating, saving, and printing a file.

Edit — Contains edit actions like undoing, cutting, copying, pasting, and clearing edits to a file.

View — Contains actions for viewing data as objects, text, sorted in some fashion, or some other criteria.

Options — Contains frequently used application-specific options.

Help — Contains helpful information like context-sensitive instructions, information on the usage of keys, index listings of help topics, and information on how to use the help function.

To initiate an action from a menu, people choose the appropriate menu selection, just like ordering dinner at a restaurant.

If a person is a mouse user, the person positions the mouse pointer over the selection and clicks a mouse button.

If a person is a keyboard user, the person can type a single-character mnemonic from the menu or use a keyboard accelerator without displaying the menu first.

2.5 Control through Graphics

Besides the menu bar, the HP OSF/Motif user interface provides other similar graphical control objects, enabling people to control their computer systems and any applications running on them by direct manipulation.

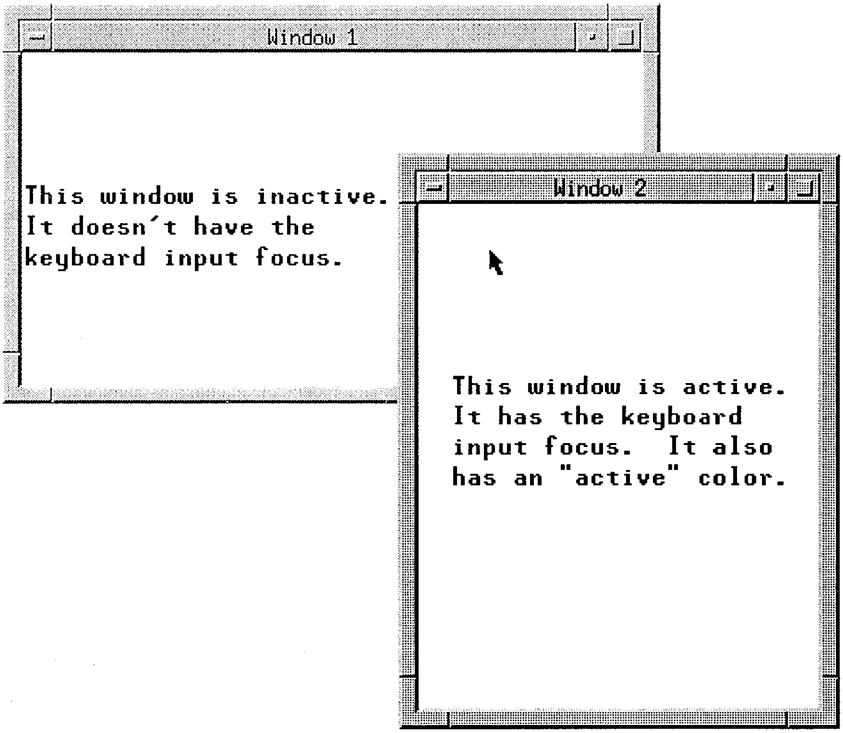
Graphical controls are modeled after real-life objects – the radio buttons on a car stereo, the push buttons from many electrical appliances, the check buttons that appear on many applications and other forms – objects with which most people are already familiar because they use the objects in everyday life. Because the graphical objects are modeled after objects that are already part of peoples' ordinary experience, the familiarity with the real tools makes the manipulation of the graphical tools provided by the HP OSF/Motif user interface seem intuitive.

Graphical tools enable people to control their communication with applications by performing three choice-type functions:

- Select an object
- Perform an action
- Adjust a setting

2.5.1 Selecting Objects

People select objects using the mouse and a point-and-click method – point at the object to be selected with the mouse pointer and click the mouse button. A common example of object selection occurs as a person who has been working in one window switches to a different window: The person points to the new window with the mouse pointer and clicks a mouse button to signal selection.



This window is inactive.
It doesn't have the
keyboard input focus.

This window is active.
It has the keyboard
input focus. It also
has an "active" color.

Figure 2-8. Changing to a Different Window is an Example of Object Selection.

HP OSF/Motif applications use the pull-down menus located on the menu bar to make frequently used selections readily available. In all cases, the object selected provides some visual cue as feedback to indicate that the object has actually been selected. Typically this cue is a change in the color or shape of the graphic representing the object.

2.5.2 Performing Actions

People perform actions using the same point-and-click method as for selecting. This provides HP OSF/Motif programs with a consistency of operation that shortens learning time by keeping the number of commands to a minimum.

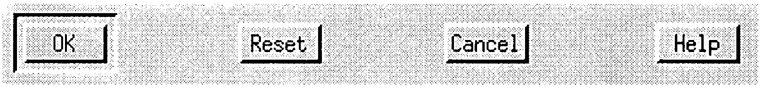


Figure 2-9. A Set of Push Buttons Make Action Performance Easy.

HP OSF/Motif applications use menus such as the pull-down menus of the menu bar and push buttons such as those in a dialog box, to make frequently used actions readily available.

2.5.3 Adjusting Settings

People adjust settings again using the point-and-click model for consistency of operation.

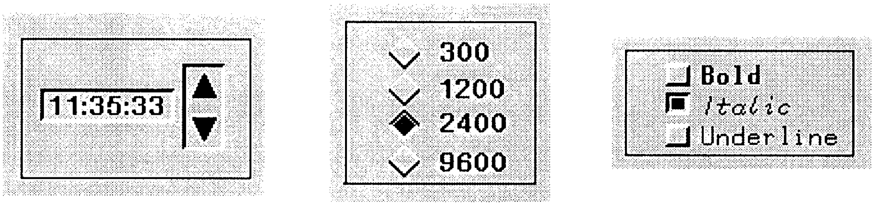


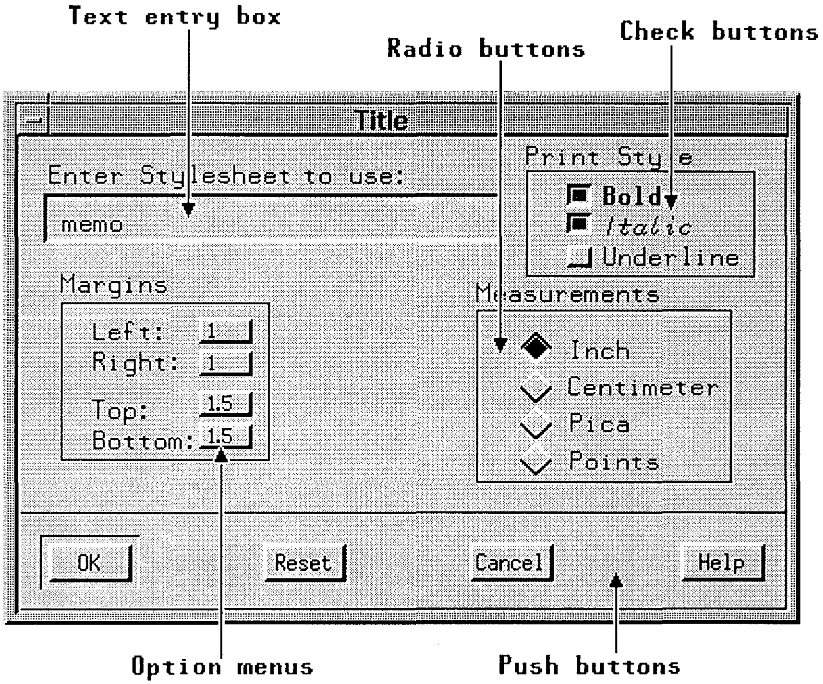
Figure 2-10. Several Graphical Controls Adjust Settings.

HP OSF/Motif applications provide several graphical controls that simulate the devices used for making adjustments to settings in the real-world. These graphical controls include check buttons, radio buttons, and dials to name a few. Each control is used in a specific situation.

To further the analogy with the real world, HP OSF/Motif graphical controls can be organized into "control panels."

2.6 Talking Back with Dialog Boxes

Dialog boxes are so named because they contain the graphical controls that enable people to communicate (carry on a dialog) with an application.



Enter Stylesheet to use: memo

Margins
Left: 1
Right: 1
Top: 1.5
Bottom: 1.5

Print Style
☒ **Bold**
☒ *Italic*
□ Underline

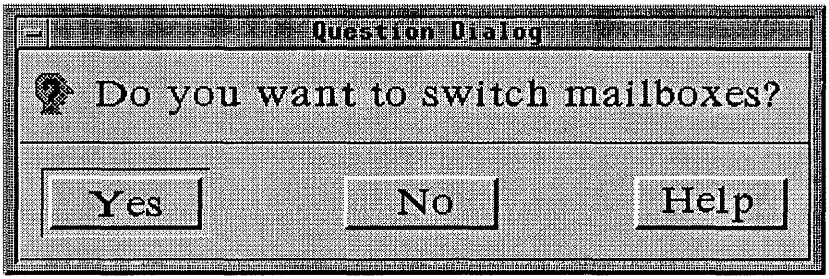
Measurements
● Inch
○ Centimeter
○ Pica
○ Points

Figure 2-11. A Dialog Box with Control Panels.

Dialog boxes contain control panels featuring sets of graphical controls grouped according to common function.

2.7 Getting the Message

A message box is a special type of dialog box. The main purpose of a message box is to display a message, warning, or other information specific to the task at hand. Compared to the "average" dialog box which has multiple controls, a message box typically contains only a message and a couple of push button controls), and requires only one response.



Do you want to switch mailboxes?

Figure 2-12. A Question Box Is a Typical Message Box.

Consistent Behavior through Consistent Operational Models 3

The HP OSF/Motif environment provides people with a behaviorally consistent graphical user interface. Using four distinct but related operational models, much of the interaction with an application program can be reduced to a simple scenario: People select an object using the input device with which they feel most comfortable; they then select an action to perform on that object.

Consistent behavior augments people's ability to perform a task by enabling them to focus on the task itself rather than on the tools or the methodology they use to perform the task. Just as an automobile is a tool for transportation that doesn't require the average person to be a mechanic, so too a user-oriented software application provides people with a productivity enhancing tool without requiring them to become computer engineers.

In part, HP's goal of consistent behavior can be achieved by your full understanding and consistent use of the following four operational models:

- The input focus model.
- The input device model.
- The navigation model.
- The object-action selection model.

3.1 The Input Focus Model

A typical HP OSF/Motif workspace can contain many windows. Like sheets of paper stacked on a desk, windows can be stacked on the workspace. Some windows may be partially or completely obscured by other windows. At any given time, however, only *one* window has the **input focus**. The window with the input focus is known as the **active window** and is the window where keyboard input will appear. The active window is also the only window that has the location cursor. Input focus can be moved from window to window.

When a window receives the input focus, the default behavior is to move the window automatically to the "front" of the workspace so that no part of it is obscured by other windows. If inappropriate, this behavior can be modified to better suit the needs or preferences of the people using your application. Additionally, the window with the input focus has a highlighted frame, supplied by the OSF/Motif Window Manager (mwm). This provides a visual cue that the window is active and distinguishes the active window from any inactive windows in the workspace.

The input focus model can use either an explicit or an implicit policy for moving the input focus from one window to another. Explicit focus is the default.

In **explicit focus**, people explicitly select which window receives the input focus. With a mouse, people move the mouse pointer into a window *and* press the Select mouse button to move the input focus to that window. In windows that contain graphical controls, the selecting of a control includes the moving of the input focus and no other explicit action is required. With a keyboard, people press Alt + Esc to move the input focus sequentially through the windows stacked in the workspace.

In explicit focus, the action being performed in a window will not be disrupted by inadvertent mouse or stylus movements. Explicit focus is sometimes known as "click-to-type" and, as mentioned, is the default behavior.

In **implicit focus**, the input focus moves to the window into which people move the mouse pointer. No explicit selection action is performed. The actual focus policy that a person chooses to implement is a matter of personal preference and the needs of the situation. Some people who use mice find implicit focus more convenient because the selection of the active window is implicit in the moving of the mouse pointer and requires no other operation. For people using keyboards only, there is no distinction between explicit and implicit focus. Implicit focus is sometimes referred to as "track pointer," "track listener," or as being "real estate driven."

3.2 The Input Device Model

Similar to other user interfaces for computers, HP OSF/Motif has two input devices that enable people to communicate with your application:

- A keyboard.
- A pointing device.

3.2.1 Using a Mouse vs Using a Keyboard

Some people enjoy controlling software programs by pointing and clicking a mouse. Other people dislike removing one hand from the keyboard to "catch" a mouse and prefer to control programs solely from the keyboard. Other people lack the room for a pointing device on their desk. Still others prefer to mix mouse usage with keyboard usage: For frequently performed functions, they use keyboard accelerators; for other functions, they point and click with the mouse.

In HP OSF/Motif-consistent applications, the keyboard is virtually interchangeable with the mouse. This enables people to pick the appropriate tool for the job or to choose the tool with which they feel the most comfortable.

You should design your application so that people can control your application using a pointing device, the keyboard, or both. Although you may decide to make the pointing device the primary means of control, you should not constrain people from using the keyboard.

Designing your application for this type of "dual accessibility" empowers people by enabling them to choose the input device they find best for the task at hand given their particular work situation and personal preferences.

3.2.2 The Keyboard Has Its Uses

The keyboard conventions outlined in this section work with ANSI keyboards. Keyboards, however, can differ greatly from manufacturer to manufacturer, and even between one model and another from the same manufacturer. Therefore, exact key labels as described here may differ from those that you are used to seeing. For example, Alt is sometimes labeled Extend or Extend char.

A Functional Anatomy of the Keyboard

In general, all keyboards have a similar functional anatomy composed of the following six parts:

Alphabetic keys

Keys representing the letters of the alphabet, the marks of punctuation, and text-formatting keys such as Tab, Return, and Spacebar.

Numeric keys

Keys that represent the numbers from 0 to 9. These are typically included as the top row of the alphabetic keys or, just as frequently, in a numeric keypad on one side of the keyboard.

Navigation keys

Keys that are pressed to move the insertion cursor. These keys are commonly called the "arrow" keys and have labels like (→, ←, ↑, and ↓. On some keyboards they are separate keys, on others they are included as part of the numeric keypad. Also included in this category are keys like Home, End, PgUp, and PgDn.

Modifier keys

Keys that are used in conjunction with other keys to "modify" the meaning or effect of those other keys. The modifier keys include Ctrl, Shift, and Alt. If your application designates a particular key as a modifier key, that key should not have any other function associated with it.

Special-purpose keys

Keys that have particular functions and frequently have labels stating their purpose. Among these keys are Help, Menu, Esc, Select, Enter, Delete, Backspace, and Insert
.

Function keys

Keys that most keyboards provide for "extra" or general functions. Usually these keys are labeled f1, f2, and the like. Some keyboards have ten function keys, others twelve or more. The HP OSF/Motif environment assumes a keyboard with at least ten function keys. Function keys are usually placed either across the top of the keyboard or on one side, often the left.

As noted, keyboards differ greatly. Some keyboards may have keys better suited to your purpose while others may not have all of the keys just mentioned. If the keyboard you use has keys better suited to your application, use those keys. If the keyboard you use doesn't have suitable keys, table 3.1 lists some common substitutions.

TABLE 3-1. Common Keyboard Substitutions.

Key	Description	Substitution
Select	Makes a selection from the keyboard	Spacebar, Return, or Ctrl + Spacebar
Menu	Invokes a pop up menu	F4
Help	Invokes the help function	F1
Alt	Modifies the meaning of another key	Extend
Esc	Cancels current action	F12
Enter	Invokes the default action	Return
Next Field	Moves location cursor to next field.	Tab or Ctrl + Tab
Previous Field	Moves location cursor to previous field.	Shift Tab or Ctrl + Shift + Tab

Also, some keyboards may have "duplicate" sets of keys. Numeric keys, for example, are often found both in the top row of alphabetic keys and in a numeric keypad; function keys are sometimes found both at the top of the keyboard and as a separate keypad on one side. These "duplicate" keys may have different keycodes associated with them. In such a case, you can design your application to provide some special functionality on the "extra" keys. For example, you might retain the top row of alphabetic keys as numeric keys and use the numeric keypad for special functions by providing some type of keyboard overlay.

Cursor Shapes

While the traditional use of a keyboard has been for text entry, the keyboard has also been adapted for use as a pointing device. Because of this, the keyboard has two cursors, one for location, the other for text insertion.

The **location cursor** reflects movement on the screen using the keyboard's cursor navigation keys and indicates the current location of the keyboard input focus. The position of the location cursor gives people a visual cue to which object they are about to select. The location cursor is displayed whether or not a mouse is attached and is in addition to the mouse pointer. For example, the location cursor can indicate which item in a list box a person wants to select. The location cursor is often shown as a rectangle around a control as shown in figure 3-1 below. The location cursor is sometimes called the "selection cursor."

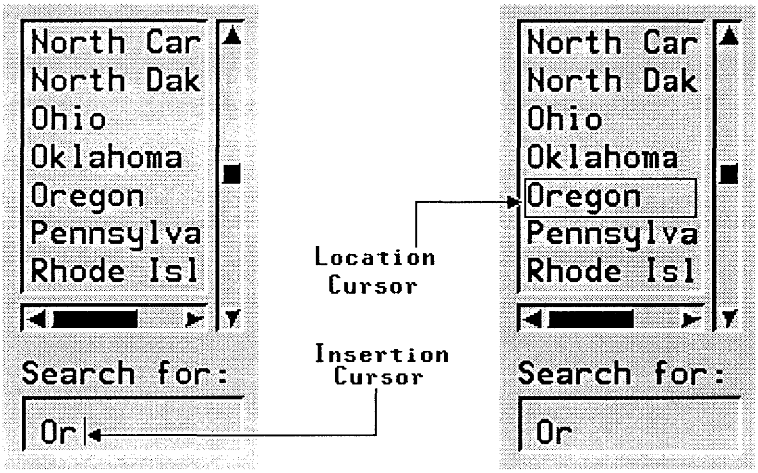


Figure 3-1. A Typical Location Cursor.

The **insertion cursor** is used in edit controls such a text entry box. It indicates the point in text or graphics where new characters will be inserted. In text entry, the insertion cursor's default shape is a pipe (|) or bar for inserting, and a block character cell or underscore (_) for overstriking. In graphics entry, the insertion cursor's shape might be a pencil, paintbrush, or some other graphically descriptive image.

Figure 3-2 shows typical insertion cursor shapes.

	Insertion	Overstrike
Active	|	▮ or _
Inactive	┆ or ^	▒ or ﹏

Figure 3-2. Typical Text Cursor Shapes.

The insertion cursor is *only* displayed where graphics or text entry is allowed. The insertion cursor should also change size to match the size of the current font.

3.2.3 Pointing Devices Provide Direct Manipulation

Direct manipulation allows people to control an application by choosing selections from a menu and by setting controls following a point-and-click method. In the point-and-click method, people move the mouse until the mouse pointer is over ("points to") the desired object. They then click the Select mouse button.

Direct manipulation usually requires a pointing device: a mouse, graphics tablet, track ball, joystick, or some other such device. Typically, keyboards with two sets of arrow keys use one set for mouse pointer navigation and the other set for location cursor navigation. Using the pointing device, people can navigate rapidly around the screen and can point at and choose objects to work on and actions to perform.

Mouse Buttons

People use mouse buttons in combination with the mouse pointer to make selections, move the input focus, or position the insertion cursor.

The HP OSF/Motif environment assumes a three-button mouse and gives the following names to mouse buttons.

TABLE 3-2. Mouse Button Names

Button	Name	Description
1	Select	Used for selection.
2	Custom	Used as needed for application-specific functionality.
3	Menu	Used to display pop up menus.

If your application requires five mouse buttons, you can emulate buttons four and five using a three-button mouse. Pressing buttons 1 and 2 together activate button 4. Pressing buttons 2 and 3 together activate button 5.

Note that the position of mouse buttons can vary depending on whether the mouse is configured for left- or right-handed operation. Additionally, the position of the Menu button is configurable. Mouse button positions are illustrated in figure 3-3.

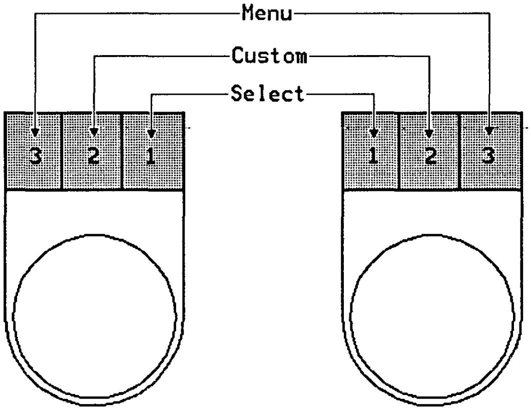


Figure 3-3. Button Positions for Left- and Right-Handed Mice.

The HP OSF/Motif environment includes the following mouse button operations:

Pressing — Holding down a mouse button.

Releasing — Releasing a mouse button after it has been pressed.

Clicking — Quickly pressing and releasing a mouse button without moving the mouse.

Dragging — Moving the mouse while a mouse button is pressed.

Double-Clicking — Clicking a mouse button twice in rapid succession without moving the mouse pointer.

The Pointer

The mouse is associated with one and only one **mouse pointer**. The mouse pointer appears on the workspace and represents the location of the mouse. Movements of the mouse pointer correspond to movements of the mouse. The mouse pointer is not confined to any specific application. People can move it anywhere on the workspace. The mouse pointer is sometimes known simply as the "pointer."

Your application should only interpret the mouse pointer position; it should *not* attempt to change it. To do so would violate people's trust in the consistency of your program and their sense of control. Also, changing the mouse pointer location may create problems in applications that use absolute location devices like graphics tablets.

Pointer Shapes

The shape of the mouse pointer provides people with an important visual cue indicating the functionality of the area in which the mouse pointer is currently located. Figure 3-4 illustrates the common mouse pointer shapes. While you shouldn't create new mouse pointer shapes for functions that are already defined, you can create mouse pointer shapes to represent functions that have no predefined shape. Also, you shouldn't use a predefined shape to symbolize a function it was not designed to represent.

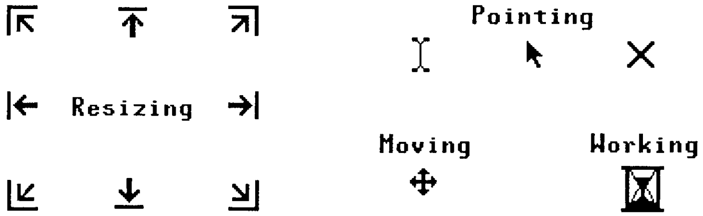


Figure 3-4. Mouse Pointer Shapes Provide Visual Cues to Activity.

Designing Pointers

If you decide to use other mouse pointer shapes in your applications, avoid shapes that are hard to see, that are hard to comprehend, that create visual clutter, and that flicker excessively when changing shape repeatedly.

You should also ensure that any mouse pointer you create has an obvious **hot spot** (active point), the area of the pointer image that marks the location of the mouse pointer to the computer. This is particularly true for mouse pointer shapes that point to objects or specify positions. People should be able to intuit the location of the hot spot, for example, the tip of an arrow or the center of a crosshair.

You should design your application so that people can set the ratio of mouse pointer speed to mouse speed. This ratio is called the **gain**. Mouse pointer speed can be constant or accelerating. Since the gain is typically set globally in the HP OSF/Motif environment, if your application needs to adjust the gain, you should implement a zoom feature rather than change the gain for a single application. A zoom feature (similar to the zoom lens of a camera) would adjust the gain by varying the "magnification" of the application.

3.3 The Navigation Model

Regardless of whether people use a mouse, a keyboard, or some combination, they will need to move the mouse pointer and the location cursor to new positions. That is, they will need to navigate around the workspace.

The mouse pointer is a graphical representation of the current location of the mouse device. The only way to move the mouse pointer is to physically move the mouse device.

