X11 User's Guide for A/UX®

Contents

4 Using **xterm** / 4-1

Contents   **ix**

Figures

Preface

X11 User's Guide for A/UX tells you how to use X11, one of Apple Computer's implementations of X Window System, Version 11, Release 4, for the A/UX® operating system. X Window System, also known as X, was developed at the Massachusetts Institute of Technology (MIT) and is considered a windowing standard in the UNIX® community.

This guide provides hands-on instructions for working in the X11 environment, including information about modifying window behavior and appearance, and procedures that show you how to use the X application programs supplied with the X11 product.

△ **Important**   X11 does not run within the A/UX software Finder™ environment. If you want to use X Window System in the A/UX Finder, you should use the MacX™ product. See *X11 Installation Guide for A/UX* for details. △

What's in this guide

Here's what you'll find inside this guide:

- Chapter 1, "Introduction to X11," gives an overview of X11, introducing concepts, terminology, and the X11 software architecture.

- Chapter 2, "Getting Started," guides you through the basics, including starting up and shutting down the X server, opening and closing client application windows, and running A/UX commands inside terminal emulator windows.

- Chapter 3, "Using twm," shows you how to use the twm window manager, including instructions for using keyboard techniques and menu commands to manipulate the windows on your screen. It also shows you how to use menus to open terminal emulator windows and set preferences for your keyboard, mouse, and the screen saver.

- Chapter 4, "Using xterm," describes the xterm terminal emulator. You use xterm windows to open other X client application windows, to run A/UX commands, or to run a program that doesn't provide its own window.

- Chapter 5, "Customizing X11," shows you how you can modify many aspects of your environment and the X11 user interface. The information in this chapter is intended for experienced users.

- Chapter 6, "Working With Fonts and Graphics," describes X applications that you can use to work with fonts, bitmap patterns, and screen dumps.

- Chapter 7, "Desktop Client Applications," describes X applications that create their own windows on a user's desktop. These X applications include graphics demonstration programs and applications that are similar to Macintosh® desk accessories.

- Chapter 8, "Advanced User and Programmer Tools," describes X applications that provide information and services useful for advanced users, programmers, and developers.

Who should read this guide

X11 User's Guide for A/UX is for both new and experienced X Window System users. For beginners, it presents tutorial steps that help you learn the basics, including how to use the window manager to manipulate windows on your screen, how to use the terminal emulator, and how to run A/UX commands and X applications inside terminal windows.

For experienced X users, this guide provides instructions for customizing the X11 environment, including how to change the contents of menus, redefine window manipulation techniques, and modify the look and feel of the screen and console.

How to use this guide

All users should begin by reading the introductory chapter, which provides an overview of X Window System architecture and operations. Then, if you are a beginner, you will probably want to read through Chapters 2, 3, and 4. Follow the steps in the tutorials at your own pace to learn how to start and stop X11 and perform basic operations. Once you know how to run X applications, move windows on the screen, and work with menus, you can go through the remaining chapters, which build on the skills you learn in earlier chapters.

If you are an experienced X user and are familiar with using window managers, terminal emulators, and X client applications, you can skim Chapters 2, 3, and 4 to ensure that you are familiar with implementation details of X11. Then read Chapter 5 to find out how to customize your environment. In addition, if you are interested in the X applications that you can use to program in the X11 environment, read Chapter 8.

What you should already know

This guide assumes a general familiarity with the UNIX operating system and UNIX commands. You need no previous experience with X Window System to use this guide.

Conventions used in this guide

A/UX guides follow specific conventions. Words that require special emphasis appear in specific fonts or font styles. The following sections describe the conventions used in all A/UX guides.

Keys and key combinations

Certain keys on the keyboard have special names. These modifier and character keys, often used in combination with other keys, perform various functions. In this guide, the names of these keys are in Initial Capital letters followed by SMALL CAPITAL letters.