The location cursor shows the current location of the keyboard focus. A person can control the location cursor with either the mouse or the keyboard. Pressing the Select mouse button on a selectable object moves the location cursor to that object. With auto-selection (see "Auto-selection" later in this chapter), navigation also implies selection.

3.4 The Object-Action Selection Model

In **object-action selection**, people first select an object, and then select an action to perform on that object. Object-action is patterned after real life and provides people with a readily comprehensible operational model for HP OSF/Motif applications.

It is helpful to note that the term **object** includes not only recognizable objects like windows and push buttons, but also objects, such as the individual letters of a text file, that are perhaps less recognizable as discrete entities.

The HP OSF/Motif selection model has the following kinds of selection:

- The selection of a single object.
- The selection of a range of objects.
- The selection of additional (non-contiguous) objects, including multiple ranges.

To make selections in HP OSF/Motif-consistent applications, people always use the same basic steps: first, they place the pointer (mouse) or location cursor (keyboard) on the object they wish to select; second, they perform a specific selection action. Thus the kinds of selection listed here and explained below are not separate types of selection. Rather, they are variations of the one selection model theme. Which variation people use depends on whether they wish to select a single object, a range of objects, or several additional (non-contiguous) objects.

Some controls, as selection objects, make specific assumptions about the selection model. For example, a set of radio buttons assumes that each button's selection is mutually exclusive. Thus, while radio buttons follow the selection model for single objects, they do not allow any other type of selection. Check buttons, on the other hand, assume that each button's selection is *not* mutually exclusive. While they also follow the selection model for single objects, they allow the selection of multiple buttons without deselecting the prior selection (as is the case in strict single selection). Check buttons are an example of what is called "multiple selection."

Table 3-3 lists the mouse button and key press operations of the HP OSF/Motif selection model.

TABLE 3-3. The HP OSF/Motif Object-Action Selection Model.

Selection Task	**Mouse Button Operation**	**Keyboard Selection Operation**
Select a single object (set the anchor point) and deselect all other objects.	Click Select	Type Select
Select a range of objects (set the anchor point at range beginning) and deselect all other objects.	Drag Select	Type Shift +navigation keys.
Moves the end point of the range to the current location.	Type Shift +click Select	Type Shift + Select
Toggle the selection of an additional object.	Type Ctrl +selection operation.	Type Ctrl +selection operation.

3.4.1 Single Selection Selects One Object Only

Single selection is probably the most common type of selection. In single selection, people select a single object upon which to perform an action.

In single selection, people move the mouse pointer or location cursor to a selectable object and then make their selection. Selecting an object changes the object's appearance. This provides people with the necessary visual cue to reinforce their sense of control over the selection process. For example, the insertion cursor in a text entry box is emphasized when the box is selected and deemphasized when the box is not selected.

In single selection, when one object is selected, any other objects previously selected are deselected.

People use single selection for such actions as selecting the active window (when the input focus policy is explicit selection) or selecting a push button or other type of control. In a text entry area, people use single selection to position the insertion cursor.

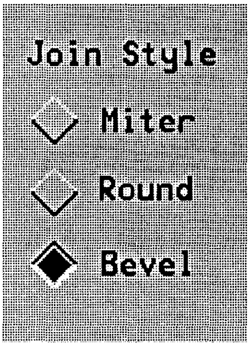


Figure 3-5. A Typical Single Selection Operation.

Single Selection with a Mouse

People working with a mouse perform the following steps to select a single object:

1. Move the mouse pointer until it lies over the object they wish to select.
2. Click the Select mouse button to select the object.

Pressing the Select mouse button changes the object's appearance providing the visual cue to which object is about to be selected. Releasing the button selects the object and completes the single selection process.

Single Selection with a Keyboard

People using a keyboard perform the following steps to select a single object:

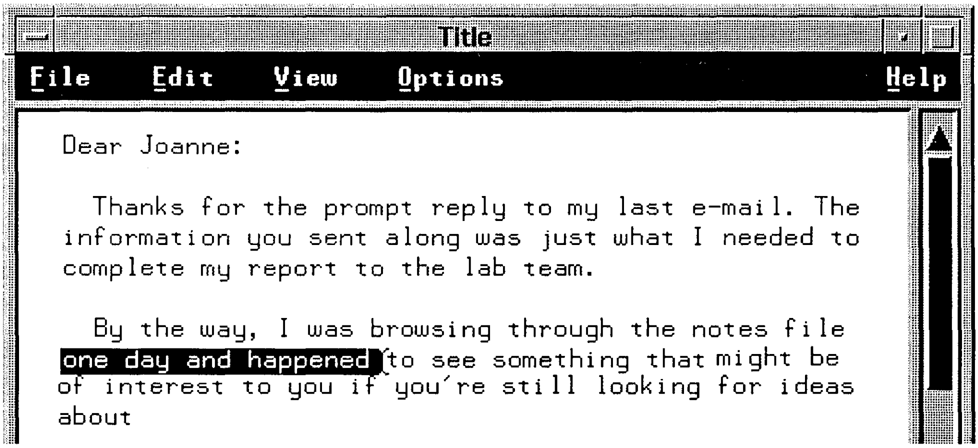
1. Move the location cursor using the cursor navigation keys (eg. Tab or arrow keys) until it lies over the object they wish to select.
2. Type the Select key.

Pressing Select has the same effect as a mouse user's pressing the Select mouse button; it changes the object's appearance and provides the visual cue that the object is about to be selected. When people release the Select key, the object selection is completed.

In an auto-selection area, as people navigate over objects, the objects are automatically selected. Thus, there is no need for a separate selection action.

3.4.2 Range Selection Selects Contiguous Objects

In **range selection**, people select a range of contiguous objects upon which to perform some action. Exactly which objects are included in a range depends on the context of the range selection.



Dear Joanne:

Thanks for the prompt reply to my last e-mail. The information you sent along was just what I needed to complete my report to the lab team.

By the way, I was browsing through the notes file one day and happened to see something that might be of interest to you if you're still looking for ideas about

Figure 3-6. Cutting Text Is a Typical Range Selection Operation.

In list boxes and spreadsheets (two dimensional list boxes), selection occurs as the mouse pointer (or insertion cursor) reaches a selectable object. In text and graphics entry contexts, on the other hand, selection includes all text or graphics objects between the anchor point and the end point. To be included, these objects must be *totally included* in the selection action. The range is based on, but not limited to, a rectangular area.

In range selection, people move the mouse pointer or location cursor to the selectable graphics object and then make their selection. The object's appearance changes to provide a visual cue to the selection. This is the same as for single selection.

In range selection, as the next object in the range is selected, the previous object in the range *remains* selected. Both objects are part of the range being selected. Any number of objects can be selected in range selection as long as they form a continuous, uninterrupted group.

People frequently use range selection during cut-and-paste operations on text. Words are selected as a range of characters and then acted upon (for example, cut or pasted).

Range selection can also be used in graphics applications and in more object-oriented environments. To be included in a range selection, an object, whether text or graphics, must be completely included by the selection operation.

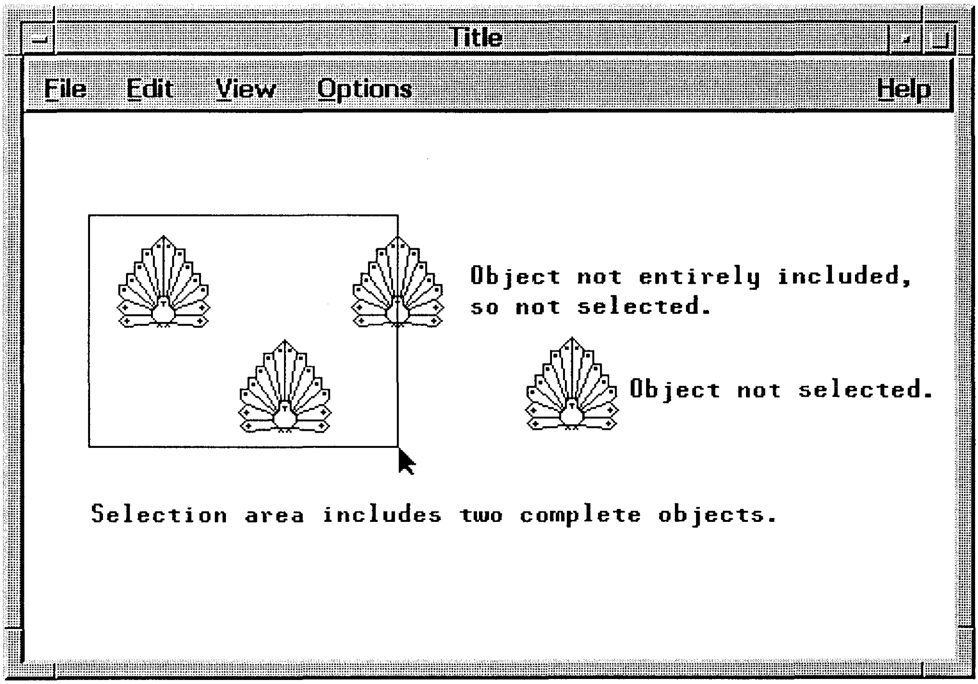

Title
File
Edit
View
Options
Help
Object not entirely included, so not selected.
Object not selected.
Selection area includes two complete objects.

Figure 3-7. In Graphics and Text, Objects Must Be Totally Included.

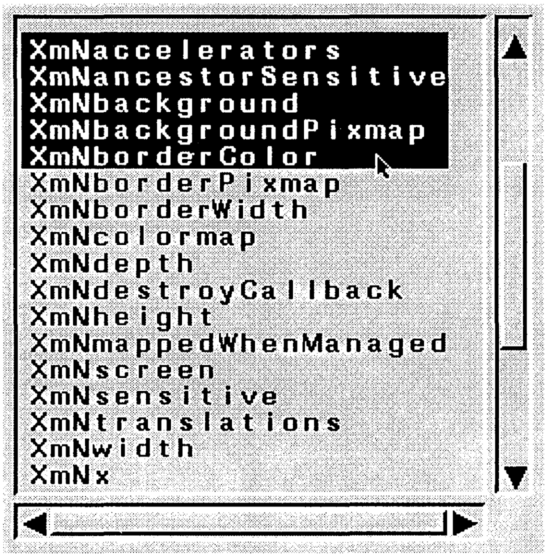
Range Selection Using a Mouse

People using a mouse for range selection perform the following steps:

1. Position the mouse pointer over the object that starts the range.
2. Press the Select mouse button.
3. Drag the mouse pointer past the object that ends the range.
4. Release the Select mouse button to complete the range selection.

Each object's appearance changes providing the visual cue of selection. When the Select mouse button is released, the range selection is complete.

For example, to select five contiguous items from a list box, people position the mouse pointer on the first item, press the Select mouse button, drag the mouse pointer to the fifth item, then release the Select button. The appearance of each selected item changes as the item is selected, providing a visual cue. Releasing the Select mouse button completes the range selection.



XmNaccelerators
XmNancestorSensitive
XmNbackground
XmNbackgroundPixmap
XmNborderColor
XmNborderPixmap
XmNborderWidth
XmNcolormap
XmNdepth
XmNdestroyCallback
XmNheight
XmNmappedWhenManaged
XmNscreen
XmNsensitive
XmNtranslations
XmNwidth
XmNx

Figure 3-8. A Range (Contiguous) Selection Operation.

An alternative to dragging the mouse pointer from the start to the end of the range is to click the Select mouse button on the start, move the mouse pointer to the end of the range, and press Shift +Select to complete the range selection.

Range Selection Using a Keyboard

People using a keyboard for range selection perform the following steps:

1. Position the location cursor (insertion cursor for editable areas) using the navigation keys so that it is over the object that starts the range.
2. Hold down the Shift key and press the navigation keys to drag the cursor past the object that ends the range.
3. Release the Shift key to complete the range selection.

As with range selection using the mouse, each object's appearance changes providing the visual cue of selection. When the Shift key is released, the selection is complete.

For the same example as above, to select five contiguous items from a list box, people position the location cursor on the first item, press the Shift key, type the navigation keys to drag the location cursor to the fifth line, then release the Shift .

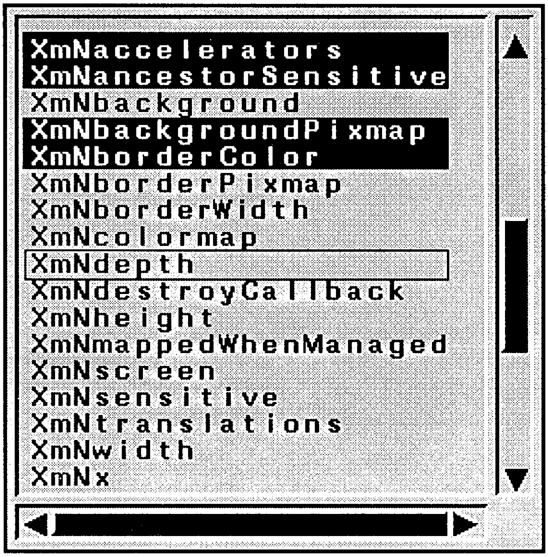
An alternative to holding down the Shift key is to type the Select key to start the selection, type the navigation keys to move the location cursor to the end of the range, and then type Shift + Select to complete the range selection. Note that this doesn't work in auto-select controls such as radio buttons because auto-selection is typically a single selection process.

3.4.3 Additional Selections Select Non-contiguous Objects

People can make one or more additional selections, forming a non-contiguous group of objects.

To begin an additional selection, people select a first object using single selection or a first range of objects using range selection. Other objects are added to the selection group by repositioning the mouse or location cursor and selecting them as non-contiguous objects.

Like single and range selection, objects or ranges that are part of a non-contiguous selection provide a visual cue that they are part of the selection.



XmNaccelerators
XmNancestorSensitive
XmNbackground
XmNbackgroundPixmap
XmNborderColor
XmNborderPixmap
XmNborderWidth
XmNcolormap
XmNdepth
XmNdestroyCallback
XmNheight
XmNmappedWhenManaged
XmNscreen
XmNsensitive
XmNtranslations
XmNwidth
XmNx

Figure 3-9. An Additional (Non-contiguous) Selection Operation.

People find the selection of additional non-contiguous objects, especially the selection of non-contiguous ranges, useful in text processing. Note also that the operation that performs a selection of additional non-contiguous objects works as a "toggle." That is, if the starting object is unselected, the toggle operation selects it (and all other objects in the range). If the object is selected, the toggle operation deselects it (and all other objects in the range).

Making an Additional Non-contiguous Selection Using the Mouse

People using a mouse to make an additional selection perform the following steps:

1. Position the mouse pointer on the next object they wish to select.
2. Press Ctrl +Select to mark the next object or next range in the non-contiguous selection.
3. (For range selection only) Drag the mouse pointer to the end of the range, then release Ctrl +Select.
4. Repeat steps 1-3 for each additional object or range in the non-contiguous selection.

For example, to select several non-contiguous items from a list box, people position the mouse pointer on the first item, click the Select mouse button to select it, move the mouse pointer to the next item for selection, press the Ctrl key and click the Select mouse button to select that item. The appearance of the selected list items changes as the item is selected, providing a visual cue to the selection.

An alternative to the press-drag-release operation is to click Ctrl +Select to begin the selection, reposition the mouse pointer at the end off the selection, then click Shift +Select. This alternate method for selecting multiple ranges is similar to the alternative method for selecting a single range described above.

Making an Additional Non-contiguous Selection Using the Keyboard

People using the keyboard to make additional (non-contiguous) selections perform the following steps:

1. Position the location cursor on the next object they wish to select.
2. Type Ctrl + Select to mark the next object or start of the next range in the non-contiguous selection.
3. (For range selection only) Press Shift and type the navigation keys to drag the location cursor to the end of the range, then release Shift to mark the end of the range.
4. Repeat steps 1-3 for each additional object or range in the non-contiguous selection.

For the same example as above, to select several non-contiguous items from a list box, people position the location cursor on the first item, type the Select key to select it, move the location cursor to the next item for selection, press the Ctrl key and type the Select key to select that item. As with mouse selection, the appearance of the selected list items changes as the item is selected, providing a visual cue to the selection.

As with the selection of a single range, the selection of additional non-contiguous ranges has an alternative method to dragging. This alternative is to type Ctrl + Select to start the range, move the location cursor using the navigation keys to the end of the range, and then type Shift + Select to complete the range selection.

3.4.4 Deselecting an Object

A selection operation can be undone (canceled) by typing the Esc key before the operation is completed.

Once completed, an entire previous selection can be deselected by making a single-object selection.

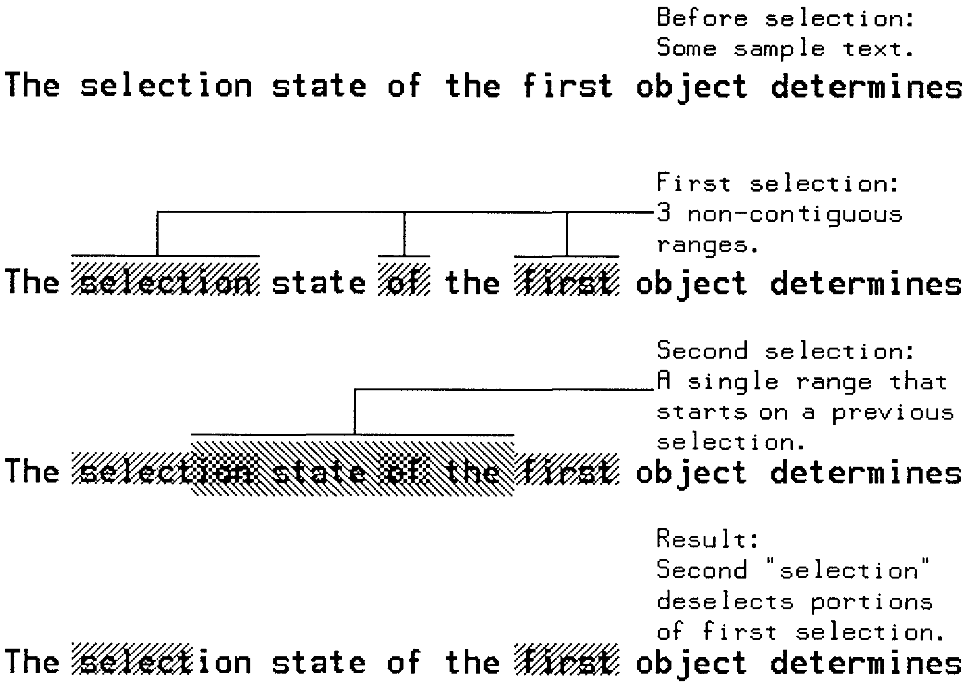

Before selection:
Some sample text.
The selection state of the first object determines
First selection:
3 non-contiguous ranges.
The selection state of the first object determines
Second selection:
A single range that starts on a previous selection.
The selection state of the first object determines
Result:
Second "selection" deselects portions of first selection.
The selection state of the first object determines

Figure 3-10. Toggling Can Be used to Select or Deselect an Object or Range.

As noted above, previously selected object or range can also be deselected by "toggling" its selection state. Usually this requires positioning the mouse pointer or location cursor on the object and then pressing Ctrl and the Select mouse button, for mouse users, or typing Ctrl and the Select key, for keyboard users. This deselection method is commonly used to deselect one object or range in a range of selected objects without deselecting the rest of the range.

Note that, when toggling the selection state of a range, the selection state of the first object determines the final selection state of all objects included in the toggle operation. If the first object in the range was previously selected, the toggle operation will make it and all other objects in the range deselected. If the first object was previously unselected, the toggle operation will make it and all other objects in the range selected.

A previously selected range of objects can be deselected by pressing Ctrl and using a drag operation to toggle their selection state.

3.4.5 Selecting the Default Action

Some objects have default actions associated with them. For example, the default action of an icon is to be restored to its usual window appearance.

For mouse users, double-clicking the mouse with the mouse pointer over an object selects the default operation for that object. For example, double-clicking the Select mouse button with the mouse pointer over an icon restores with the mouse pointer over the window menu button closes the window. Double-clicking the Select mouse button with the mouse pointer in a dialog box performs the default action associated with that dialog box.

For keyboard users, typing the Enter key with the location cursor over an object selects the default operation for that object. For example, typing Enter with the location cursor over an icon, restores the icon to its window appearance. Typing Enter with the location cursor in a dialog box performs the default action associated with that dialog box.

3.4.6 Auto-selection

Auto-selection combines the act of moving the location cursor with the act of selecting the object. In the selection model presented in this chapter, people move the location cursor to an object and then explicitly select the object by pressing the Select mouse button or Select. When auto-selection is employed, people simply move the location cursor to an object and the object is selected; no explicit selection action is required.

Auto-selection provides a convenient shortcut for single-selection controls. Depending on the nature of your application, this may be useful to increase efficiency. However, with auto-selection, navigation is no longer a "harmless" task. For example, navigation through a group box of control settings would change the setting for each control. Thus, you should carefully evaluate the consequences of auto-selection before you implement it.

The HP OSF/Motif Window Manager 4

The HP OSF/Motif user interface provides a rich visual environment to facilitate the communication between people and your application. This environment is composed of discrete graphical elements. It is the functionality provided by these graphical elements that is properly called the **user interface**, the communication between people and application, that is the focal point of this style guide.

The graphical elements of the HP OSF/Motif user interface facilitate communication by providing people with a metaphor, a figurative concept suggestive of real world objects. Through this metaphor, interaction with your application is more familiar (thus more intuitive) and less technical than the traditional user interface provided by the command-line prompt. This metaphor is called variously the "desk," "desktop," "workspace," or "workbench." While desktop is perhaps the more widely known term, this style guide uses the term "workspace" to emphasize that applications in the HP OSF/Motif environment need *not* be "office" oriented and that the functionality and graphical elements of the user interface are *tools* that empower people to get things done with their computers.

The elements of the user interface, the objects that people see (windows, icons, menus, dialog boxes) appear on the workspace and can be stacked one on top of the other like papers on a desk or tools on a workbench.

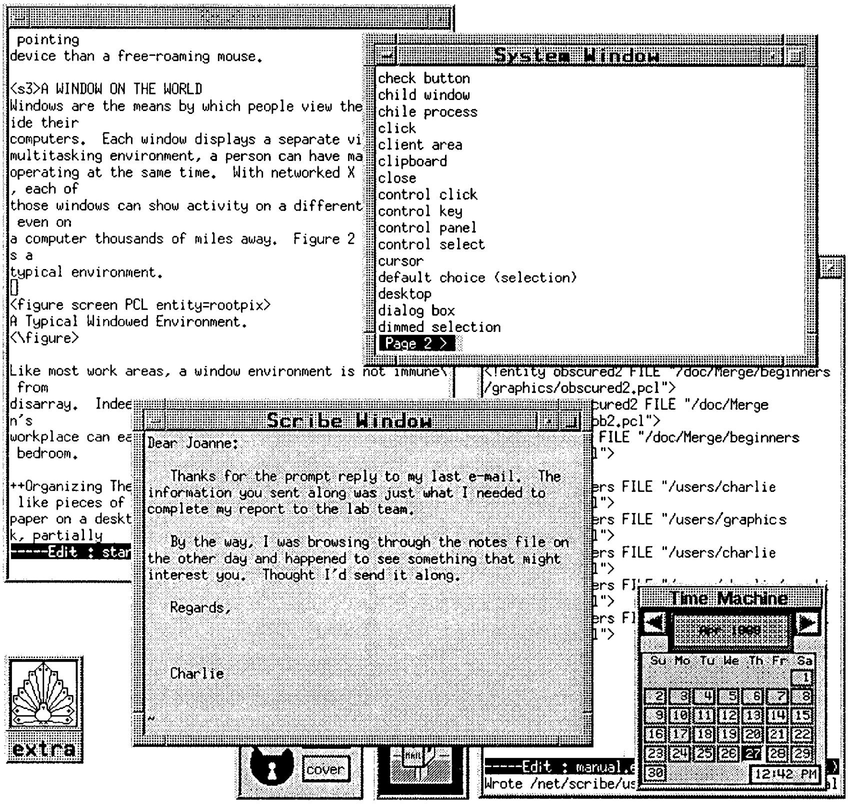
This chapter discusses the following major points about the graphical elements of the HP OSF/Motif workspace:

- The types of windows.
- The anatomy of a window.
- The icon box.

4.1 The Types of Windows

In the HP OSF/Motif environment, people communicate with your application using windows. A **window** is an area of the screen (usually rectangular) that provides people with the functional means to communicate with your application and through which your application can communicate with them.

A typical HP OSF/Motif environment may have several applications in operation simultaneously. Each application typically has a main or "primary" window that displays data and in which people carry on their "primary" interaction with the application. Additionally, applications usually have one or more secondary windows (dialog boxes) that carry on context-specific dialogues with the people using the application. Figure 4-1 illustrates a typical HP OSF/Motif environment. Several elements, including the control buttons on the window frame, have been configured to suit individual preference.



```
 pointing
device than a free-roaming mouse.

<s3>A WINDOW ON THE WORLD
Windows are the means by which people view the
ide their
computers.  Each window displays a separate vi
multitasking environment, a person can have ma
operating at the same time.  With networked X
, each of
those windows can show activity on a different
 even on
a computer thousands of miles away.  Figure 2
s a
typical environment.
[]
<figure screen PCL entity=rootpix>
A Typical Windowed Environment.
<\figure>

Like most work areas, a window environment is not immune\
 from
disarray.  Indee
n's
workplace can ea
 bedroom.

++Organizing The
 like pieces of
paper on a deskt
k, partially
-----Edit : star
```

```
check button
child window
chile process
click
client area
clipboard
close
control click
control key
control panel
control select
cursor
default choice (selection)
desktop
dialog box
dimmed selection
Page 2 >
```

```
<!entity obscured2 FILE "/doc/Merge/beginners
/graphics/obscured2.pcl">
cured2 FILE "/doc/Merge
bb2.pcl">
 FILE "/doc/Merge/beginners
l">

ers FILE "/users/charlie
l">
ers FILE "/users/graphics
l">
ers FILE "/users/charlie
l">
ers FI
l">
ers FI
l">
-----Edit : manual.
Wrote /net/scribe/us
```

Dear Joanne:

Thanks for the prompt reply to my last e-mail. The information you sent along was just what I needed to complete my report to the lab team.

By the way, I was browsing through the notes file on the other day and happened to see something that might interest you. Thought I'd send it along.

Regards,

Charlie

~

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Figure 4-1. A Typical HP OSF/Motif User Environment.

While your application can be made up of many windows, each window will be one of only two basic types:

- A primary window.

- A secondary window.

4.1.1 Primary Windows

A primary window is the window through which all the other windows used by your application are generated. A primary window may be obscured by overlapping windows. Your application can have one *or more* primary windows.

The primary window is the only window through which an application can be closed. That is, when people close the last primary window of an interactive application, the application session should end. Closing the primary window causes all secondary windows associated with that window to go away.

When people invoke your application, the first task of the application is usually to display a primary window. Because of this, people often think of the primary window as the "main" window. This belief provides people with a valuable point of reference: They feel in control knowing there is a main window from which all else follows and to which they can return when they want.

You should design your application to encourage this sense of control. You should design your application so that, as people open and close windows in their dialog with your application, they can always return to the primary window. The primary window should remain consistent in appearance and behavior with the last time they were there.

4.1.2 Secondary Windows

Context-specific dialogs usually occur inside secondary windows called dialog boxes. When the dialog is completed, the secondary window usually disappears.

Secondary windows are always related to a parent window. Sometimes the parent is a primary window, sometimes another secondary window. A primary window can have any number of secondary windows as its children.

Secondary windows are not constrained to be inside the primary window, but they will always appear "on top of" that parent window in the window hierarchy. Thought should be given to how your application distinguishes between primary and secondary windows. One method is to include identifying information in the title area. For application-oriented programs, the application name is followed by the file name in the title bar; for object-oriented programs, the object name is followed by the function. Secondary windows then have title areas that include the application or object name and the implied action.

When a primary window is minimized, its secondary windows are temporarily removed from the display.

4.2 The Anatomy of a Window

The HP OSF/Motif Window Manager (MWM) provides windows with a window frame that contains the components shown in figure 4-2. These components are functional, providing a convenient way for mouse users to invoke window management functions. Keyboard-only users use the window menu provided by MWM to invoke window management functions. Window frame components are sometimes called "decorations."

Along with window frame components, the HP OSF/Motif window layout includes a **client area**, the area inside the window frame, for the use of your application.

In general, a window consists of the following components:

- Window menu button.
- Window control buttons.
- Title area.
- Resize border.
- Client area.

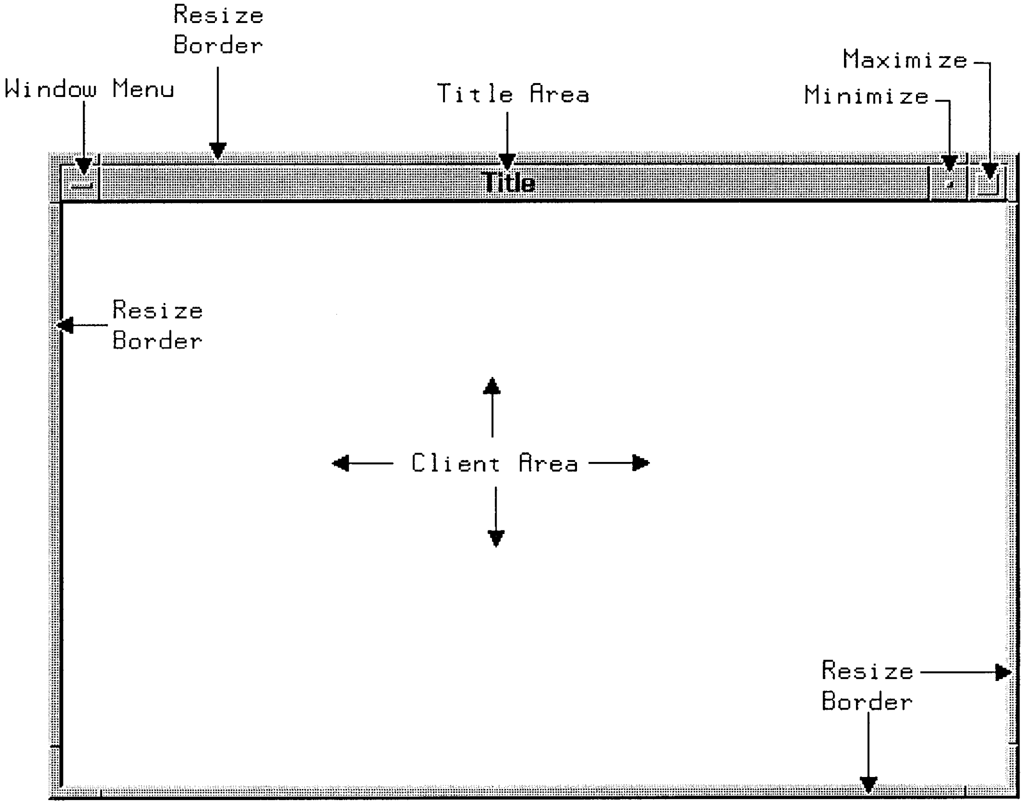


Figure 4-2. An HP OSF/Motif Window Layout.

While figure 4-2 illustrates the components of a typical window layout, specific implementations may require some amount of modification. For example, some windows, ones that it would be inappropriate to resize, could have their resize borders removed.

4.2.1 Window Menu and Window Menu Button

The window menu button is located in the title bar, on the left side of the title area, and is used to display of the window menu. (Also, double-clicking the Select mouse button with the mouse pointer over the window menu button closes the window.) The window menu provides easy access to important window management functions. People can browse the menu to see what actions are available or to refresh their memories. The window menu pulls down from the upper-left corner of the window frame. Mouse users display the window menu by pressing the Select mouse button with the mouse pointer on the window menu button. Keyboard users display the window menu by typing Shift + Esc when the input focus is in the window.

The window menu and window menu button is sometimes called the "system menu" and "system menu button."

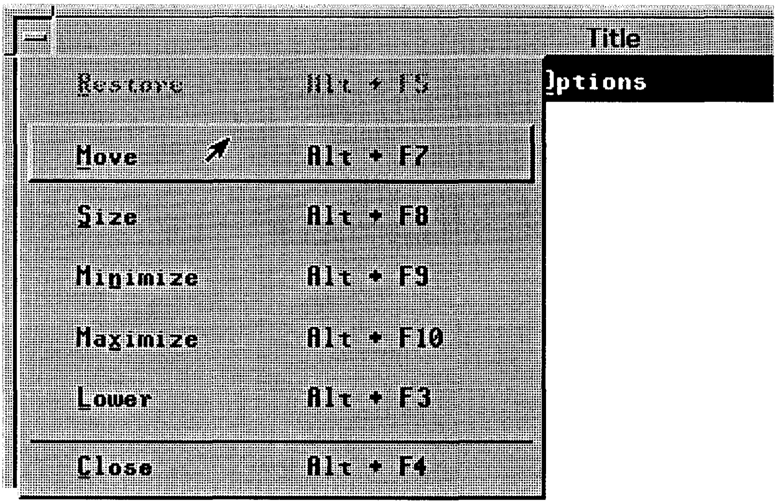


Figure 4-3. The Window Menu Button with Menu Pulled Down.

Figure 4-3 illustrates the common selections of the window menu. In the illustration, the Move selection is being chosen. Also, the Restore function is de-emphasized to provide a visual cue that, in the present context, the function is unavailable. The selections of the window menu have the following functions:

Restore Alt+F5 Restores a minimized or maximized window to its "normal" size. This selection is de-emphasized (grayed out) when the window is in its normal state.

Move Alt+F7 Moves a window around on the workspace.

Size Alt+F8 Stretches or shrinks a window in the direction indicated by the mouse pointer.

Minimize Alt+F9 Changes a window into its icon.

Maximize Alt+F10 Enlarges a window to its maximum specified size.

Lower Alt+F3 (Optional) Moves a window to the "back" of the workspace (the bottom of the window stack).

Close	Alt+F4	Closes a window and removes it from the workspace.

Keyboard accelerators for the window menu are optional, but, if you decide to use accelerators, you should use the accelerators suggested above.

4.2.2 Window Control Buttons

Window control buttons are push buttons located in the upper right of the HP OSF/Motif window frame. They provide a short-cut way of invoking window management functions without pulling down the window menu. People invoke a window management function by clicking on the appropriate window control button.

The functions chosen for window control buttons are implementation dependent. Figure 4-4 illustrates two such functions, Maximize and Minimize.

Minimize Button

The minimize button is located to the immediate right of the title area. It provides the same function as the Minimize selection of the window menu. People click the Select mouse button with the mouse pointer on the minimize button to shrink a window to an icon.

Maximize Button

The maximize button is located between the minimize button and the resize border. It provides the same function as the Maximize selection of the window menu. People click the Select mouse button on the maximize button to enlarge a window to its maximum size. The maximize size of a window is established by the application. Clicking the Select mouse button with the mouse pointer on the maximize button of a maximized window restores the window to its original size, the same function as the Restore selection of the window menu.

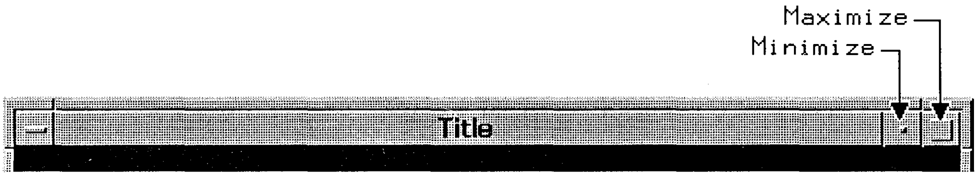


Figure 4-4. Window Control Buttons on the Window Frame.

4.2.3 Title Area

The **title area** is the horizontal bar that lies between the window menu button and the window control buttons, and, as part of the window frame, highlights when the window has the input focus. The title in the title area identifies the window.

Pressing the Select mouse button with the mouse pointer on the title area and dragging the mouse pointer on the screen will move the window to a new location. Clicking the Select mouse button with the mouse pointer on a title area (or frame) raises that window to the top of the window stack.

The window manager displays an **application title** in the title area. The application title is supplied by the application and clearly identifies the window and its role within your application.

You provide the window manager with a title. In object-oriented environments, this title should be the name of the object followed by the application name; in file-based environments, the title could be the name of the application followed by the file name. Multiple windows of the same application should have titles that identify them with the application in some way, but can otherwise be distinct from one another.

The title area should not display the version number of your application. Nor should you use it to display system messages. Use the title area for information that stays relatively constant throughout the work session of your application.

4.2.4 Resize Border

Your application suggests the initial size of its windows to the window manager. Window sizes vary according to the work people perform in them. At any time, people should be able to alter the size of the primary window.

The window manager provides a functional window frame which surrounds the client area. The window frame highlights when input focus is passed to the window. Resizable windows have a wide frame border that people can drag when they want to change the window's size.

4.2.5 Client Area

The **client area** is the portion of the window in which people perform most application-level tasks. For example, if people are working with a graphics editor or a text editor, the client area contains the figure or document being edited. The client area is inside the window frame and can be composed of multiple work areas. Figure 4-5 shows a window with a hypothetical client area.

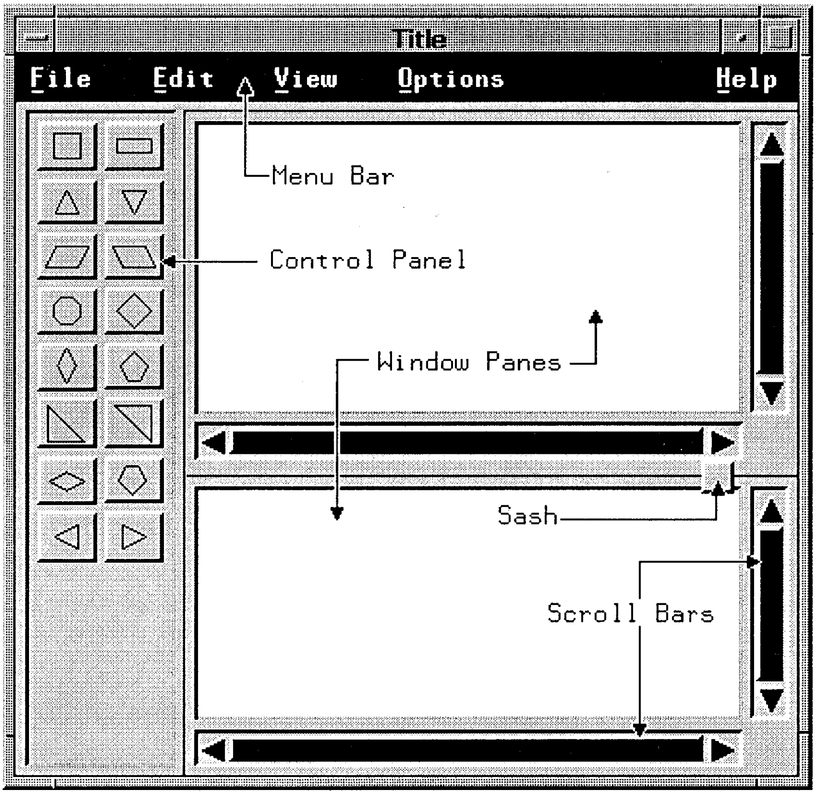


Figure 4-5. The Anatomy of A Typical HP OSF/Motif Application.

4.3 The Icon Box

By default, minimized windows are placed on the workspace in a row beginning at the lower left corner of the screen. However, under the management of the HP OSF/Motif Window Manager, people can choose to group minimized windows in an icon box. They can also choose to have minimized windows placed at the location the normal window occupied.

An icon box acts as a typical window in the sense that it has a window frame and frame controls. Like other windows it can be sized, moved, minimized, maximized, restored, and lowered. However, an icon box cannot be closed.

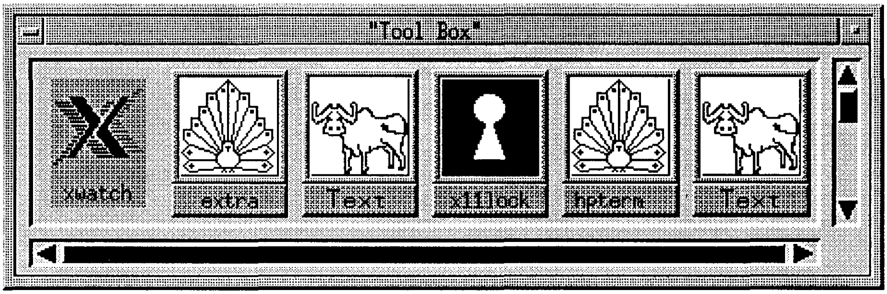


Figure 4-6. A Typical HP OSF/Motif Icon Box.

Client Area Design 5

The HP OSF/Motif Window Manager is responsible for providing window management services for the windows of all applications in the HP OSF/Motif environment. Your application is responsible for organizing the client area of the main window, any subareas, and any secondary windows.

This chapter discusses the following client area design topics:

- Client areas.
- Grouping like controls.
- Presenting multiple controls.
- Laying out an application's areas.

The chapters following this one discuss the use of controls, menus, and dialog boxes in more detail.

5.1 Client Areas

Organizing client areas is an important part of your design process. Depending on the nature of your application, you may choose to divide the client area into one or more subareas. Additionally, you may choose to design your application with secondary windows (dialog boxes). Subareas and secondary windows visually reinforce the organization of your application and increase peoples' sense of control over its operation.

5.1.1 Client Subareas

As mentioned, you can divide the client area of your application window into one or more subareas. Client subareas are very application specific, with the possible exception of the menu bar.

Figure 5-1 illustrates a hypothetical HP OSF/Motif main application window using some common subareas.

Control panels

Some applications benefit by organizing part of the client area into a subarea called a **control panel.** A control panel is a group of like controls having similar functions. Control panels can either be a part of the client area, if their use is required frequently, or be a part of a dialog box, if their use is required occasionally.

Control panels are made up of the controls discussed in chapter 6.

Menu bar

The **menu bar** is the horizontal bar that appears just below the title area. It contains a list of menu topics from which people can select. A single letter **mnemonic** for each menu topic is underlined.

Keyboard users select a topic by typing F10 to move the location cursor to the menu bar and then typing the mnemonic letter of the topic. Mouse users select a topic by positioning the mouse pointer over that topic and selecting it with the Select mouse button. Selecting a topic causes a pull-down menu to display selectable items related to that topic.

Note that commands are not included as "topics" in the menu bar because they would prohibit people from browsing the menu topics.

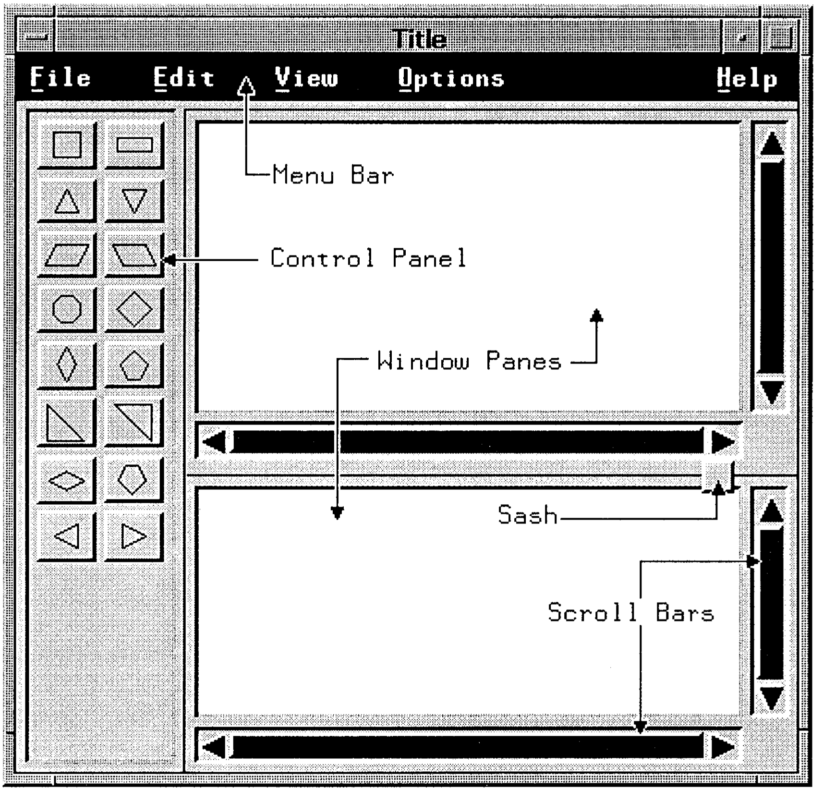


Figure 5-1. A Hypothetical Window with Subareas.

Because menus are a principal method of interaction between people and HP OSF/Motif applications, most applications require a menu bar. Refer to chapter 7 for more detailed information about menus.

Other Client Subareas

In addition to a menu bar and control panels, your application design might call for other types of client subareas.

Message Area You may decide that it is more efficient for people to view messages within a subarea of the client area rather than in a separate message box. If so, the messages should appear at the bottom of the client area so that messages aren't obscured by pull-down menus. The message can be either in a line reserved for messages or in a line temporarily used for the message and then returned to its previous use. Warnings and messages that require immediate action are not displayed this way; but rather are always displayed in message boxes to give them greater visibility.

Command Line Although the HP OSF/Motif environment espouses a graphical-oriented, object-action selection model, your particular application may permit the use of typed commands to enhance the control people have over your application.

Command lines should generally run from border to border across the bottom of the client area just below the message area.

5.1.2 Client Controls

Each subarea can contain a variety of controls enabling people to manage that subarea. Like subareas, the controls chosen for a particular subarea, are very application specific. Which controls you choose depend on the needs of the people who will use your application and your application design.

Window panes

Depending on the needs of your application and the people who use it, you may decide that it is better to divide your client area into **window panes** rather than fixed-partition subareas.

Panes can be either vertical (one on top of the other) or horizontal (side by side). People can resize panes by dragging the boundary between the panes. Making one pane bigger, makes the other pane smaller, while the overall size of the window remains the same.

Scroll bars

People use scroll bars to scan rapidly through the contents of a window. The current location of the scroll bar is shown by the position of the slider in the scroll bar area.

Resizable windows and window panes require scroll bars if the information in them will become too big for the area and become obscured by the border of the subarea.

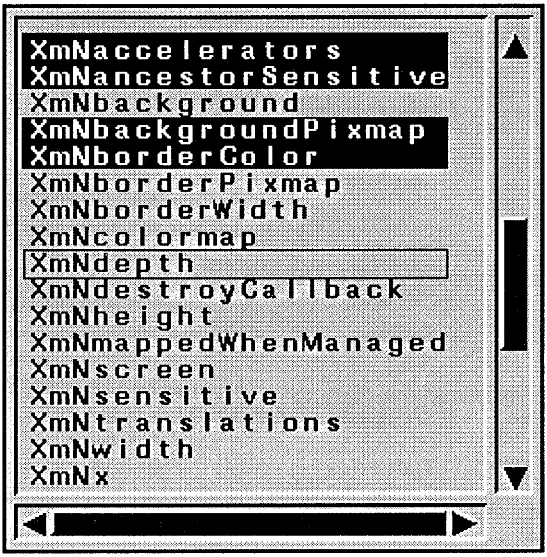


Figure 5-2. Scroll Bars Can Be Either Vertical or Horizontal.

Scroll bars are located to the right or on the bottom of the area to be scrolled. Scroll bars can also be used in dialog boxes or in combination with other controls. Their operation is discussed in chapter 6.