The key names are

CAPS LOCK	ESCAPE	SHIFT
COMMAND	LEFT ARROW	TAB
CONTROL	RETURN	UP ARROW
DOWN ARROW	RIGHT ARROW	

For example, suppose you enter

```
Applee
```

instead of

```
Apple
```

To erase the additional *e,* you would position the cursor (or insertion point) to the right of the word and press the DELETE key once.

Sometimes you will see two or more names joined by hyphens. The hyphens indicate that you use two or more keys together to perform a specific function. For example,

Press COMMAND-K

means "Hold down the COMMAND key and press the K key."

Terminology

In A/UX guides, a certain term can represent a specific set of actions. For example, the word *enter* indicates that you type an entry and press the RETURN key. The instruction

Enter ls

means "Type ls and press the RETURN key."

Here is a list of common terms and the corresponding actions you take.

Term	Action
Choose	Activate a command in a menu. To choose a command from a pull-down menu, click once on the menu title while holding down the mouse button, and drag down until the command is highlighted. Then release the mouse button.
Click	Press and then immediately release the mouse button.
Drag	Position the pointer on an object, then press and hold down the mouse button while moving the mouse. Release the mouse button when the object reaches the desired position on the screen.
Enter	Type the letter or letters and press the RETURN key.
Press	Type a *single* key *without* pressing the RETURN key. Or position the pointer on an object and hold down the mouse button.
Select	Position the pointer on a selectable object and click the mouse button.
Type	Type an entry *without* pressing the RETURN key.

The Courier font

Throughout A/UX guides, words that you see on the screen or that you must type exactly as shown are in the Courier font.

For example, suppose you see the instruction

Type date on the command line and press RETURN.

The word date is in the Courier font to indicate that you must type it.

Suppose you then read this explanation:

Once you type `date` and press RETURN, you'll see something like this:

```
Tues Oct 17 17:04:00 PDT 1989
```

In this case, `Courier` is used to represent exactly what appears on the screen.

All A/UX manual page names are also shown in the `Courier` font. For example, the entry `ls`(1) indicates that `ls` is the name of a manual page.

Font styles

Words that you must replace with a value appropriate to a particular set of circumstances appear in *italics*. For example, if you see

`cat` *filename*

replace the italicized word with the name of the file you wish to view. If you want to view the contents of a file named `Elvis`, type the word `Elvis` in place of *filename*. In other words, enter

`cat Elvis`

New terms appear in **boldface** where they are defined.

A/UX command syntax

A/UX commands follow a specific command syntax. A typical A/UX command has this form:

`command` [*flag-option*] [*argument*]...

The following table outlines the elements of an A/UX command.

Element	Description
command	The command name.
flag-option	One or more optional arguments that modify the command. Most flag options have the form [-*opt...*], where *opt* is a letter representing an option. Most commands have one or more flag options.
argument	A modification or specification of a command, usually a filename or symbols representing one or more filenames.
[]	Brackets used to enclose an optional item—that is, an item that is not essential for execution of the command.
...	Ellipses used to indicate an argument that can be repeated any number of times.

For example, the wc command is used to count lines, words, and characters in a file. Here is the full syntax for that command, including all possible flag options and the optional argument *name*.

wc [-c] [-l] [-w] [*name..*]

Thus, you can enter

wc -w /Priscilla

to count all of the words in the file /Priscilla, where wc is the name of the command, -w is the flag option that instructs the command to count all of the words in the file, and the optional argument /Priscilla is the file to be searched.

Command reference notation

A/UX and X11 reference manuals contain references for commands, programs, and other related information. Material is organized within the reference manuals by section numbers. The standard A/UX cross-reference notation is

cmd (sect)

where *cmd* is the name of the command, file, or other facility; *sect* is the section number where the entry resides.

- Items followed by section number (1M), (7), or (8) are listed in *A/UX System Administrator's Reference*.

- Items followed by section number (1), (1C), (1G), (1N), or (6) are listed in *A/UX Command Reference*.

- Items followed by section number (2), (3), (4), or (5) are listed in *A/UX Programmer's Reference*.