Other Controls

The HP OSF/Motif object-action selection model uses a number of other **controls**. These controls are graphical representations of real life controls and include the following:

- Push buttons.
- Radio buttons.
- Check buttons.
- List boxes.
- Entry boxes.
- Scales.
- Scroll bars.

Controls are discussed in chapter 6.

5.1.3 Other Client Areas

In addition to organizing your application's client area into subareas, you may choose to organize your application and its operation using some of the other methods provided by the HP OSF/Motif environment. These include pop-up menus, cascading menus, dialog boxes (secondary windows). Like subareas, these other areas visually reinforce the organization of your application and increase peoples' sense of control over its operation.

Menus

HP OSF/Motif menus work just like real life menus. Menus enable people to choose from a list of possible selections. Besides the pull-down menus on the menu bar, the HP OSF/Motif environment has the following types of menus:

Pop ups — Menus that "pop-up from nowhere" rather than being related to a menu bar.

Cascading — Menus that cascade to the right and down from either pull-down or pop-up menus and provide a subsidiary level of selection.

Option — Menus that display from a dialog box, usually by pressing a push button.

Menus are discussed in chapter 7.

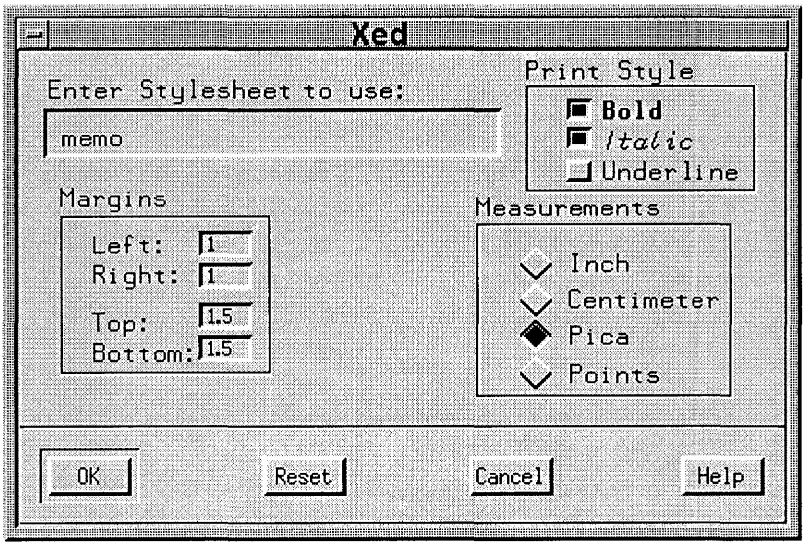
Dialog Boxes

Controls that are not frequently in use during the operation of your application can be included in a **dialog box**. A dialog box is a separate window from the application's main window and contains controls that should be readily accessible but that people don't need to have displayed permanently in the client area.

Dialog boxes are discussed in chapter 8.

5.2 Grouping Similar Controls

Some controls perform similar functions or are logically related. For ease of use, as well as for proper visual design, these controls should be grouped into sets. This keeps the user interface of your application organized. Similar or related controls can be placed either in **group boxes** or in window subareas. Typically, group boxes occur in dialog boxes and are often surrounded by a simple frame. Window subareas often occur as part of an application's main window and contain frequently used controls that must remain readily available.



Enter Stylesheet to use: memo

Print Style
☒ Bold
☒ *Italic*
□ Underline

Margins
Left: 1
Right: 1
Top: 1.5
Bottom: 1.5

Measurements
○ Inch
○ Centimeter
● Pica
○ Points

Figure 5-3. Related Controls Can Be Grouped into Sets.

Grouped controls provide a visual cue that the controls are related to each other by isolating them from other controls. A group box or control panel usually has a title printed near it. Controls can be grouped in one row or column, or multiple rows or columns.

5.2.1 Designing Grouped Controls with Push Buttons

Grouped controls that contain push buttons should adhere to the following guidelines:

- Use push buttons sparingly.
- Use push buttons only for frequently used commands.

Unless you are trying to duplicate the physical appearance of an existing piece of equipment, place push buttons in dialog boxes in a row along the bottom of the box or in a column along the right side of the box. This is because most push buttons correspond to actions and the action is the last selection people make before leaving the dialog box.

5.2.2 Combining Controls

The HP OSF/Motif environment does not restrict the combination of controls to only those combinations mentioned in this style guide. The criteria for developing control combinations should always be consistent: will this combination empower the people who use it to work more efficiently and, as a result, become more productive.

5.3 Presenting Multiple Controls

The HP OSF/Motif environment uses four basic ways of displaying multiple controls:

- Pull-down menus.
- Pop-up menus.
- Dialog boxes.
- Window Subareas.

5.3.1 Pull-Down Menus

Pull-down menus usually contain push buttons, radio buttons, or check buttons as selection items. Selections can lead to dialog boxes or other controls. Menu items are always presented in a vertical column. To display a pull-down menu requires some degree of "mouse travel" or the use of an Alt +*mnemonic* accelerator. While the selections of a pull-down menu do not appear until the menu is selected, the title of a pull-down menu is always displayed in the menu bar. Pull-down menus combine a visual cue of their presence with an efficient use of space.

5.3.2 Pop-Up Menus

Pop-up menus, like pull-down menus, usually contain push buttons, radio buttons, or check buttons, and can have selections that lead to dialog boxes or other controls. Also like pull-down menus, pop-up menus are always presented as a vertical column. Pop-up menus are associated with a particular area of the screen. The advantage of pop-up menus is that they require no mouse travel; they simply pop-up at the current mouse location (provided that location has a menu associated with it). While pop-up menus take up no screen space until they are displayed, they provide no visual cue to their existence and their presence must be memorized because it can't be intuited.

5.3.3 Dialog Boxes

Dialog boxes can contain all the button, box, and valuator controls. Usually, when the dialog box is displayed, all necessary controls are present. However, your application may require some "extra" display operations that should be in the dialog box but need not be displayed all the time. If so, you can include an option menu push button in your dialog box.

Dialog boxes allow a large degree of flexibility in the arrangement of controls. Controls can be grouped in boxes, organized in rows or columns, and separated by white space for better visibility.

Message boxes are usually displayed by the application program without any explicit action from people using the program. Dialog boxes, on the other hand, are usually displayed as a result of some explicit action. People can display dialog boxes directly using a keyboard accelerator. This saves the time and extra steps required when selecting them from a menu. Like pop-up menus, dialog boxes use space efficiently because they are not visible until displayed. However, dialog boxes (unlike pop-up menus) provide a visual cue to their existence, the button used to display them. Dialog boxes are removed from the workspace when their primary window is minimized and returned to the workspace when their primary window is restored.

5.3.4 Control Panels

Control panels, like dialog boxes, can contain all the button, box, and valuator controls. They also offer the same flexibility of arrangement. Control panels, since they are permanently displayed, offer the potential of having frequently used controls always available. Being displayed, they offer a strong visual cue to their presence, but they do take up screen space.

5.4 Laying Out a Client Area

The nature of your application may be such that you will need to design control panels or dialog boxes specific to your situation. When doing so, it is important for the HP OSF/Motif-conformity of your application to use the following criteria:

5.4.1 Lay Out Controls in Natural Scanning Order

Design the layout of your application windows according to the natural scanning order of the people who will be using your application. In many cases, this order will be from left to right and from top to bottom.

5.4.2 Arrange Controls in the Sequence People Use Them

Intimately connected to the use of natural scanning order in control layout is the arrangement of controls in the sequence in which people will use them. The natural scanning order gives the position; the sequence of use gives the priority.

For example, suppose you have a dialog box that has push buttons that accept changes, test them, restore original values, and cancel the dialog box without making changes. The natural scanning order in western societies dictates that the push buttons be laid out from left to right. The sequence of use dictates that the OK button should probably be on the left as the most frequently used button, followed, in decreasing order by Apply, Reset, and Cancel.

5.4.3 Let People Adjust the Client Area

To increase the control people have over your application, your application should allow people to adjust the client area to fit their needs.

If your application uses window panes, it should allow people to adjust the size of the panes to suit their needs by repositioning the **sash**, the border separating the two panes. When one pane increases in size, the other pane decreases by the same amount. The overall sizes of the window frame and the client area *do not* change as the panes are resized.

People should be able to adjust the sash using either a mouse or a keyboard operation. Moving the sash with the mouse is typically a button-press, drag, button-release operation like moving a window using the title area. Moving the sash with the keyboard typically requires a selection and the use of the navigation keys.

5.4.4 Choose the Appropriate Control

Radio buttons, option menus, and list boxes can all be used to choose one option from a list of multiple options. Choosing the right control for the job depends on the number and nature of the options in the list.

For choosing a single option from among a small number of mutually exclusive options, a radio button is usually the easiest for people to operate. For more options, an option menu push button takes up a small amount of space and is relatively easy to use. For many options, the list box is the easiest for people to use; also, more than one item can be chosen at a time.

5.4.5 Decide Between a Pop-Up Menu and Push Buttons

Pop-up menus provide people with quick access to application functions. So do control panels containing push buttons. How do you decide which is more appropriate for your application?

Generally, pop-up menus are preferable when people are focused on their work areas. In these situations, moving the mouse between a control panel and the work area would be distracting.

Push buttons and a control panel are preferable when people make frequent selections, need to make several selections at the same time, or are already manipulating the mouse primarily in the control panel area.

5.4.6 Decide Between Dialog Boxes and Menus

You should design your application so that it is consistent with other HP OSF/Motif applications. To do so, you should understand when to use dialog boxes and when to use some other method of control. In particular, you should know the difference between dialog boxes and menus.

As you design your application, you will encounter many instances in which the same objective can be accomplished with either a dialog box or a menu. The menu selections act similar to the controls used in most dialog boxes. However, there are differences.

A menu is short-lived. It appears quickly, but exists only while a selection is being chosen. As soon as the selection is made, the menu disappears. A dialog box, on the other hand, can stay displayed until told to go away, but usually takes up more workspace. While the dialog box is displayed, people can make several different selections.

Additionally, a menu is usually modal in nature. Until a menu goes away, people can't interact with any other part of the application. Dialog boxes, on the other hand, are frequently modeless. People can still interact with other parts of the application while the dialog box is displayed.

Thus, if a modeless state were required, a modeless dialog box would be the appropriate solution. In the case of people browsing current settings or making a single selection, a menu would be faster. However, in the case of making several selections, a menu would keep disappearing after each selection and would have to be continually redisplayed. Thus a dialog box would be a better design choice if several selections were usually made at the same time.

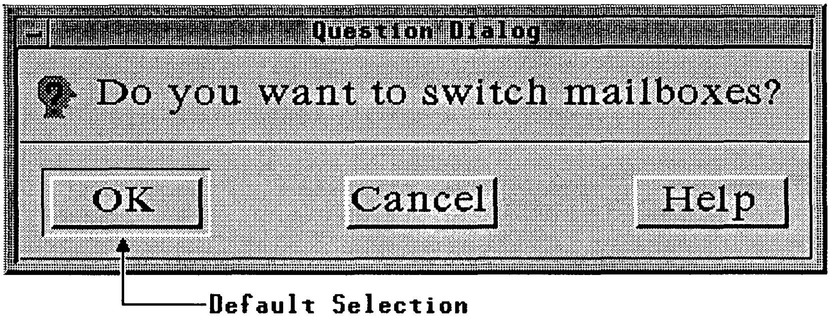
5.4.7 Align Columns of Controls

While push buttons are usually placed in a row along the bottom of the dialog box, check buttons and radio buttons are frequently placed in columns. When using columns, align the check buttons or radio buttons vertically so that the location cursor doesn't "bounce around" as people tab through the selections. Proper vertical alignment also enables people to slide the mouse pointer in a single direction rather than having to zigzag through a slalom course of misaligned selections.

5.4.8 Making Life Easier with Defaults

To make life easier for people, your application should use default values for common settings or obvious selections. A default selection should be visually distinguishable from other selections. Typically this is done by surrounding the default selection with line.

Default check buttons need only display with a check in them. The default selection in a set of radio buttons also displays selected. A default push button is typically indicated by a double border. People activate the default push button by double-clicking the Select mouse button with the mouse pointer in the area containing the push button or by pressing Enter on the keyboard. Figure 5-4 shows a default push button.



Do you want to switch mailboxes?

Default Selection

Figure 5-4. A Typical Default Push Button.

Controls 6

People control applications in the HP OSF/Motif environment using a number of graphical controls. These controls are of the following three types:

Buttons — Like the control buttons in real life, people generally operate HP OSF/Motif graphical control buttons by "pressing" them with the Select mouse button or Select key.

Boxes — Like boxes in real life, graphical control boxes generally contain things of interest to people.

Valuators — Like some analog gauges in real life, valuators provide people with a way to specify or control incremental changes.

6.1 Types of Buttons

HP OSF/Motif-consistent applications currently use three types of buttons: push buttons, radio buttons, and check buttons. Which button you use for your application depends on the situation you wish to control. Buttons should be of a size large enough so that people can easily position the mouse pointer on them.

6.1.1 Push Buttons

A **push button** consists of two parts:

- A graphical image that represents the button.
- A label or icon describing the action invoked by the button.

When people position the mouse pointer anywhere on a push button and click the Select mouse button, or position the location cursor on the push button and type Select, the action represented by the push button occurs.

The label of a push button should be short, usually a verb for action push buttons, such as Cancel or Apply. Response push buttons can have text such as OK or Yes, however, the question that prompts the response should be carefully worded to avoid ambiguity.

A push button can be used to display another dialog box. This is the case with an option menu in a dialog box. A push button used this way should provide a visual cue to its functionality by following its button label with an ellipsis (. . .).

6.1.2 Radio Buttons

A **radio button** consists of two parts:

- The graphic image that visually represents the button.
- A label that describes the choice represented by the graphic image.

Each radio button represents a single-choice selection. Radio buttons are always in a fixed set of at least two buttons and always represent *mutually exclusive choices*.

Conceptually, radio buttons work like the buttons on a car radio (from which they derive their name). People select or unselect a radio button by clicking the Select mouse button when the mouse pointer is over the radio button or by pressing Select when the location cursor is over the radio button. Like a car radio, when one radio button is selected, the previously selected button is unselected.

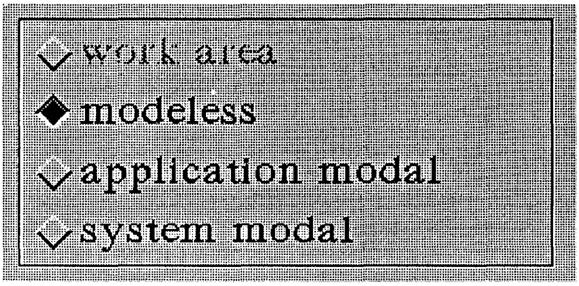


Figure 6-1. A Typical Set of Radio Buttons.

Radio buttons that refer to similar kinds of options should be grouped in sets. Sets of radio buttons that refer to different kinds of options should be grouped separately. Sets can be arranged in either rows or columns. Use white space to visually separate multiple sets of radio buttons into a control panel.

Radio buttons are usually circles in 2-dimensional environments and usually diamonds in 3-dimensional environments to distinguished them visually from check buttons which are usually square.

6.1.3 Check Buttons

A **check button** consists of the following two parts:

- A square box or "button" that is empty when unselected but that is filled in or contains some other visual cue when selected.
- A label that identifies the purpose of the check button. The label is at the same level as and to the right of the check button.

Check buttons enable people to select choices that are *not mutually exclusive*. People select or unselect a check button by clicking the Select mouse button when the mouse pointer is on the check button or by pressing [Select] when the location cursor is over the check button.

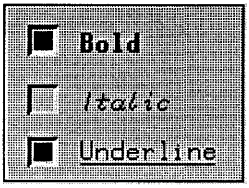


Figure 6-2. A Typical Set of Check Buttons.

6.2 Types of Boxes

HP OSF/Motif-consistent applications currently use two types of control boxes: list boxes and entry boxes. Which box you use depends on the situation.

6.2.1 List Boxes

A **list box** typically consists of the following parts:

- A title that describes the purpose or contents of the list box. The title generally appears above the list box.
- A window containing the listings.
- Vertical and horizontal scroll bars, as needed. The scroll bars enable people to view the listings.

List boxes enable people to select from an existing list of items that is either long or variable in length.

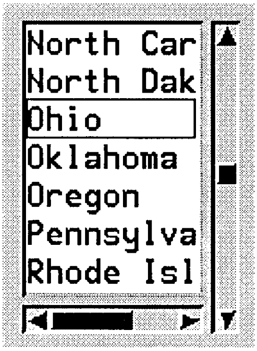


Figure 6-3. A Typical List Box.

People move the slider on the scroll bar to change their current view of the list. To choose a selection, people position the mouse pointer on the selection and click the Select mouse button or position the location cursor on the selection and type Select . The usual methods for range and additional non-contiguous selections are also supported.

List boxes do not support mnemonics. Instead, they have a speed-search function that works as follows: When people type the first letter of an item in a list box, the box scrolls to the first occurrence of an item that begins with that letter. For example, if the list box contained an alphabetical listing of the states in the United States, a person would type "O" to view the states beginning with Ohio (see figure 6-3).

Double-clicking the Select mouse button in a list box chooses a selection and activates the default push button in the dialog box.

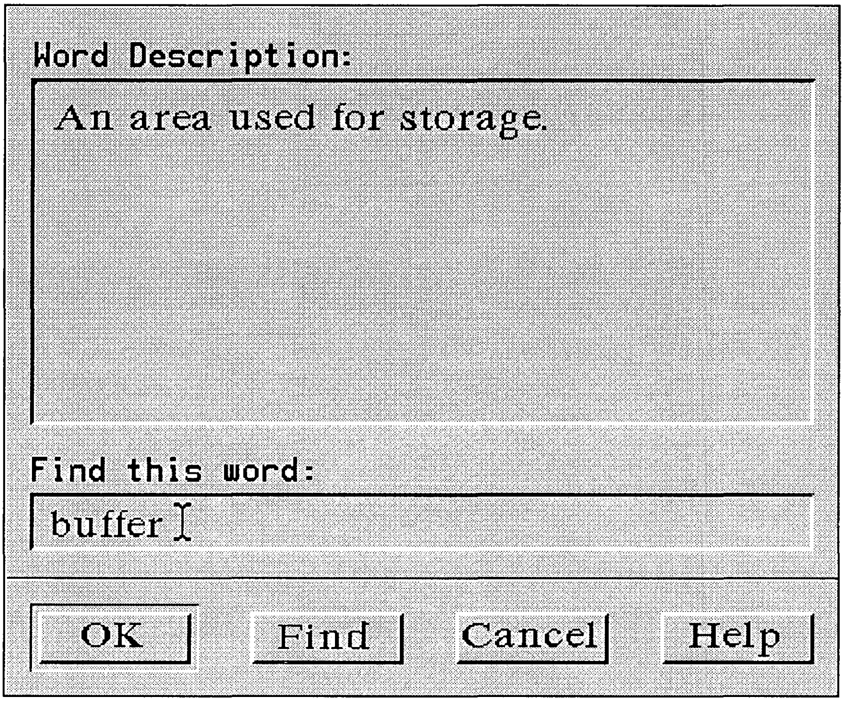
6.2.2 Entry Boxes

An **entry box** consists of the following parts:

- A title or label.
- The box in which text is entered.

Entry boxes enable people to enter text. The entry box may scroll horizontally if the text entered is longer than the box. The entry box may also be more than one line high, in which case it can have a vertical scroll bar like a list box.

The title describes what is to be entered in the entry box. Titles generally appear above or to the left of the box.



Word Description:

An area used for storage.

Find this word:

buffer

Figure 6-4. A Typical Entry Box.

When people move the mouse pointer into an entry box, the mouse pointer changes from the default shape to the shape of the text insertion cursor.

Text Editing Functions

Entry boxes follow the rules for basic text editing including the use of the following common editing functions:

Insert Mode — In insert mode, the character typed on the keyboard is inserted at the location of the insertion cursor. Text that follows the cursor is shifted to the right.

Replace Mode — In replace mode, the character at the location of the insertion cursor is removed and the character typed on the keyboard is inserted in its place.

Delete — Removes the character to the right of the insertion cursor. Text following the cursor is shifted to the left.

Backspace Removes the character to the left of the insertion cursor. Text following the cursor is shifted to the left.

Pending Delete

Entry boxes include a function known as "pending delete." Using this function, people can select a range of text to be overwritten in an entry box and then simply begin typing the replacement text. The selected text is deleted and the new text is inserted in its place as typed.

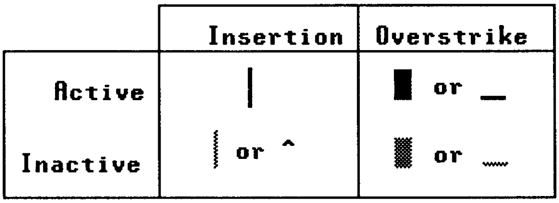
Text Cursor Shapes

A text insertion cursor shows where text will be inserted or overstruck. Insertion cursors should always provide people with a visual cue to the current text mode, insert or overstrike.

In insert mode, the cursor typically appears as a vertical bar or pipe (|) between two text characters. When people type a new character, the character appears to the left of the cursor. The cursor moves one character space to the right. In an inactive window or entry box the insertion cursor is de-emphasized.

In overstrike mode, the cursor typically appears as a block or underline located at the text character that will be replaced. When people type a new character, the cursor replaces the existing character with the new character and moves to the right to the next character. In an inactive window or entry box the block is de-emphasized.

Figure 6-5 shows typical text cursor shapes.



	Insertion	Overstrike
Active	|	█ or _
Inactive	┆ or ^	▒ or ﹏

Figure 6-5. Typical Text Cursor Shapes.

Your application should allow people to control whether insertion cursor blinks or not.

While several insertion cursors may appear on the workspace, only *one* cursor, the one in the window with the input focus, can be active; all other insertion cursors are inactive and have a de-emphasized shape.

An insertion cursor can change size and should be set to the height of the current font.

6.2.3 Pre-formatting Entry Areas

Where possible, the entry boxes of your application should be pre-formatted. Typical instances are entry boxes for supplying phone numbers or social security numbers. Pre-formatted entry boxes increase the uniformity of the data entry ("Garbage in–garbage out") while easing the burden of remembering and correctly typing formats.

When the text entered in an entry box is all the same length (for example, phone numbers or social security numbers), you can implement **auto tabbing**. Auto tabbing speeds data entry in fixed-length fields by automatically moving the cursor to the next field as people finish making an entry in the current field. This saves moving the mouse or pressing the navigation keys.

6.3 Types of Valuators

The HP OSF/Motif toolkit includes several types of valuators that enable you to provide people with analog-style controls.

6.3.1 Using a Scale

Your application could use a **scale** valuator. A scale enables people to enter a value from a range of values by adjusting in analog fashion a sliding arrow to a specific position along a line.

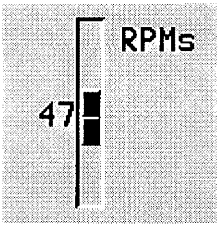


Figure 6-6. A Typical Scale.

A scale consists of the following components:

Scale bar — The scale bar may contain tick marks and represents the range of available values.

Slider — The slide arrow marks the currently chosen scale value.

Digital readout — The digital readout is an optional number directly opposite the slider that is the digital representation of the currently chosen analog scale value.

6.3.2 Using a Scroll Bar

People use scroll bars to scan rapidly through the contents of a window or to choose from a continuously variable set of values such as color intensity. The current location or setting of the scroll bar is shown by the position of the slider in the scroll bar for text windows or by example for variable choices such as color intensity.

If people try to scroll beyond the end of the text, nothing should happen.

Scroll Bar Components

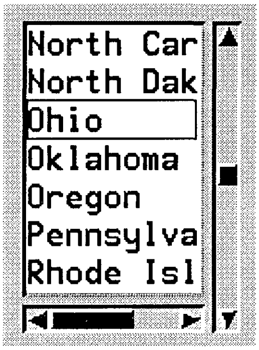
Scroll bars have the following components:

Scroll region — The scroll region is the "background" of the scroll bar and represents visually the length of the area that people can scroll.

Slider — The slider represents the window through which people look at the displayed data. Put another way, the position of the slider box on the scroll region provides a visual cue that marks the location of people's viewpoint in relation to the total scrollable area.

Stepper arrows — The stepper arrows enable people to scroll incrementally through the data and provide a visual cue to the direction of the scrolling movement.

Your application can use either horizontal or vertical scroll bars or both. The slider moves back and forth in the scroll region showing the position of the currently displayed section relative to the entire contents.



North Car
North Dak
Ohio
Oklahoma
Oregon
Pennsylva
Rhode Isl

Figure 6-7. Scroll Bars Can Be Either Vertical or Horizontal.

Operating Scroll Bars

Viewing text or graphical information through a window is like viewing the stars through binoculars. To change the view of the sky, people move the binoculars, not the stars. When the binoculars move up, the stars appear to move down; whichever direction the binoculars move, the stars appear to move in the opposite direction. Similarly, when people use a scroll bar to view a file, the file appears to move in the direction opposite to the movement of the slider. For example, in a text window, if the slider of a vertical scroll bar moves up, a text display seems to move down as previous lines in the file appear at the top of the window.

Table 8-1 discusses the different ways people can operate a scroll bar.

TABLE 6-1. Operating Scroll Bars.

Action	Description
Clicking the Select mouse button with the pointer on a stepper arrow.	Highlights the stepper arrow and moves the window through the underlying file *by a single unit* in the direction of the arrow.
Pressing the Select mouse button with the pointer on a stepper arrow.	Highlights the stepper arrow and *causes a continuous scroll*, in unit steps, in the direction indicated by the arrow.
Pressing Shift and clicking the Select mouse button with the pointer on a stepper arrow.	Highlights the stepper arrow and moves the window to the top or bottom of the underlying file depending on the direction of the arrow.
Clicking the Select mouse button with the pointer on the scroll region.	Moves the window through the underlying file by one window length minus one unit for overlap.
Pressing the Select mouse button with the pointer on the scroll region.	Continuously moves the window through the underlying file by one window length minus one unit for overlap.
Pressing the Select mouse button and dragging the slider.	Moves the slider and continuously moves the window to a location consistent with the new slider location.

Automatic Scrolling

When people drag the mouse pointer (with a mouse button pressed) beyond the top or bottom of the window, your application should continue the selection by scrolling in the direction of the mouse pointer. This automatic scrolling operates at the same speed as when people press on the directional arrows. The automatic scrolling ends when people move the mouse pointer back into the window or release the Select mouse button.

Slider Size

The slider itself may vary in size to represent the proportion of the entire contents currently covered by the window.

Scroll bars are usually located to the right or on the bottom of the area to be scrolled.

6.3.3 Application Extras

Besides the controls supplied by the HP OSF/Motif toolkit, other dialog box controls can be supplied by the application as needed. One such application extra is the stepper button. A **stepper button** typically consists of the following parts:

- A title or label.
- A group box or panel separating the stepper button visually from the other parts of the dialog box.
- A text box displaying the current value of the stepper button.
- A scroll bar for "stepping" through the list of values to find a new value.

A stepper button enables people to select a value by scrolling through a "circular" list of possible values. The stepper button is similar in operation to the digital read-out of a stereo tuner where pressing the button "steps" the read-out through the available radio stations.

The values that "read out" as people use the stepper button can be either letters or numbers, but they should be in consecutive order, either alphabetical or numerical, as opposed to random order. People should be able to anticipate the appearance of values.

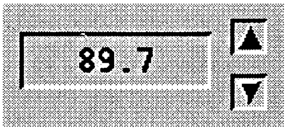
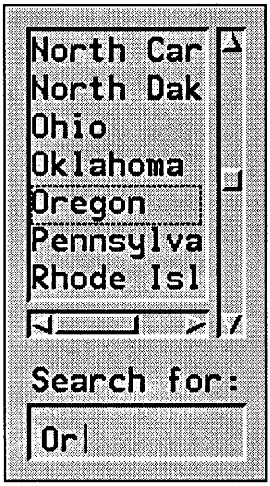


Figure 6-8. A Typical Stepper Button.

6.4 Combining Controls

Entry boxes and list boxes are often used in combination to provide people with particular capabilities. Some common examples are the following:



Search for:

Or

Figure 6-9. A Combination List and Entry Box for Incremental Searching.

incremental searching

As people type each letter of an entry in the entry box, the list box moves to the part of the list that matches what has been typed so far.

For example, if the list box contained an alphabetical listing of the states in the United States, a person would type "O" to view the states beginning with Ohio. If the person then typed "R," the list would move to "Oregon." When the desired item appears in the list box, people click the Select mouse button to select the item. The list box can still be used in the normal way.

automatic entry

People scan the items in the list box and click the Select mouse button to put the current item in the entry box. The entry box can still be used in the normal way.

List boxes used in incremental searches or in automatic entry must be single selection.

Menus 7

A **menu** typically consists of a title and a list of selections. Similar to restaurant menus, menus in the HP OSF/Motif environment display a list of selections from which people choose an appropriate action. Menus provide people with a simple means to quickly access the functions in your application. This chapter discusses the following aspects of menus:

- What types of menus there are.
- What components make up menus.
- How people operate menus.
- What are the common HP OSF/Motif menus.
- How to design HP OSF/Motif-consistent menu extensions.

7.1 What Types of Menus There Are

The HP OSF/Motif environment has the following four types of menus:

Pull downs Menus that "pull down" from a fixed location in the menu bar.

Pop ups Menus that "pop up" at the current pointer location, wherever that may be.

Cascading Menus that "cascade" to the right from another menu, providing more detailed selections related to the original menu selection.

Option Menus that display from an "option button" in a window.

7.1.1 Pull-down Menus Are Always Available

In most cases, pull-down menus provide an important part of the communication between people and application programs. As you consider design alternatives for the user-interface of your application, you'll realize why.

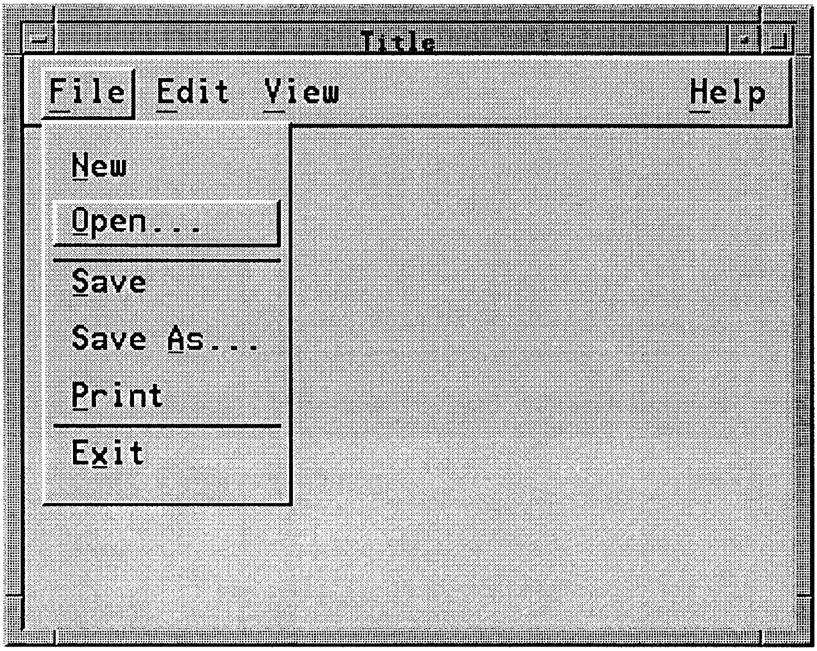


Figure 7-1. Pull-down Menu Provide Easy Access to Functionality.

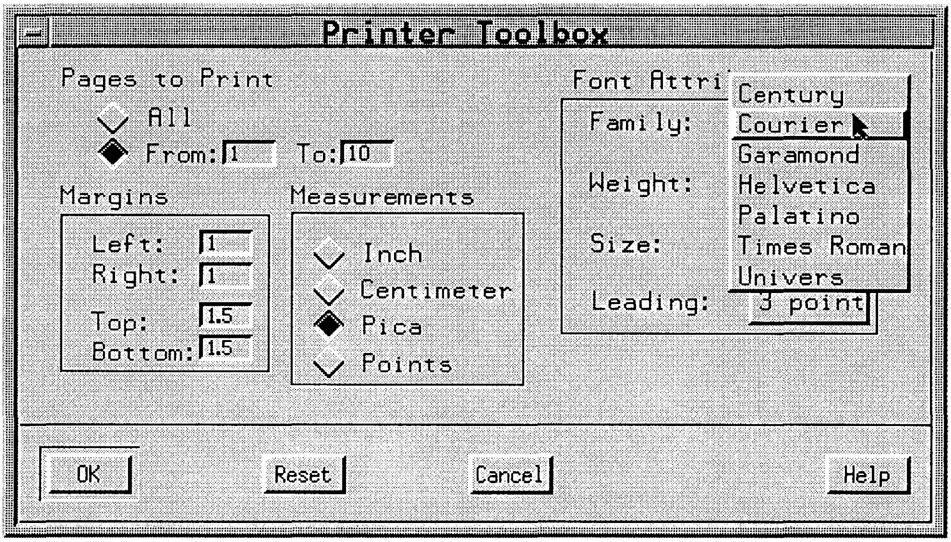
Pull-down menus, with the exception of the window menu, are always associated with a menu bar. The result is that, if you design your application to include a menu bar, the titles of available pull-down menus are always visible to the people running your application. Thus, the major portion of the functionality of your program is only a point-and-click (or point-and-drag) away from people's fingertips.

Additionally, people soon learn that they can drag the pointer across the titles in the menu bar, displaying each pull-down menu in turn, thus providing a handy "table of contents" to your application's functionality. This is another way of empowering people because it enables them to *browse* your program's functionality, refreshing their memories, rather than forcing them to *remember* the arcane command-line syntax of a particular function.

7.1.2 Option Menus

An option menu enables you to view the current setting of a multiple choice control list without displaying the entire list. Windows which include selections from a list, but where space is at a premium make good candidates for option menus.

Only the option push button is usually visible. The label on the option button displays the current setting of the control. When people select the option button, the option menu appears, showing the list of available choices. When people make a selection, that selection becomes the new label visible in the option push button.



Printer Toolbox

Pages to Print
◇ All
◆ From: 1 To: 10

Margins
Left: 1
Right: 1
Top: 1.5
Bottom: 1.5

Measurements
◇ Inch
◇ Centimeter
◆ Pica
◇ Points

Font Attri
Family: Courier
Weight:
Size:
Leading: 3 point

Figure 7-2. A Typical Option Menu.

7.1.3 Pop-Up Menus Save Space

Pop-up menus (also called "context menus") have the advantage that they take up no permanent screen "real estate." Not being associated with a menu bar, they simply "pop up" at the current pointer location. A workspace menu is an example of a pop-up menu.

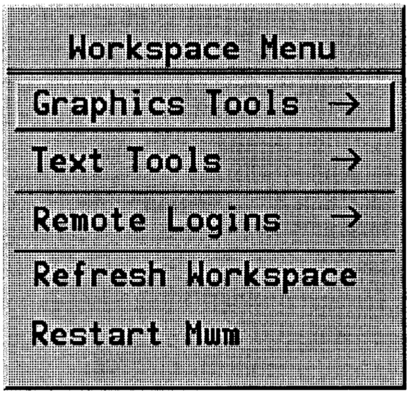


Figure 7-3. Pop-up Menus Save Space and Mouse Travel.

A second advantage of the pop-up menu is that it requires a minimum of mouse movement. To display a pull-down menu, people must position the pointer on a title somewhere in the menu bar; to display a pop-up menu, people need only press the Menu mouse button.

Pop-up menus are related to the context of the area in which they are selected. For example, a workspace menu that pops up when people position the pointer over the workspace and click the Menu mouse button may be associated with system-wide functions.

However, pop-up windows, by their very nature, do not provide a visual cue to their availability. People must learn *and remember* that a pop-up menu is associated with a certain area. Hence, your application design should not use pop-up menus too casually.

7.1.4 Cascading Menus Provide Further Selection Detail

Cascading menus add detail to pull-down and pop-up menus. You can think of them as "submenus" or "child menus" of other menus. Cascading menus provide you with a mechanism to organize menu selections in a tree structure, thus simplifying the presentation of complex selection lists. To maintain ease of use, the menu section tree should be no more than three levels deep. A cascading menu appears when people select or drag the pointer onto or across its title on the "parent" menu.

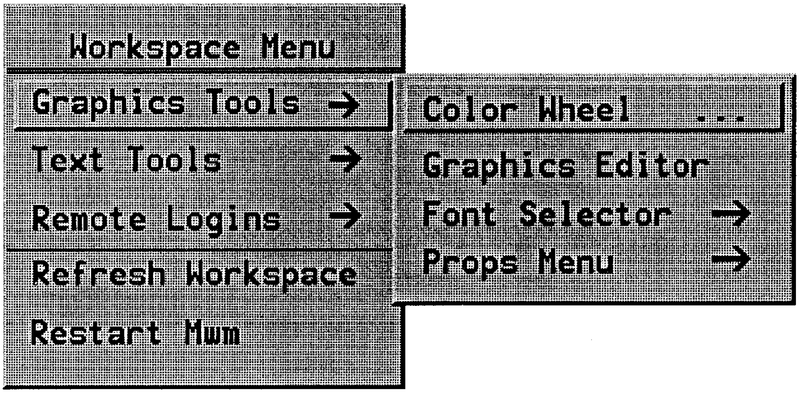


Figure 7-4. Cascading Menus in Two Tiers.

Cascading menus typically appear to the right of their parent menu selection (in countries where people read from left to right). While cascading menus differ from the two other types of menus in the method of their appearance, cascading menus behave just like pull-down and pop-up menus as far as the choosing of a selection.

7.2 What Components Make Up Menus

All menus, regardless of type, have the same components.

7.2.1 Menus Have Titles

Menus have titles that name them. A menu's title should be unique to eliminate the possibility of confusion. The title should clearly indicate the purpose of the menu.

The title of a pull-down menu is on permanent display in the menu bar. The optional title of a pop-up menu displays at the top of the menu. The title of a cascading menu displays as a selection in the parent menu.

The titles of pull-down menus that appear on the menu bar employ single-character mnemonics as memory aids to increase the efficiency of more experienced people. Mnemonics are explained later in this section.

Menu titles are visually distinct (that is visually separated in some way) from the menu's selections. Typically, this is accomplished by placing a separator line below the title.

7.2.2 Menus Have Selections

Menu selections are listed below the menu title and, like the titles of pull-down menus, can also employ mnemonics. Additionally, a menu selection lists any keyboard accelerator associated with the selection. Selections can be text or graphics. Selections can also be grouped with a separator to provide a visual cue of similarity or related functionality.

HP OSF/Motif menu selections can be one of three types. Figure 7-5 shows a sample menu containing the three selection types.

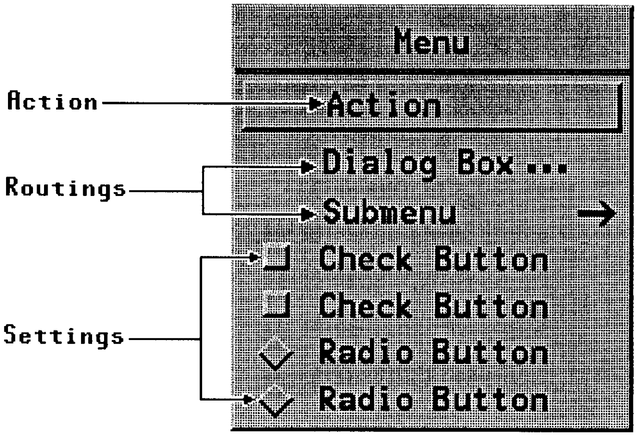


Figure 7-5. Menu Selections Have Three types.

TABLE 7-1. Menu Selection Types

Type	Description
Actions	Issue a command or carry out an action.
Routings	Display a dialog box (dialog menu items are indicated by an ellipsis (. . .)) or a cascading menu (cascading menu selections are indicated by an arrow (→)).
Settings	Set an application state using check buttons (multiple selections) or radio buttons (mutually exclusive selections).

Menu selections that are currently not available are visually de-emphasized.

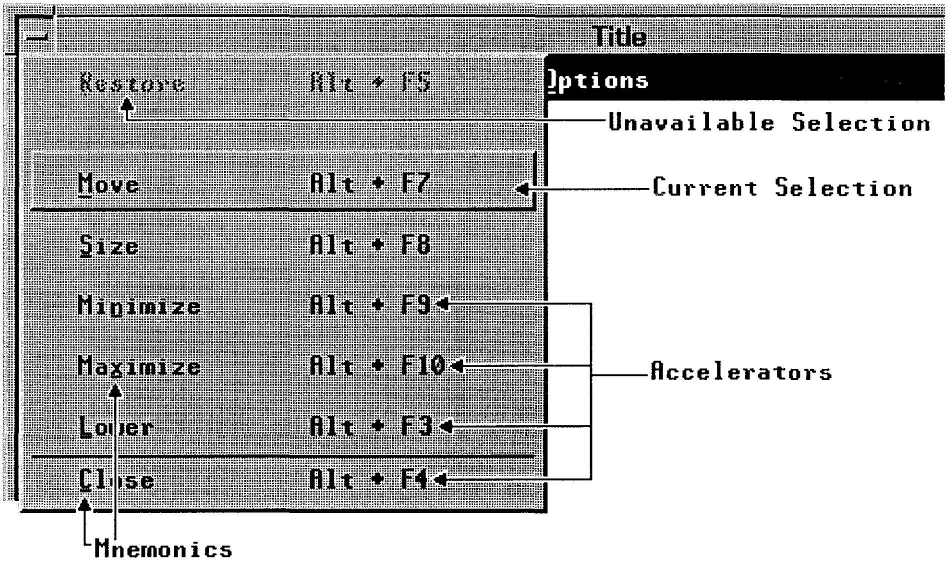


Figure 7-6. The Anatomy of a Menu.

7.2.3 Menus Have Mnemonics

A **mnemonic** is a single character that provides a shortcut for making selection from the keyboard. People type the mnemonic for a selection rather than using the navigation and select keys. In the HP OSF/Motif environment, all menus (and the titles of pull-down menus on the menu bar) have mnemonics associated with their selections.

HP OSF/Motif menu mnemonics are a single letter, usually the initial letter of the selection. Mnemonics are not case sensitive. Typing either the upper-case or lower-case mnemonic for a selection chooses that selection, the same as if the selection were chosen with a mouse operation. If the title of a cascading menu is selected, the menu is displayed. To display a pull-down menu while the location cursor is outside the menu bar, people type Alt +mnemonic.

Typically, an underline visually designates a letter as the mnemonic for a selection, as in figure 7-6 above. When an initial letter cannot be used, as in the case where two selections begin with the same letter, some other letter in the selection name, preferably with some mnemonic value, should be chosen instead. If the mnemonic letter does not appear in the selection's text, it appears in parentheses after the text.

Mnemonics are only accessible when the menu containing them is displayed. The pull-down menus in a menu bar are always accessible (by pressing Alt +*mnemonic*) since the menu bar is always displayed.

7.2.4 Menus Have Keyboard Accelerators

Menu selections can also have **keyboard accelerators**, a key or key sequence that invokes a menu selection *without* displaying the menu. Many people appreciate the shortcuts that accelerators provide and like the idea of not having to remove one hand from the keyboard to operate the mouse.

Keyboard accelerators do not have to be associated with each and every menu selection. In most cases, it is sufficient to provide accelerators only for frequently used functions. Providing accelerators should be a matter of utility, not design conformity.

If a keyboard accelerator exists for a menu selection, it appears right justified on the same line as and separated from the selection's text by enough space to make it visually distinct. Other than this, keyboard accelerators provide no visual cue to their existence and so people must memorize them. This is why frequently used functions make the best candidates for accelerators.

Accelerators, and mnemonics as well, enable people to establish a personal repertoire of frequently used commands, a flexible personal vocabulary of experience.

7.3 How People Operate Menus

People operate menus in two steps: first, they display the menu; second, they choose a selection. When a selection has been chosen (unless it leads to a cascading menu), the menu disappears.

7.3.1 Displaying Menus

In the case of pull-down menus, mouse users move the pointer to the title of the menu and press the Select mouse button. Keyboard users move the location cursor to the menu bar by typing F10, then type the pull down menu's mnemonic or use arrow keys to select the menu's title. An alternate method of displaying a pull down menu is to press the Alt modifier key and type the menu mnemonic.

In the case of pop-up menus, mouse users move the pointer over the pop-up area (for example, the workspace) and press the Menu mouse button. Keyboard users type the Menu key to display a pop-up menu.

In the case of cascading menus, mouse users move the pointer to the title of the cascading menu on the parent menu and click (or continue to press) the Select mouse button. Keyboard users use the navigation keys and Select or Enter to achieve the same effect as the mouse user's actions.

Additionally, people can access your application's menus using either of the following two methods:

Dragging — People can drag the menu, pressing *and holding down* the mouse button to maintain the display. They release the mouse button to make their choice.

Clicking — People can click the appropriate mouse button or type the appropriate key to display the menu. The menu remains displayed until either a selection is made or the display is canceled. A selection is made by moving the pointer to the selection and clicking the mouse button or by moving the location cursor to the selection and typing the Select or Enter key.

7.3.2 Browsing the Menu Bar

People can browse the menus listed on the menu bar by pressing the Select mouse button and dragging the pointer across the menu titles. As the pointer crosses each title, the menu associated with the title pulls down. To browse the menu bar from the keyboard, people display a menu and use the left and right arrow keys to move laterally across the bar.

7.3.3 Choosing Menu Selections

To choose a menu selection, people position the pointer or location cursor on the selection and make a selection action.

People using the drag method drag the pointer onto the selection they desire and release the mouse button they pressed to initiate the drag process.

Mouse users using the click method slide the pointer onto the selection and click the appropriate mouse button. Keyboard users using the click method use the navigation keys to position the location cursor on the selection they desire and type the Select key.

A menu item that has been selected provides a visual cue of its selection. Typically, this cue is a change in color, either highlighting or reversed video. In a 3-D implementation, the current selection not only changes color, it also has a raised, 3-D appearance.

Releasing the mouse button or keyboard key selects the item under the mouse pointer or location cursor. If the release occurs on a command (action), check button, or radio button, the specified action takes place, and any menus displayed disappear. If the release occurs on the title of a submenu, the menu is displayed.

7.3.4 Avoiding Menu Selection

To avoid making a menu selection, mouse users dragging the menu drag the pointer off the menu and release the mouse button. Mouse users clicking the menu slide the pointer off the menu and click the Select button. Keyboard users press the Esc key.

7.4 The Common HP OSF/Motif Menus

The following five pull-down menus provide general functions common to most applications. They are a part of most menu bars in the HP OSF/Motif environment:

- File
- Edit
- View
- Options
- Help

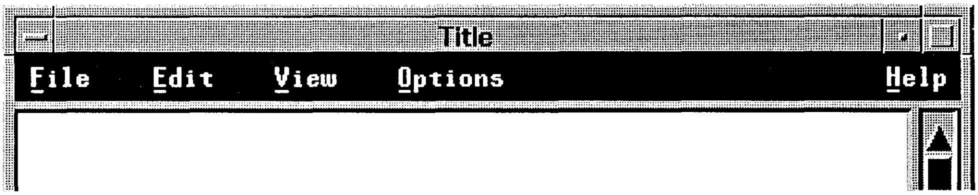


Figure 7-7. A Typical OSF/Motif Menu Bar.

While the File and Edit menus have recommended contents, the View and Option menus are more application-specific. Thus the View and Option menus illustrated here represent only sample implementations. This chapter also contains a sample Help menu. You can create additional menus for your application. The File and Edit menus (if used) should be the first two menus in your menu bar. The Help menu is always the last (right most).

While it is recommended that you include the common menus in the menu bar of your application, your choice of menu titles and items depends on the nature of your application. Should your application require it, you should design more relevant titles and selections for your application's menu bar and menus, but do not change the meanings of words used in the common menus.

7.4.1 A Look at the File Menu

The File menu presents actions that deal with files in their entirety. All applications that deals with files should provide the File menu.

The selections in the menus are divided into four groups:

- Selecting actions that connect files to your application.
- Saving actions that transfer a file to a storage medium.
- Output actions that send the changed file to an output device.
- Other actions.

These groups are important as you design extensions to the menu. Suppose you wanted to add an action called Plot to your application. You would place it in the File menu because it deals with the file as a whole. Within the File menu, you would place it with Print in the output group because Plot writes the file to an output device.

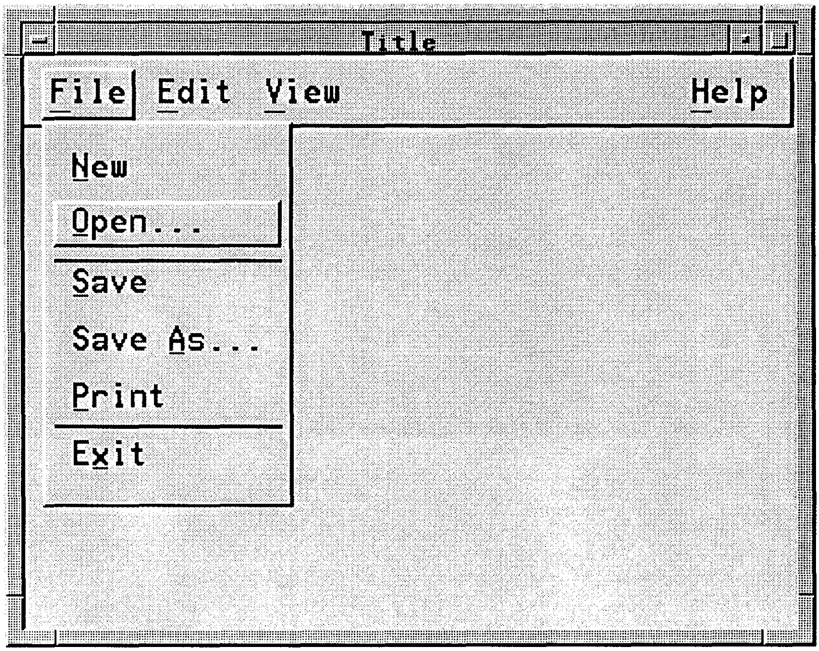


Figure 7-8. The File Menu And Its Selections.

For consistent operation with other HP OSF/Motif applications, if you include the File menu in the menu bar of your application, it should appear as the first title, placed on the far left, and with "F" as its mnemonic.

The File menu contains the following selections:

New — Creates a new file. The New operation clears existing data from the client area and replaces the current filename in the title bar with "Untitled" or some other application-generated name. If completion of the operation will obliterate current changes to the file, you must display a message box asking users whether they want to save their changes.

Open . . . — Opens an existing file. The Open operation prompts people for the name of the file by displaying a dialog box for the purpose. The title bar is updated with the name of the newly opened file. If completion of the operation will obliterate current changes to the file, you must display a message box asking users whether they want to save their changes.

Save

Saves the currently opened file to a storage device without removing the existing contents of the client area. If the currently opened file has no name, Save prompts people for a file name by displaying a dialog box for the purpose. Saved along with the file should be the current state of the file, including: window size, window location, and scrolling position within the file. This enables people to reopen the file and begin their work where they left off.

Save As . . .

Saves the currently opened file under a new name. The Save As operation prompts people for the name of the file to be saved by displaying a dialog box for the purpose. If people try to save the new file under an existing name, the Save As operation displays a warning message alerting them of a possible loss of data.

Print

Schedules a file for printing. If your application requires specific printing information before printing, the Print . . . selection displays a dialog box in which to gather it.

Exit

Ends the current application and closes all windows associated with it. This action is equivalent to closing all primary windows of the application. If completion of the operation will obliterate current changes to the file, you must display a message box asking users whether they want to save their changes.

You are encouraged to include this action even though it duplicates much of the functionality of the Close action in the window menu. This assures people have a way to end the application even if they are not running the OSF/Motif window manager. If your application does not have a File menu, put Exit at the end of the first pull-down menu.

7.4.2 A Look at the Edit Menu

The Edit pull-down menu contains actions that modify the contents of the data which your application is currently dealing. Many of the actions relate to cut and paste functionality.

The selections are grouped as follows:

- Undo actions that reverse the effect of people's actions.
- Actions that relate to the system-wide clipboard.
- Actions that do not relate to the clipboard.

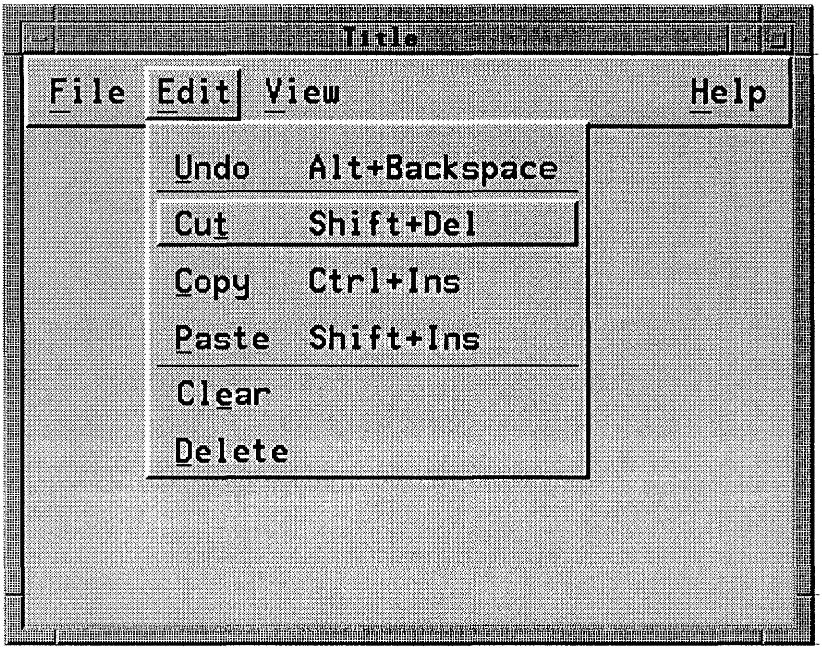


Figure 7-9. The Edit Menu And its Selections.

For consistent operation with other HP OSF/Motif applications, if you include the Edit menu in the menu bar of your application, it should appear as the second title from the left and with "E" as its mnemonic. The Edit menu contains the following selections:

Undo Alt+Backspace

Reverses the most recently executed action. To provide a visual cue to people, the Undo selection should be dynamically modified to indicate what is being undone. If the most recently executed action were a paste, the action name would be Undo paste. Your application should be able to undo all of the actions in the Edit pull-down. In addition, text applications should support Undo typing which restore Text applications should also support Undo typing which restores text after it has been selected and replaced by newly typed text.

Cut Shift+Del

Removes a selected portion of data from the client area to the "clipboard" buffer. Your application determines whether the area that used to be occupied by the removed data is left blank or whether the remaining data is compressed to fill in the space. Usually graphics applications leave the space blank while text application compress the remaining text to fill in the space.

Copy Ctrl+Ins

Copies a selected portion of data to the clipboard without removing the original data from the client area.

Paste Shift+Ins

Pastes the contents of the clipboard into a client area at the selected location. Your application determines whether the pasted data is reformatted to fit in the client area and whether existing data moves to create room for the pasted data. Text applications will typically reformat the pasted text to fit into the margins of the text field and they will move the existing text to make room for the new text. Graphics applications might do neither. They might insert the graphics unmodified and overlay existing data.

Clear

(Optional) Removes a selected portion of data from the client area without copying it to a "clipboard" buffer. Clear erases the data, but leaves the space formerly occupied by the data.

Delete

(Optional) Removes a selected portion of data from the client area without copying it to a "clipboard" buffer. The remaining data is compressed to fill the space formerly occupied by the deleted data.

7.4.3 A Look at a View Menu

A View menu enables people to control how data is displayed. The data itself remains unchanged. Entries in the menu control such aspects of presentation as:

- Appearance of the data. Examples may be iconic or text-based.
- How much of the data is down.
- In what order the data is sorted.
- To what level the data is summarized.

For consistent operation with other HP OSF/Motif applications, if you include a View menu in the menu bar of your application, it should have the "V" as its mnemonic.

The content of View menus is very application specific. Figure 7-10 shows a sample View menu for a hypothetical file browser.

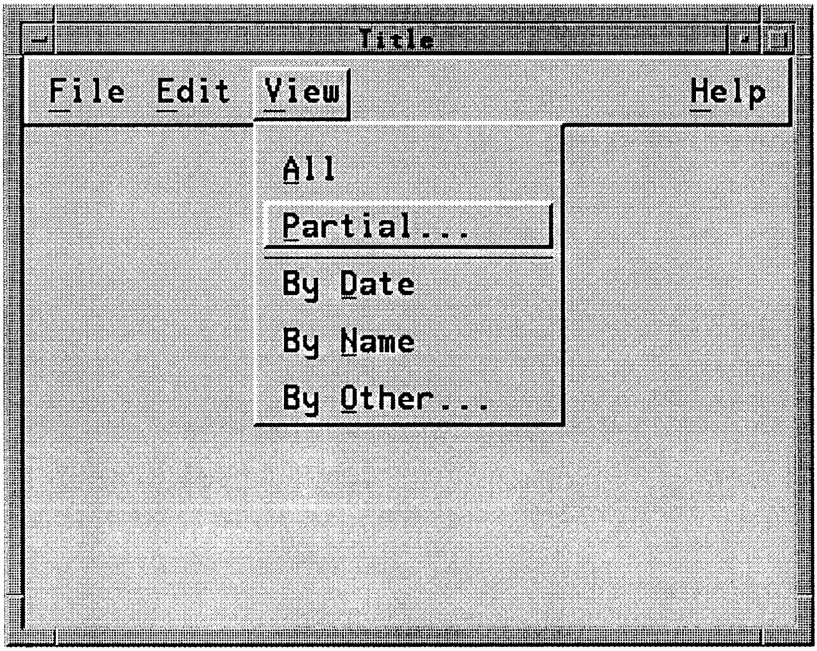


Figure 7-10. A Typical View Menu and its Selections.

The entries in the sample View menu are:

All — Views an entire list of items.

Partial . . . — Views a partial list of items. The selection criteria are determined by a dialog box.

By Date — Views a list ordered by date of entry.

By Name — Views a list ordered by item name.

By Other . . . — Views a list ordered by selection criteria determined by a dialog box.

Remember that the menu above is only an example. The needs of your application control what entries you place in your View menu.

7.4.4 A Look at an Options Menu

An Options menu enables people to customize various aspects of your application. Just like the View menu, the Options menu's content depends entirely on the needs of your application.

For consistent operation with other HP OSF/Motif applications, if you include a Options menu in the menu bar of your application, it should have the "O" as its mnemonic.

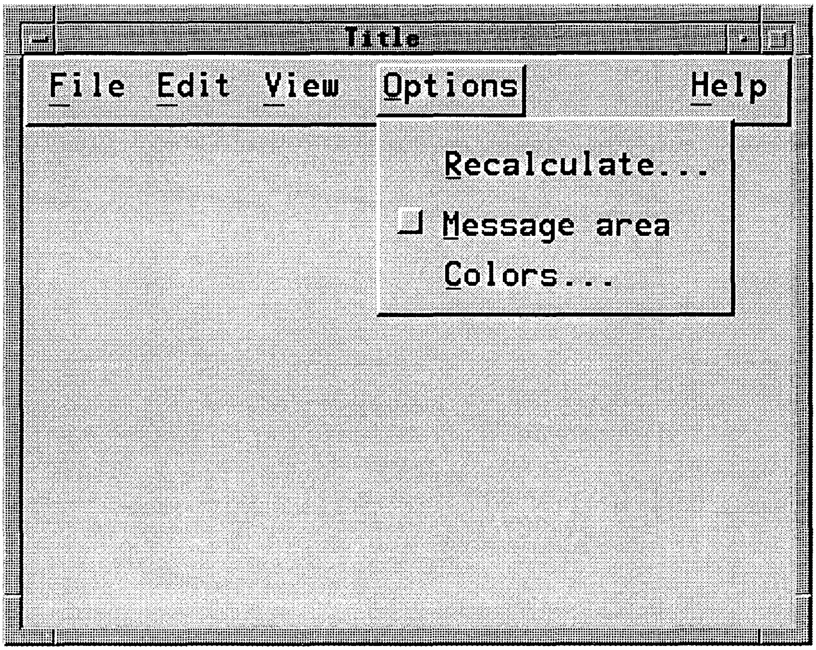


Figure 7-11. A Typical Options Menu and its Selections.

The content of Options menus is very application specific. Entries in the menu for a hypothetical spreadsheet application might include such entries as:

Recalculate . . .	Displays a dialog box to control when cell formulas are recalculated.
Message area	A check box that controls whether a message area is displayed. The message area shows helpful explanations of commands for novice users.
Colors . . .	Displays a dialog box used to select colors used by the application.

Remember that the menu above is only an example. Select entries appropriate to your application.

7.4.5 A Look at the Help Menu

Good applications provide help facilities for people to use when the need arises. No matter how intuitive you design your application to be, sometimes people "get stuck."

For consistent operation with other HP OSF/Motif applications, if you include a Help menu in the menu bar of your application, it should appear as the last title, placed on the far right of the menu bar, and with "H" as its mnemonic.

Below appears a sample help menu. Note that you should only include those functions actually supported by your application.

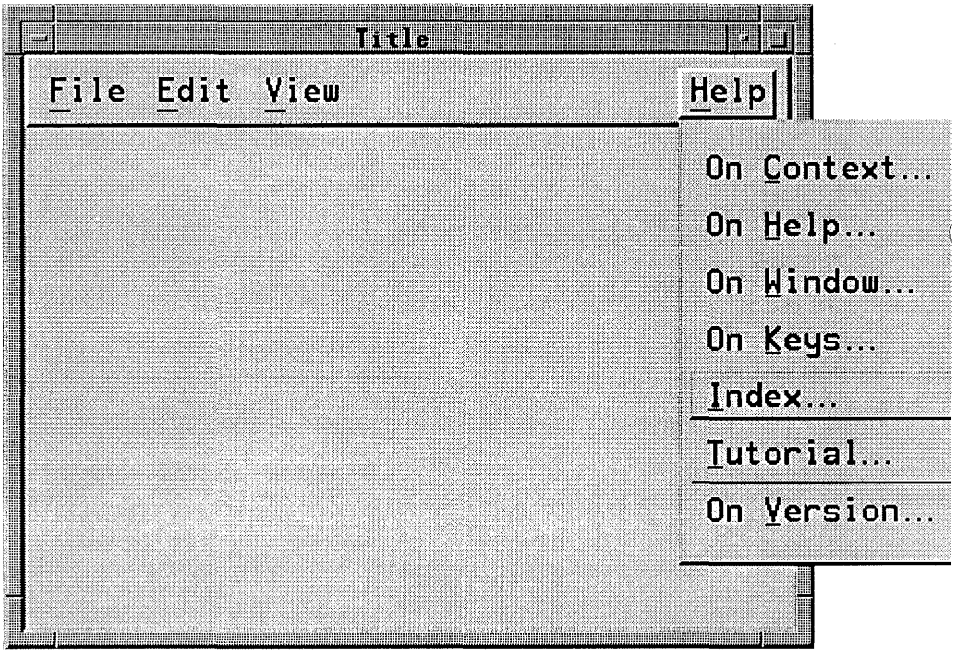


Figure 7-12. The Help Menu And its Selections.

The sample Help menu contains the following selections:

On Context . . . Provides context-sensitive help about the specific situation that exists when the help was requested.

On Help . . . Provides information on how to use the application's help facility.

On Window . . . Provides general information about the operation of the window from which the help was requested.

On Keys . . . Provides information about the application's use of function keys, mnemonics, and keyboard accelerators.

Index . . . Provides an index, with a search capability, for all help information in the application.

Tutorial . . . Provides access to the application's tutorial if such a tutorial exists.

On Version . . . Provides the name, version, and date of the application.

The help windows your application displays generally are normal secondary windows whose parent window is the window from which people requested help. The title in the menu bar is the name of the application (or name of the object in object-oriented systems) followed by the word "Help." Help menus may also have a title in the client area that describes the topic for which help is being provided.

7.5 How to Design OSF/Motif Menus

You will most likely have to design menus specific to your application. The following suggestions provide some basic guidelines for menu design.

7.5.1 Group Like Menu Selections Together

As a general rule, wherever possible, organize the selections on your menus into logical groups.

However, it's not a good idea to put a destructive command (such as Delete or Quit) next to other frequently chosen selections. This is because of a common problem of menus called the "off by one" error: People mistakenly choose the selection next to the one they intended to choose. So be aware of the result if a person mistakenly chooses the menu selection above or below the intended selection.

Use a visual cue such as a separator line to divide menu selections into logical groups.

Sets of related items, such as radio buttons or check buttons, should be located together and separated from other menu items.

7.5.2 List Selections in Order of Frequency

Order menu selections according to the frequency of usage, positioning the most frequently used selections near the top of the menu.

As much as the logical grouping of the menu allows, order the selections of your menu in decreasing frequency of usage. Within logical groups, the same principle applies: List menu selections according to the frequency of usage with the most frequently used selections at the top.

As you order menus, maintain a global perspective. Consistency across your entire application is generally more important than frequency of use in a particular menu. Evaluate frequency of use over the entire environment faced by people who use your application.

7.5.3 Keep Menu Structures Simple

If your application requires submenus, keep the menu structure simple. While it is possible to create menu structures that start from a single pull-down or pop-up menu and cascade down several levels of submenus, it is seldom necessary to do so. The awe of people who see submenu after submenu cascading down the screen quickly turns to consternation over such poor design.

Avoid using many submenus in your application. It is better to use a few more pull-down and pop-up menus, or to have more selections per menu, than to have multiple cascading submenus.

As a general guideline, use as few menu levels as possible with perhaps four levels as a maximum. A dialog box is a good alternative to menu complexity. So is redesigning your menu structure to eliminate complexity.

7.5.4 Provide Accelerators and Mnemonics

Keyboard accelerators and mnemonics enable people who have become familiar with your application to take short cuts, increasing their efficiency, while not affecting those people who are still learning the basics.

Mnemonics require the display of a menu but are preferred by some people because they allow those people to operate the mouse with one hand and make the selection with the other (as the mouse hand returns to the keyboard). Mnemonics can be made more memorable by carefully choosing the mnemonic letter.

Keyboard accelerator sequences don't require the display of a menu, hence they are active whenever your application's window is active. They are designed for menu entries that people use very frequently. To make the accelerators for your application more memorable, design your application so that accelerator sequences are consistent and progress logically.

7.5.5 Control Availability of Menu Selections

During use of your application, the application will enter a state where some menu selections will not make sense. In that case, make them unavailable while that state occurs. This prevents errors. It avoids embarrassing people with message boxes telling them that they have made an error and chosen an improper selection.

Making menu selection unavailable provides a temporary constraint. Menu selections that are *never* applicable to your application should *not* be included in the menu.

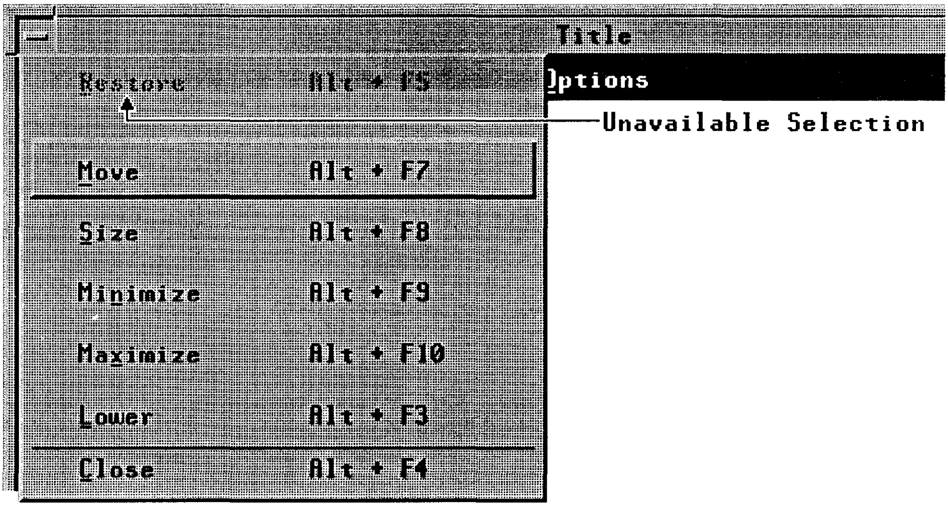


Figure 7-13. Unavailable Menu Selections Give a Visual Cue.

Unavailable menu selections provide a visual cue, such as being dimmed, that they are not functional in the current context. But even if all selections in a menu are unavailable, people should still be able to display the menu (but not choose any of the selections) and get help.

7.5.6 Consider Use of Graphic Images

An important option to consider is the use of graphic images (bitmaps) for selections. Since vision is such an important part of most peoples' sensory perception, a *good* picture can be more readily understood and easily remembered than a line of text. Graphics also ease the task of localizing your application for other countries.

7.5.7 Keep Menu Selections Stable

You should generally keep menu selections the same for the duration of an application's invocation. Settings in menus may be set or unset and any selection may become unavailable, but the selections themselves should not change.

Do not replace menu selections during an application's use. Entries that are *temporarily* unavailable should provide an appropriate visual cue. Entries that will *never* be used should be removed during application startup.

You may want to reword some menu selections slightly in order to better reflect their meaning during the current state of the application. The Undo entry in the Edit menu does this.

Adding menu entries dynamically is discouraged. If your application does require this, however, add them at the end. Number the selections and use the number as the mnemonic. This is the only case where menu selections should be numbered.

The above discussion of dynamic changes in menus applies only to changes made by the application in response to changing conditions in the application. They do not apply to any changes due to user customization.

7.5.8 Allow People to Customize Menus

Should you choose to, you can further empower the people using your application by allowing them to create their own menu titles and selections and associate them with their own choice of functions.

Dialog Boxes 8

A **dialog box** is a window that contains graphical controls that people use to converse with your application. While the HP OSF/Motif toolkit supplies you with graphical controls for most occasions, you must select the appropriate controls and create the dialog boxes for your application.

This chapter discusses the following aspects of dialog boxes:

- The characteristics of dialog boxes.
- Dialog box actions.
- The anatomy of a dialog box.
- Common dialog boxes.
- How people converse with dialog boxes.

8.1 The Characteristics of Dialog Boxes

People use dialog boxes to control the operation of your application. From a technical point of view, a dialog box is any window that solicits or displays information or instructions. While dialog boxes, because they enable people to control your application, are similar to menus, they can be much more flexible than menus. It is important that you design dialog boxes with the needs of your application and the needs of the people who use your application in mind.

8.1.1 The Purpose of Dialog Boxes

Dialog boxes have a variety of purposes. Some display information. These "message boxes" may be rather plain and relatively simple. Other dialog boxes solicit data from people. These may include elaborate combinations of text and graphics and a variety of controls (entry boxes, list boxes, radio buttons, etc.) through which people can convey information to your application.

You are not restricted to these purposes. In fact, many of your dialog boxes may serve a combination of purposes. However, you will probably notice that many of your dialog boxes require the same or similar actions. The "Dialog Box Actions" section of this chapter describes common dialog actions. Your application should take advantage of these when appropriate.

8.1.2 Ending a Dialog

Many dialog boxes disappear after people have acknowledged the information, provided the information requested, or selected an action. This is another area in which dialog boxes are similar to menus. Unlike menus however, you may want a particular dialog box to remain visible after a selection. This enables people to continue their dialog without having to redisplay the dialog box. The "Find String" dialog box of some editor programs is an example. It remains visible until people explicitly close it.

8.1.3 Making Controls Unavailable

As people use your application, the operational context that develops may make the use of certain controls inappropriate. For example, the use of the "Minimize" selection in a window menu is inappropriate when the window is already minimized. In such cases, you should make the inappropriate controls unavailable. This is also called disabling the controls. Unavailable controls are visually de-emphasized. The unavailable controls can be in any window of your application. While unavailable controls do not operate, people should still be able to get help on them.

If a control can *never* be used, that control should not appear at all. Applications that have various authorization levels, for instance, should only show those controls that people are authorized to use.

Some dialog boxes require people to complete the interaction with them before the application continues. In essence, these dialog boxes make *all* other controls in the application temporarily unavailable. This type of dialog box is sometimes called **application-modal**. Others take over the entire display and make all other windows on the screen unavailable. These windows are called **system-modal**. Message boxes that require people to perform some immediate action before processing can continue are examples of modal windows. Use this feature sparingly, since it compromises the basic premise of the HP OSF/Motif interface: empowering the user.

8.2 Dialog Box Actions

Most often dialog boxes present information or solicit data. Typically, they also provide several actions from which people select the one desired.

While on occasion your application may require special dialog box actions, most of your dialog boxes will share common actions. The HP OSF/Motif user interface has identified actions that are likely to occur in many dialog boxes and has given them common names and definitions. No dialog box will contain all of the common actions listed below. Select the ones appropriate to your application or determine your own actions using the guidelines in this section. The list is ordered in the approximate sequence they appear in dialog boxes.

Action Performs the action specified by the label. You specify actions that are appropriate to your application. See figure 8-3 for an example.

Yes Indicates an affirmative response to a question posed in the dialog box and closes the window. See note below.

No Indicates a negative response to a question posed in the dialog box and closes the window. See note below.

OK Combines Apply and Close actions (both are described below) in one convenient push button.

Apply Applies any changes made to controls in the dialog box.

Retry Causes the task in progress to be attempted again. This action is commonly found in message boxes reporting some sort of hardware error.

Stop Ends the task in progress at the next possible breaking point. This action is commonly found in progress message boxes.

Text... Opens another dialog box that extends the current dialog.

Reset Resets any changes to controls in the dialog box to the values they had when the dialog box was opened.

Cancel Combines Reset and Close actions in one convenient push button.

Close Removes the dialog box. People explicitly close dialog boxes that remain open after each use. This function duplicates the Close action in the window menu of the dialog box. You are encouraged to include it in your dialog box in case people run your OSF/Motif application without the HP OSF/Motif window manager.

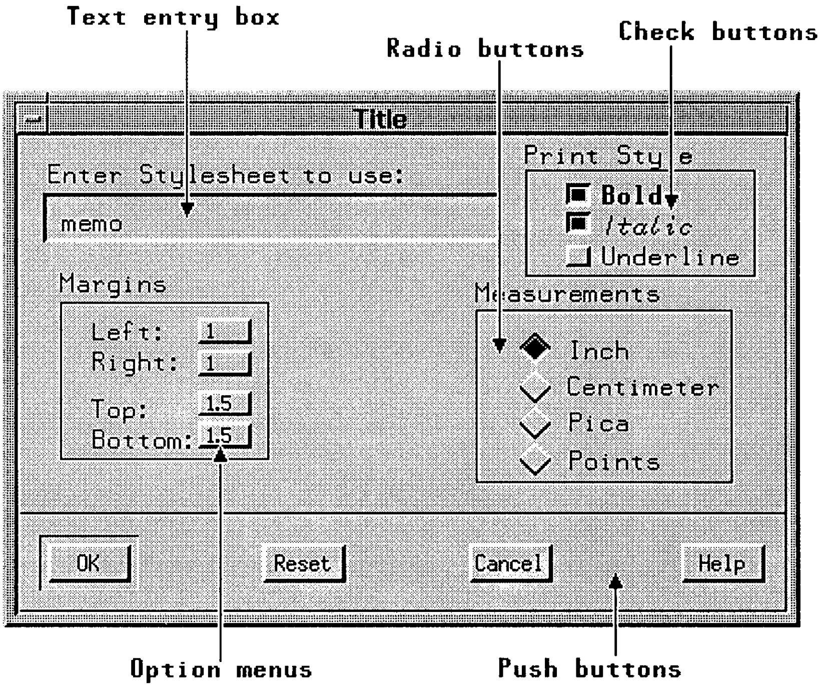
Help Enters the help subsystem.

While Yes and No are not actions, they imply "do" and "don't" and are used in dialog boxes in that context. However, you should be careful to avoid ambiguity. Only if a question is very simple, phrased without negatives, and results in an action that is not damaging should you use Yes and No. Otherwise, use an action.

8.3 The Anatomy of a Dialog Box

Dialog boxes are composed of combinations of the controls described in chapter 6.

Dialog boxes differ from one another, not because their controls differ. Controls *don't* differ in behavior. A push button, for example, is always "pushed;" a check button is always "checked." This is fundamental to common behavior. Rather, dialog boxes differ because your choice of controls and combinations of controls differs depending on the needs of your application.



Enter Stylesheet to use: memo

Margins

Left: 1
Right: 1
Top: 1.5
Bottom: 1.5

Print Style

☒ **Bold**
☒ *Italic*
☐ Underline

Measurements

◆ Inch
◇ Centimeter
◇ Pica
◇ Points

Figure 8-1. Dialog Boxes Can Contain A Variety of Controls.

While the HP OSF/Motif Toolkit provides a variety of common controls, you can create additional controls should the need arise. A stepper button is an example of an application-supplied control.

8.3.1 Arranging Push Buttons in Dialog Boxes

You should arrange push buttons in your dialog box in an order that supports people's natural progress through the controls in the dialog box. This guideline usually results in push buttons either in a row at the bottom of the window or in a column at the right .

Of the two positions, in a row along the bottom of the dialog box is preferable because push buttons are frequently used to end the dialog and thus are the last thing people encounter as they scan the contents of the dialog box.

Push buttons are commonly found in the following combinations and sequences:

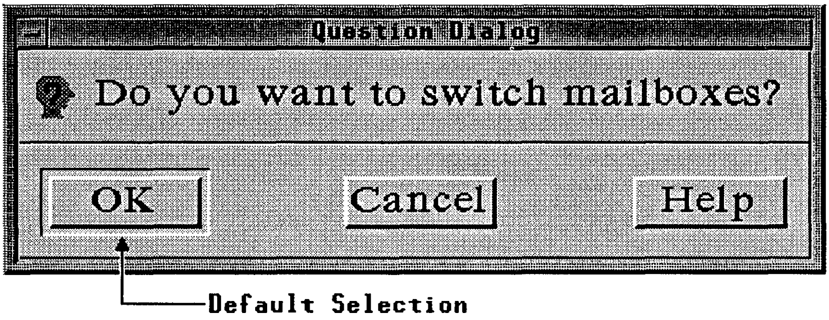
- OK Cancel Help
- Apply Reset Close Help
- Yes No Help
- Retry Stop Help

In this order, the positive (confirming) selections are toward the left, the selections negating change are toward the right, and Help is consistently placed as the right most push button. If you choose to arrange push buttons in a column, the positive selections should be toward the top with Help on the bottom.

8.3.2 Default Push Buttons

A **default push button** enables people to easily select the most likely response to a dialog box query. The default push button is always visually distinguishable from other push button selections. The OK push button is frequently the default push button in dialog boxes.

If possible, the action performed by the default push button should be reversible. If an action is potentially destructive (for example, if loss of data could occur), its button should not be made the default push button.



Do you want to switch mailboxes?

Default Selection

Figure 8-2. The Default Push Button Is Always Visually Distinct.

People can start the default push button action in either of two ways:

- By double-clicking when making a selection in the dialog box.
- By pressing Enter after making a selection in the dialog box.

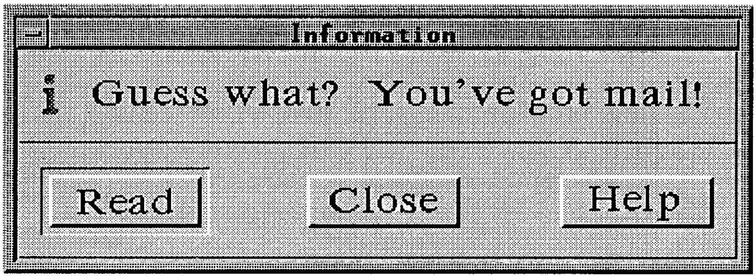
When people use the keyboard to navigate through the push buttons, the button with the location cursor becomes the default push button. This ensures that pressing Enter over a push button invokes that push button. When the location cursor leaves the push buttons, the original default button once again becomes the default.

8.4 Message Dialog Box Types

Some dialog boxes provide people with a message and ask for an acknowledgement or response of some sort. These dialog boxes, called **message boxes**, are not requested by people, but are displayed by the application as a result of some event. They can be divided into five general types depending on the nature of their message: information, progress, question, warning, and action.

8.4.1 Information

Some dialog boxes simply convey information. They inform people. They do not interrupt any tasks. For example, they may display the fact that a person has newly arrived electronic mail. When people acknowledge the information, the information box goes away. Figure 8-3 illustrates a typical dialog box that conveys information.



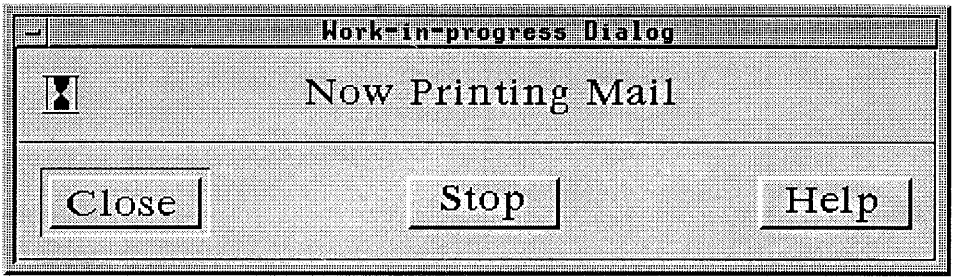
Guess what? You've got mail!

Figure 8-3. A Typical Information Box.

Here the Read push button represents an action. Conceivably, Read would start some type of mail program. Close would remove the dialog box without starting the mail program.

8.4.2 Progress

Other dialog boxes convey current progress. Their message tells people of the progress of an operation and allows people to choose between continuing the operation or canceling it. Figure 8-3 illustrates a typical dialog box that conveys the status of work in progress. You may want to show in your application's progress box how much of the total work has been completed.

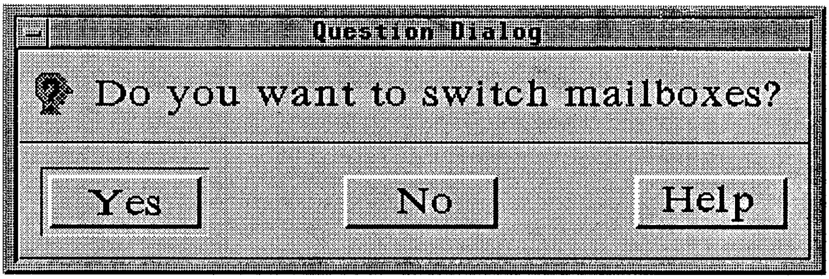


Now Printing Mail

Figure 8-4. A Typical Work-In-Progress Box.

8.4.3 Question

Some dialog boxes clarify a previous response by asking a question. The work does not proceed until the question has been answered. The question briefly explains the situation and is phrased so that the reply is a choice between mutually exclusive alternatives. Figure 8-5 illustrates a typical dialog box that asks a question.

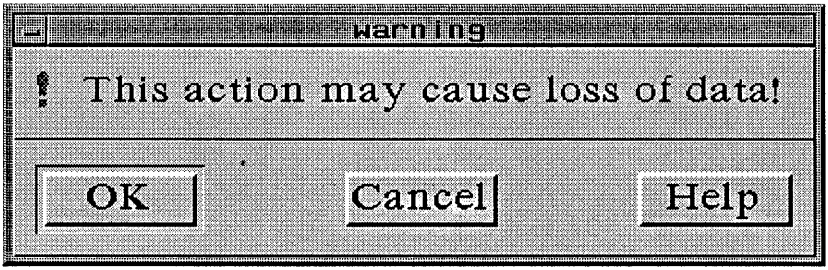


Do you want to switch mailboxes?

Figure 8-5. A Typical Question Box.

8.4.4 Warning

Other dialog boxes convey a warning. They alert people to some eminent danger (for example, the potential loss of data) and allow people to choose between ignoring the warning and continuing the operation or heeding the warning and canceling the operation. Figure 8-6 illustrates a typical dialog box that warns of possible danger.

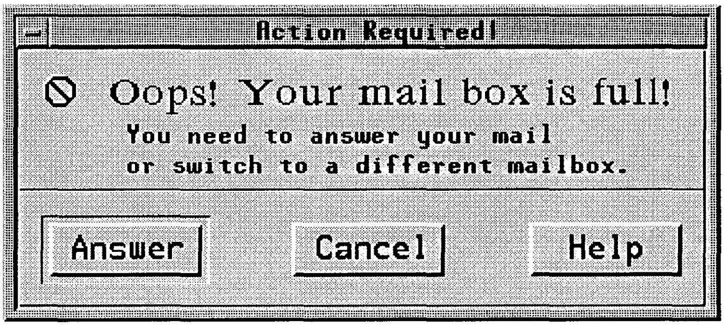


This action may cause loss of data!

Figure 8-6. A Typical Warning Box.

8.4.5 Action

Still other dialog boxes convey a message that requires immediate action. They alert people that some action is required (or that some condition exists that requires action) by constraining the operation of the application until the action has been completed. Figure 8-7 illustrates a typical dialog box that conveys an action message.


Action Required!

Oops! Your mail box is full!

You need to answer your mail
or switch to a different mailbox.

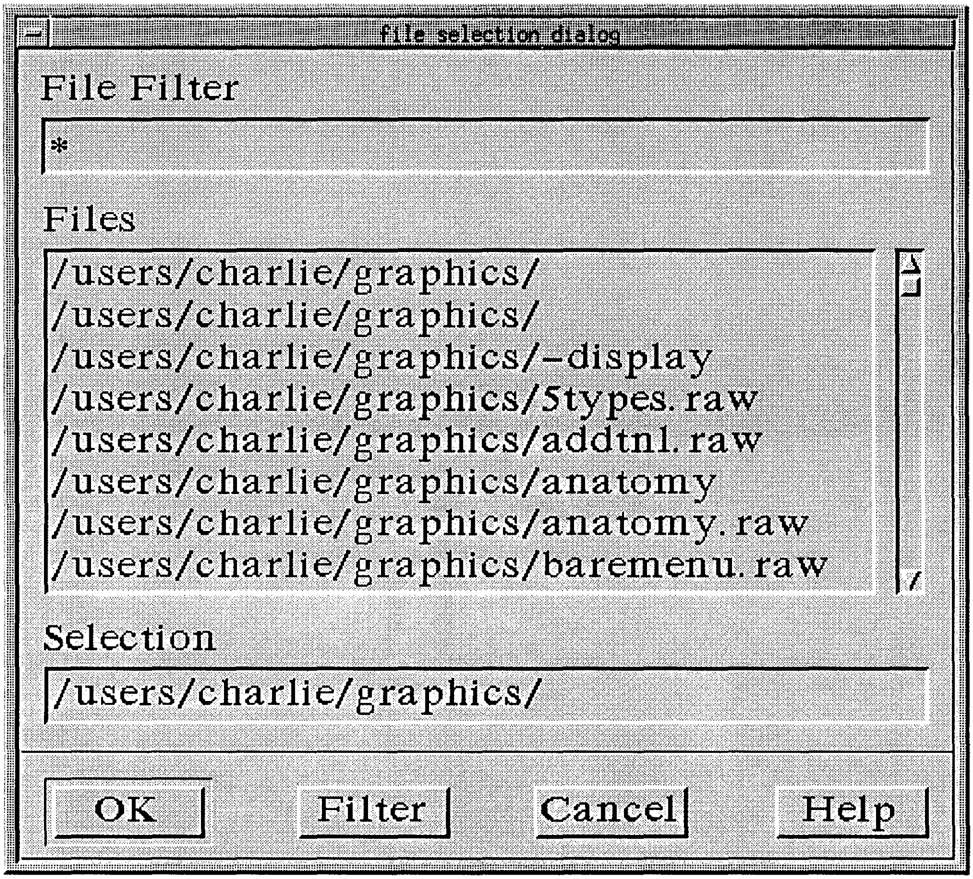
Figure 8-7. A Typical Action Message Box.

8.5 Common Dialog Boxes

Some dialog boxes are explicitly requested by people who want to perform a specific operation. Usually, these dialog boxes are application specific, but some operations are common to a wide variety of applications. Examples include operations to save a file, enter a command, and make a selection.

8.5.1 File Selection

The File Selection dialog box is used by people to enter the name of a file they want to save. Figure 8-8 illustrates a sample File Selection dialog box. The box is most commonly used in conjunction with the common File menu described in the previous chapter.



*

/users/charlie/graphics/
/users/charlie/graphics/
/users/charlie/graphics/-display
/users/charlie/graphics/5types.raw
/users/charlie/graphics/addtnl.raw
/users/charlie/graphics/anatomy
/users/charlie/graphics/anatomy.raw
/users/charlie/graphics/baremenu.raw

/users/charlie/graphics/

Figure 8-8. A File Selection Dialog Box.

8.5.2 Command

The Command dialog box is used by people to enter a command either to the application or to the computer operating system. The Command dialog box provides the same function as a command line prompt and provides a history list from which previous commands can be reselected. If your application requires command input, you should use either the command line or the command dialog box. Figure 8-9 illustrates a typical Command dialog box.

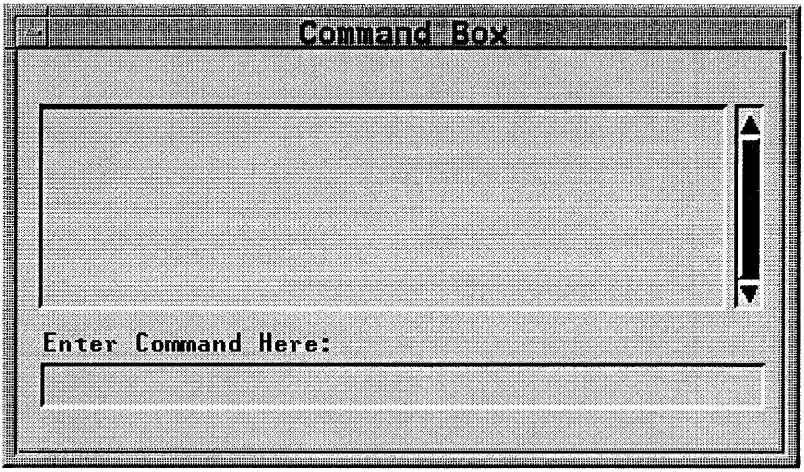


Figure 8-9. A Typical Command Dialog Box.

8.5.3 Selection

The Selection dialog box is used by people to choose from a list of several possible selections. Figure 8-10 illustrates a typical Selection dialog box.

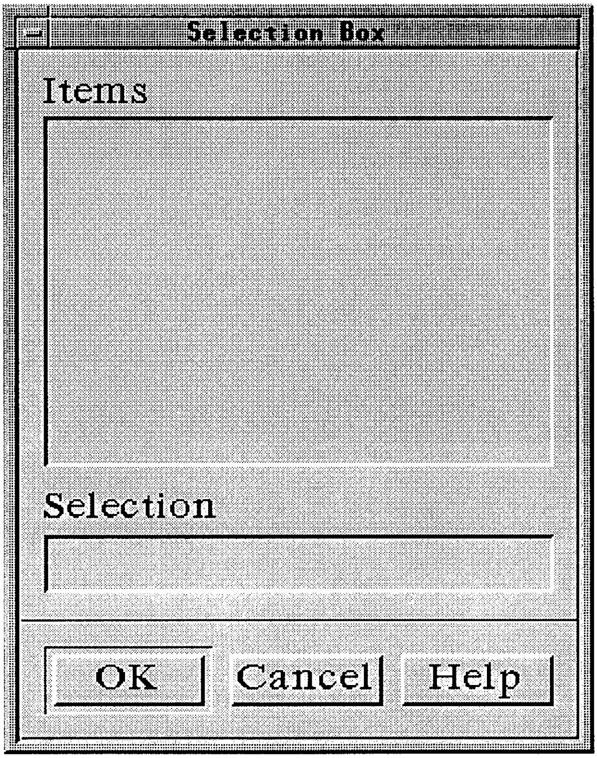


Figure 8-10. A Typical Selection Dialog Box.

8.6 Starting a Dialog

In general, dialog boxes are started by people choosing selections from controls such as menus. Menu selections that lead to dialog boxes follow the selection name with ellipsis (. . .) as a visual cue.

In the case of message boxes, the opposite is true. The application starts the dialog box based on some event. In the case of information and progress boxes, work progresses unimpeded by the presence of the dialog box. For the others, work on a task cannot proceed until a question has been answered, a warning of potential damage acknowledged, or some specific action taken.

8.7 Navigating Through a Dialog Box

In many cases, people don't need to carry out a long-winded conversation with a dialog box to get what they want. Rather, they will call the dialog box from the menu, make a few changes to some settings, and send the dialog box away. Availability of an appropriate default action is key to ease of use.

Your application should provide people with the ability to carry on a selective conversation with a dialog box and to easily navigate through the box. Mouse users can slide the pointer directly to a control. Keyboard users type the Tab and Arrow keys to step through a dialog box.

8.7.1 Determining Dialog Box Location and Size

Your application determines the size and location of its dialog boxes.

Location

Whenever possible, don't place a dialog box so that it obscures important information in the underlying window. A dialog box should be horizontally offset from the underlying window. This enables people to see the selections they have chosen.

When a dialog box relates to an item in an underlying window, position the dialog box next to the item. As much as possible, avoid covering information in the underlying window that people might refer to in conversing with the dialog box.

If it is necessary to display two dialog boxes, offset the top dialog box (the one that was called last) to the right and below the title of the underlying dialog box (the one that was called first).

Obviously, there is only a limited amount of screen real estate in which to place a dialog box. While the above suggestions seem simple enough, they can not always be followed completely. Therefore, dialog boxes, once displayed, should be movable so that people can relocate them as needed to see information in underlying windows.

Size

The initial size of a dialog box should be large enough to contain the dialog controls without crowding or visual confusion, but otherwise as small as possible.

International Implementation 9

Designing software for the international market requires producing applications that are easily translatable. In a typical scenario, when an application is translated into another language, some portion of it remains constant, while other portions change.

The constant portion forms the **base** of the product. The base contains all parts that are internationally invariable. The portion that changes from country to country is the **localization** portion. Identifying and separating the base from the localization portion is important in creating applications that are efficient to translate.

This chapter discusses the following localization issues:

- Collating sequences.
- Country-specific data formats.
- Icons and pointer shapes.
- Scanning direction.
- Text translation.
- Messages.
- Copyrights.

9.1 Collating Sequences

To produce an alphanumeric list, printable characters are sorted according to a **collating sequence**. Printable characters include letters, numbers, punctuation characters, and other symbols such as asterisks (*) and ampersands (&). The collating sequence defines the value and position of a character relative to the other characters.

Many applications make frequent use of collating sequences to produce alphanumeric lists. Examples of alphanumeric lists include the following:

- A directory listing of file names.
- The output from a sorting utility.

- An index produced by a text-processing application.
- The lists produced by a database application, such as lists of names or addresses.

If your application contains sorting functions, it needs to be flexible enough to accommodate individual country-specific collating sequences. Collating sequences must not be hard-coded into your application. At run-time, your application should refer to a table holding the collating sequences.

9.2 Country-Specific Data Formats

The formats for many types of data vary from country to country. When designing data formats, use the same format for input and display. If the format changes, it should change for both. Do not make formats dependent upon other features of your application or dependent upon each other.

Country-specific data includes the following:

- Thousands separators.
- Decimal separators.
- Positive and negative values.
- Currency.
- Dates.
- Time formats.
- Time zones.
- Telephone numbers.

9.2.1 Thousands Separators

For separating units of thousands, the comma, period, space, and apostrophe are considered to be valid separators. Your application should allow for arbitrary separators or no separators at all. The thousands separator should be definable by people who use your application or at the time of localization.

9.2.2 Decimal Separators

For separating decimal fractions, the period, comma, and the center dot are considered to be valid separators. Do not use the same character for both the thousands separator and the decimal separator. The decimal separator should be definable by people who use your application or at the time of localization.

9.2.3 Positive and Negative Values

The plus (+) and minus (-) signs are valid either before or after a number. In applications such as a spreadsheet, allow negative numbers to be enclosed in parentheses.

9.2.4 Currency

For currency, the comma, period, and colon are valid separators. They should be modified independently of the separators used for other data formats.

The currency symbol is a valid separator. It can be placed before or after the numerical value, or be used as the decimal separator. Allow for one or no space between the currency symbol and the amount. The currency symbol can be changed.

9.2.5 Dates

If your application displays or prints the date, the date should be in the format and language of the people using your application. Properly formatting the date requires the following:

- Alphanumeric date formats should allow sufficient storage and display space to accommodate date names in other languages.
- The position of each component of an alphanumeric date should be able to be varied or omitted.
- The hyphen, comma, period, space, and slash are all valid separators for the day, month, and year.
- Numeric date formats should allow the month and day fields to be reversed. For example, the 4th of August, 1990, can be displayed as either 4/8/90 or 8/4/90.
- The format for entering the date should match the display format.

9.2.6 Time Formats

For time formats, hour, minute, and second components always appear in that order. Hours and seconds are never used without minutes.

The colon, period, and space are valid separators for hours, minutes, and seconds. In 24-hour notation, the four-digit format may use a separator.

9.2.7 Time Zones

For time zone displays, allow up to three characters. Also, allow for the time offsets to be fractions, because time offsets are not always an integer number of hours from Greenwich Mean Time (GMT).

9.2.8 Telephone Numbers

Telephone numbers differ in the number of digits and the format used. The space, hyphen, period, comma, and square brackets are all valid separators.

For international numbers, the plus sign is frequently used in Europe to indicate that a number is a country code. For national numbers, it is common (but not universal) to put the city code or area code in parentheses.

9.3 Icons and Pointer Shapes

It may not always be possible to design an icon, pointer shape, or other graphical symbol that adequately represents the same object or function in different countries. Culture is inherent even in seemingly universal symbols. For example, sending and receiving mail is a commonly understood function, but representing that function with an icon of a mail box may be inappropriate because the appearance of mail boxes varies widely among countries. An envelope may therefore be a more appropriate icon.

When used correctly, graphical symbols offer the following advantages:

- They are language independent and do not need to be translated.
- They can be used instead of computer terms that have no national-language equivalent.
- They may have more impact when used with text as warnings than the text alone.

9.4 Scanning Direction

Some languages scan left to right across the page (or display screen) and from top to bottom. In other languages, however, this is not the case. The scanning direction of the country of localization may have an impact on the location of controls in dialog boxes, the order of selections in menus, and other areas.

9.5 Translating Screen Text

Well-written screen text makes an application easier for people to understand. It also makes translation easier.

In particular, relationships between nouns become very explicit in other languages. You should avoid stringing three or more nouns together. Use prepositions to clarify the relationship of the nouns.

Also, text translated from English is likely to expand 30% to 50%. If text space is limited, for example an error message restricted to one screen line, shorten the original version to allow for the translation.

While common words are easy to understand and translate, jargon, when translated into other languages, remains jargon and serves only to confuse and intimidate people who use your application. Avoid using jargon whenever the jargon is not a part of the working vocabulary of the people using your application.

Use the following guidelines to write screen text for translation:

- Brief and simple sentences are easy to understand and aid in translating.
- Affirmative statements are easier to understand than negative statements.
- Active voice is easier for both application users and application translators to understand.

9.6 Messages

Languages have different grammatical structures that must be accommodated when a product is adapted for a local environment.

Do not build messages dynamically; that is, do not store separate subsections of a message and assemble them for output. This will avoid embarrassing juxtapositions of words and phrases. Store each message as a complete string of variable length in separate modules or files. Do not embed messages in your code.

Default Keyboard and Mouse Button Bindings A

Different keyboards may have different numbers of keys or different key labels. Similarly, different mice have different numbers of buttons. Nevertheless, the HP OSF/Motif environment is able to employ a consistent behavior because it prescribes that the same function always be invoked in the same way.

In particular, common and frequently performed functions are associated with (bound to) particular sequences of keyboard key presses and mouse button operations. These default keyboard and mouse actions, and the functions they invoke, are presented in this appendix in tabular form.

A.1 Default Keyboard Assignments

A.1.1 Modifier Keys

The typical keyboard also has the following **modifier** keys:

- Shift
- Alt or Meta
- Ctrl

People use these keys to modify the meanings of other keys or mouse button operations. In other words, modifier keys are *always* used in conjunction with other keys or mouse actions. Accelerators are an example of modifier usage.

A.1.2 Keyboard Function Assignments

The following table includes navigation, function, and special-purpose keys and their associated functions.

TABLE A-1. Default Key Assignments for Functions.

Key	Function			
	No Modifier	**Shift**	**Control**	**Alt**
←	Left	Range selection	Word left	
→	Right	Range selection	Word right	
↓	Down	Range selection		Lower window to bottom of stack.*
↑	Up	Range selection		Raise window to top of stack.*
Delete	Delete	Cut	Erase to end of line	
End:	End of line		End of data	
Esc	Cancel	Window menu		Next Application
F1	Help†			
F2				
F3				Lower*
F4	Prompt/Pop-up menu†			Close window*
F5				Restore window*
F6	Switch window panes			Switch windows*
F9				Minimize window*
F10	Switch to menu bar			Maximize window*
Home	Beginning of line		Beginning of data	
PageDn	Page down		Page right	
PageUp	Page up		Page left	
Spacebar	Select†		Select†	Window Menu‡
Tab	Next Field†	Previous Field†	Next Field†	Next application‡

* An optional assignment remaining available for user convenience.

† Suggested substitution when special purpose key is not available.

‡ Provides compatibility with MS-Windows.

A.1.3 Default Key Assignments for Text Keys

The following table contains the default key assignments used by text entry boxes and other editing windows:

TABLE A-2. Default Assignments for Text Entry Keys.

Key	Function			
	No Modifier	Shift	Control	Alt
Backspace	Delete to left			Undo last action
Ins	Insert/Replace	Paste	Copy	
Del	Delete	Cut		

Entry boxes follow the rules for basic text editing including the use of the following common editing functions:

Insert/Replace Toggles between edit modes. In insert mode, the character typed on the keyboard is inserted at the location of the insertion cursor. Text that follows the cursor is shifted to the right. In replace mode, the character at the location of the insertion cursor is removed and the character typed on the keyboard is inserted in its place.

Clear Removes the selected text. The remaining unselected text is not shifted to fill in the deletion. The removed text is not copied to the clipboard.

Delete Removes the selected text. The remaining unselected text is shifted to fill in the deletion. The removed text is not copied to the clipboard. If no text is selected, the character to the right of the insertion cursor is deleted.

Delete to Left Removes the selected text to the left of the insertion cursor. Text following the cursor is shifted to the left.

Cut Copies the selected text onto the clipboard, removing the text from the entry box.

Copy Duplicates the selected text onto the clipboard, leaving the text unchanged.

Paste Copies the contents of the clipboard into the entry box at the location of the insertion cursor.

Undo last action Reverses the effect of a person's most recent editing action.

A.1.4 Default Key Assignments for Cursor Navigation

The following table contains the default key assignments used by the HP OSF/Motif window manager for cursor navigation:

TABLE A-3. Default Assignments for Cursor Navigation Keys.

Key	Description
Next Field	Moves the cursor forward to the next field going from left to right and top to bottom. At the last field, the cursor wraps to the first field.
Previous Field	Moves the cursor back to the previous field going from right to left and bottom to top. At the first field, the cursor wraps to the last field.
Home	Moves the cursor to the beginning of the current line.
Ctrl + Home	Moves the cursor to the beginning of the data.
End	Moves the cursor to the end of the current line.
Ctrl + End	Moves the cursor to the end of the data.
↑	Moves the location cursor to the previous choice in the client area. At the first choice, the cursor wraps to the last choice. In text entry areas, moves the insertion cursor up one line.
↓	Moves the location cursor to the next choice in the client area. At the last choice, the cursor wraps to the first choice. In text entry areas, moves the insertion cursor down one line.
←	Moves the location cursor to the previous choice. At the first choice, the cursor wraps to the last choice on the line above. At the top first choice, the cursor wraps to the bottom last choice. In text entry areas, moves the insertion cursor left one character.
→	Moves the location cursor to the next choice. At the last choice, the cursor wraps to the first choice on the line below. At the bottom last choice, the cursor wraps to the top firs choice. In text entry areas, moves the insertion cursor right one character.
PageUp	Scrolls a window, displaying the previous page of information.
PageDn	Scrolls a window, displaying the next page of information.
Ctrl + PageUp	Scrolls a window, displaying information previously out of view to the left.
Ctrl + PageDn	Scrolls a window, displaying information previously out of view to the right.
F6	Moves the cursor from one pane to another in a split window. Precedence is left to right, top to bottom.
F10	Moves the cursor to the menu bar.
Ctrl + ←	Moves the cursor to the previous word in a field.
Ctrl + →	Moves the cursor to the next word in a field.

A.1.5 Substitutions for Special Purpose Keys

Not every keyboard contains all the special-purpose keys mentioned in this style guide. Where a special-purpose key is available, it should be used as described unless your keyboard contains a more appropriate key. Some keyboards may not include either a more appropriate key or a special-purpose key. The following table presents some suggested substitutions for special purpose keys.

TABLE A-4. Special Purpose Keys and their Substitutions.

Key	Substitution
Select	Spacebar or Return or Ctrl + Spacebar
Menu	F4
Help	F1
Alt	Extend
Esc	F12
Enter	Return
Next Field	Tab or Ctrl + Tab
Previous Field	Shift + Tab or Ctrl + Shift + Tab

If neither a special-purpose key nor a recommended substitution is available, function key equivalents should be used. Applications that do not need a function key for one of the common functions may use that function key for application-specific purposes.

A.2 Default Mouse Assignments

A.2.1 Default Button Bindings

The HP OSF/Motif environment assumes a three-button mouse. The following figures illustrate the default HP OSF/Motif mouse button locations and button assignments for right- and left-handed mice:

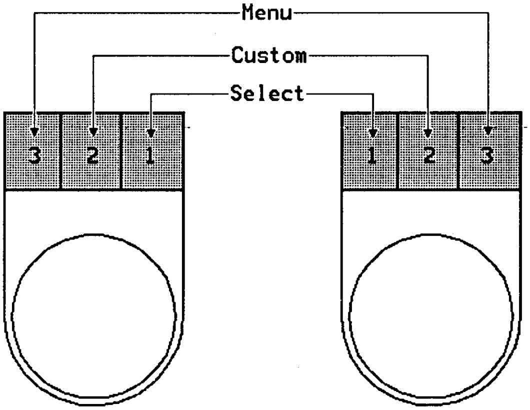


Figure A-1. Default HP OSF/Motif Mouse Button Assignments.

A.2.2 Operating Scroll Bars with a Mouse

TABLE A-5. Operating Scroll Bars with a Mouse.

Mouse Action	Response
Clicking the Select button with the pointer on stepping arrows.	Highlights the stepper arrow and moves the window through the underlying file by a single unit, in the direction indicated by the arrow. You must determine the appropriate unit to be scrolled in your application. For example, a unit in a spreadsheet may be a single row or column.
Pressing the Select button with the pointer on stepper arrows.	Highlights the stepper arrow and causes a continuous scroll, in unit steps, in the direction indicated by the arrow.
Clicking the Select button with the pointer in the scroll region.	Moves the window through the underlying file by one window length minus one unit for overlap. Clicking below (or to the right of) the slider advances to the next window's worth of information. Clicking above (or to the left of) the slider moves back to the previous window's worth of information.
Pressing the Select button with the pointer in the scroll region.	Continuously moves the window through the underlying file by one window length minus one unit for overlap until the user releases the select button or until the slider moves under the pointer.
Dragging the slider with the Select button pressed.	Moves an outline of the slider. Releasing the select button repositions the window to a location consistent with the new slider location. The user can release the select button when the pointer is anywhere in the window, not just within the scroll region. This action can be canceled by clicking any of the other buttons before releasing the select button.

A.3 Object Selection

TABLE A-6. The HP OSF/Motif Object-Action Selection Model.

Selection Task	**Mouse Button Operation**	**Keyboard Selection Operation**
Select a single object (set the anchor point) and deselect all other objects.	Click Select	Type Select
Select a range of objects (set the anchor point at range beginning) and deselect all other objects.	Drag Select	Type Shift +navigation keys.
Moves the end point of the range to the current location.	Type Shift +click Select	Type Shift + Select
Toggle the selection of an additional object.	Type Ctrl +selection operation.	Type Ctrl +selection operation.

Glossary of Terms G

accelerator

A key or sequence of keys (typically a modifier key and some other key) that provides a "shortcut," immediately accessing a program function.

active window

The window where what you type appears. If there is no active window, what you type is lost. Only one window can be active at a time.

button

A button on a mouse pointing device; mouse buttons can be mapped to the keyboard. A graphical control that works by "pressing" it.

button binding

Association of a mouse button operation with a window manager or application function.

cancel

A label given to a push button in some dialog boxes that performs the action of closing the dialog box without implementing any changes.

cascading menu

A submenu that provides selections that amplify the parent selection on a pull down or pop up menu.

check button

A control used to select settings that are not mutually exclusive. The visual cue to the selection is frequently that the button is filled in or "checked."

click

To press *and release* a mouse button. The term comes from the fact that pressing and releasing the buttons of most mice makes a clicking sound.

client

An application program written specifically for the X Window System.

client area

The area within the borders of a primary window's frame that is controlled by an application.

clipboard

An area used to store data during cut-and-paste operations.

close
A label given to a push button in some windows that performs the action of closing the window.

Ctrl+click
Clicking the mouse while pressing and holding down the Ctrl key.

Ctrl key
The keyboard key usually labeled Ctrl and used as a modifier key.

control panel
An area of a window similar to the control panels in real life that is used to hold push buttons and other graphical controls.

Ctrl+selection operation
Pressing and holding down Ctrl and then pressing the Select mouse button or the Select key to toggle the selection state of an additional selection.

cursor
A graphical image used to locate the focus of keyboard activity in a window. The insertion cursor, usually a pipe (|) or block, shows where text or graphics will appear. The location cursor, usually a rectangle, shows where a selection will be made when the Select key is typed or the Select mouse button is clicked.

default (selection)
An object or action that is specified for selection if no other selection is specified.

desktop
Another term for workspace.

dialog box
A secondary window that displays under the direction of the user and contains application controls.

dimmed selection
A selection that is not currently available.

double-click
To press *and release* a mouse button twice in rapid succession.

drag
To press *and hold* a mouse button while moving the mouse. A drag ends when the mouse button is released.

entry box
An area used for text entry. Typically, an entry box is part of a dialog box.

graphical user interface
A form of communication between people and computers that uses graphics-oriented software such as windows, menus, and icons, to ease the burden of the interaction.

grayed selection
A menu selection that is not currently available and so has been "dimmed."

group box

A graphical rectangle drawn or etched around a set of controls that provides a visual cue that the controls contained within the box are logically related.

Help key

A special-purpose key on some keyboards that provides help on the object having the location cursor.

highlight

A graphic technique used to provide a visual cue to the current selection or to the current location of the input focus. Highlighting is frequently accomplished by reversing the video of the selection.

hotspot

The area of a graphical image used as a pointer or cursor that is defined as the "point" of the pointer or cursor.

hourglass

A graphical image used to symbolize the passage of time and provide a visual cue that the application is currently performing an operation.

I-beam

A graphical image used to represent the location of the mouse pointer in a text entry box and which provides a visual cue that text can be entered in an area.

Icon

A small graphical image used to represent a window. Windows can be turned into icons or "minimized" to save room or unclutter the workspace.

inactive

A window that does not have the input focus.

insertion cursor

The graphical symbol that provides the visual cue to the location of the insertion point.

insertion point

The point in a text entry area or graphics entry area, shown by the presence of the insertion cursor, where text or graphics will appear when keys are typed on the keyboard.

keyboard

One of many input devices; the traditional method of entering text into an application.

label

The text part of an icon or graphical control.

list box

A control that provides people with a scrollable list of options from which to choose.

location cursor

A graphical symbol that marks the current location of the keyboard input focus for selection. Typically, this symbol is a box that surrounds the current object. The location cursor is sometimes known as the "selection cursor."

lower

To move a window to the bottom of the window stack on the workspace.

maximize

To enlarge a window to its maximum size.

maximize button

A control button placed on the MWM window frame and used to initiate the maximize function.

menu

A list of available selections from which a person chooses. See also pull-down menu, pop-up menu, and cascading menu.

menu bar

A rectangular area at the top of the client area of a window that contains the titles of the common pull-down menus for that application.

message box

The generic name for a dialog box that provides information, gives the current state of a work in progress, asks a question, issues a warning, or draws attention to the need for some immediate action.

minimize

To turn a window into an icon. The term "iconify" is sometimes used instead of minimize.

minimize button

A control button placed on the MWM window frame and used to initiate the minimize function.

mnemonic

A single character (frequently the initial character) of a menu selection which, when the menu is displayed and the character is typed on the keyboard, initiates the selection.

modifier key

A key that, when pressed with another key, changes the meaning of the other key. Ctrl, Alt, and Shift are modifier keys.

mouse

A pointing device commonly used in conjunction with a keyboard in point-and-click, object-oriented user interfaces.

mouse button

One of the buttons on a mouse pointing device. Mouse buttons can be pressed, released, dragged, clicked, and double-clicked. Button operations signify an action such as the selection of an object.

open

To start an action or begin working with a text, data, or graphics file.

paste

Inserting data into an area. Pasting is commonly used in reference to text files where a block of text is "cut" from one area and "pasted" into another area.

point

To position the pointer or location cursor.

pointer

The graphical image that appears on the workspace and represents the current location of a mouse or other pointing device.

pointing device

A device such as a mouse, trackball, or graphics tablet that allows people to move a pointer about on the workspace and "point" to graphical objects.

pop-up menu

A menu that provides no visual cue to its presence, but simply "pops up" when people perform a particular action. Pop up menus area associated with a particular area of the workspace, such as the client area of an application, and people must memorize where these areas are.

press

To hold down a mouse button or a key. Note that to hold down a mouse button *and move* the mouse is called "dragging."

primary window

A top-level window where an application's main interaction takes place. Primary windows can be minimized.

pull-down menu

A menu that is "pulled down" from a client application's title bar.

push button

A graphic control that simulates a real-life push button. People use the pointer and mouse to "push" the button and immediately start some action.

radio button

A graphic control that simulates the buttons on a real-life car radio. Radio buttons come in sets of two or more. Each button represents a mutually exclusive selection. Radio buttons are typically used for setting states or modes.

release

To let up on a mouse button or key that has been pressed. Sometimes it is the press that initiates the action; sometimes it is the release.

resize

To change the height or width of a window.

resize border

The MWM frame part that surrounds the client area of an application and that is used to change the height or width of the window.

restore

To return an icon or maximized window to the size and location it had before being minimized or maximized.

root menu

See "workspace menu.

root window

See workspace.

save

To write a data file to a storage device for safekeeping.

scroll bar

A graphical device used to change a person's viewpoint of a list or data file. A scroll bar consists of a slider, scroll area, and scroll arrows. A person changes the view by sliding the slider up or down in the scroll area or by pressing one of the scroll arrows. This causes the view to "scroll" up or down in the window adjacent to the scroll bar.

selection cursor

See location cursor.

select

To choose an object to be acted upon or an action to be performed.

selection

The act of selecting an object or action. The object or action that is selected. Menus are composed of selection choices. Dialog boxes contain selectable controls.

Select button

The mouse button used to make a selection.

Select key

The special-purpose keyboard key used to make a selection. Keyboards without a Select key use a substitute to provide the select functionality.

setting

A parameter that does not cause an action, but rather influences the outcome of related actions. Once set, a setting influences subsequent actions.

Shift key

One of the modifier keys on the keyboard.

Shift + click

A modified mouse button operation performed by pressing the Shift key and clicking a mouse button.

Shift + select

A modified mouse button or keyboard operation performed by pressing the Shift key and then doing a selection operation with either the mouse (clicking the Select mouse button) or keyboard (typing the Select key.

Shift + drag

A modified mouse button or keyboard operation performed by pressing the Shift key and then doing a drag operation with either the mouse (pressing a mouse button and moving the pointer) or the keyboard (typing the cursor navigation keys).

size

Used as a verb to describe changing the size of a window on the workspace.

slider

One of graphical components of a scroll bar or scale. The slider is the object that is dragged along the scroll area to cause a change.

state

A generic term used to describe the condition or mode of an object or action.

submenu

A cascading menu.

system menu

See window menu.

switch

To change keyboard focus from one window to another or from one application to another.

text cursor

See insertion cursor.

title area

The area at the top of the window frame immediately beneath the resize border. The title area has two functions: it contains a title or name that identifies the window and it can be "grabbed" and dragged to relocate the window.

title bar

The bar across the top of an MWM managed window that consists of the window menu button, the title area, and the window control buttons.

transient window

A window of short duration such as a dialog box. The window is only displayed for a short time, usually just long enough to convey some information or get some operational directions.

type

As a verb, to press and release a keyboard key.

underlined letter

A letter used as a mnemonic. The underline provides the visual cue to the mnemonic function.

window
A visual area on the display screen, usually rectangular.

window decoration
The frame and window control buttons that surround windows managed by the a window manager such as the HP OSF/Motif Window Manager.

window frame
The area surrounding a window. A window frame can consist of a resize border, a window menu button, a title bar, and window control buttons.

window manager
A program that controls the size, placement, and operation of windows on the workspace. The window manager includes the functional window frames that surround each window object and may include a separate menu for the workspace.

window menu
The menu that displays when the window menu button is pressed. The window menu typically contains selections for restoring, moving, sizing, minimizing, maximizing, and closing the window.

window menu button
The graphical control button that appears at the left side of the title bar in the window frame.

workspace
The CRT screen. The area on which the windows of a person's environment display. The workspace is sometimes called the "desk," "desktop," or "root window."

workspace menu
An optional pop up menu associated with the workspace.

Index

HEWLETT
PACKARD

HP Part Number
98794-90007

Microfiche No. 98794-99007
Printed in U.S.A.   E0989

98794-90608
For Internal Use Only