- Items followed by section number (1X) are listed in *X11 Command Reference for A/UX*.

- Items followed by section number (3X) or (3Xt) are listed in *X11 Programmer's Reference for A/UX*.

For example,

`cat(1)`

refers to the command `cat`, which is described in Section 1 of *A/UX Command Reference*.

References can be also called up on the screen. Use the `man` command to display pages from reference manuals, known as manual pages, directly on the screen. For example, enter the command

`man cat`

to display the manual page for the `cat` command, including its description, syntax, options, and other pertinent information. To exit, press the Space bar until you see a command prompt, or type `q` at any time to return immediately to your command prompt.

Cross-referencing

An A/UX guide often refers to information discussed in another guide in the suite. The format for this type of cross-reference is "Chapter Title," *Name of Guide*.

For a complete description of A/UX guides, see *Road Map to A/UX*. This guide contains descriptions of each A/UX guide, part numbers, and ordering information for all the guides in the A/UX documentation suite.

Conventions specific to *X11 User's Guide for A/UX*

Many applications designed for X Window System use a three-button mouse to select text and manipulate windows. X11 uses the Macintosh computer's one-button mouse and two keyboard keys to simulate the use of a three-button mouse. This section describes the three-button mouse equivalents and other keyboard conventions specific to using X11 for A/UX.

To simulate the use of a three-button mouse, here's what you do:

- Use the actual Macintosh mouse button when an application calls for Button 1 or the Left button.

- Use the LEFT ARROW key on the Macintosh keyboard when an application calls for Button 2 or the Middle button.

- Use the RIGHT ARROW key on the Macintosh keyboard when an application calls for Button 3 or the Right button.

In addition, the COMMAND key (marked with the Apple® symbol) and the UP ARROW key are used as Meta keys, to modify mouse and keyboard commands.

Finally, when the CONTROL key is called for, you can use the actual CONTROL key or the DOWN ARROW key.

◆ *Note:* To use the arrow keys explicitly as arrow keys, hold down the OPTION key and press the desired arrow key. You can also use the -optionmouse option with the x startup script to change the way arrow keys simulate a three-button mouse. See *X11 for A/UX Release Notes* for more information about the -optionmouse option.

Figure P-1 summarizes the three-button mouse equivalents and the use of other keys.

■ **Figure P-1** Three-button mouse equivalents, Meta keys, and arrow keys

If you want to indicate	Use this	
Left mouse button or Button 1	Actual mouse button	
Middle mouse button or Button 2	LEFT ARROW key	
Right mouse button or Button 3	RIGHT ARROW key	
Meta key	COMMAND or UP ARROW key	
Control key	CONTROL or DOWN ARROW key	
True Left Arrow key	OPTION and LEFT ARROW key	
True Right Arrow key	OPTION and RIGHT ARROW key	

Additional reading

The following publications are available to help you learn about X11.

- Apple publications

  □ *X11 Installation Guide for A/UX* provides an overview of the Apple X11 for A/UX product and includes instructions for installing X11 on A/UX.

  □ *X11 Command Reference for A/UX* contains detailed information about the X11 applications described in this guide. Like *A/UX Command Reference, A/UX Programmer's Reference,* and *A/UX System Administrator's Reference,* material in *X11 Command Reference for A/UX* is organized by section numbers. Section (1X) describes X11 user commands. See "Command Reference Notation," earlier in this Preface, for more information about A/UX reference manuals.

  □ *X11 Programmer's Reference for A/UX* contains detailed information about the Xlib and X Toolkit programming libraries that come with X11. Like *A/UX Command Reference, A/UX Programmer's Reference,* and *A/UX System Administrator's Reference,* material in *X11 Programmer's Reference for A/UX* is organized by section numbers. Section (3X) describes the X11 programming libraries, and Section (3Xt) describes the X Toolkit programming library. See "Command Reference Notation," earlier in this preface, for more information about A/UX reference manuals.

  □ The suite of A/UX manuals provides specific information about A/UX. For a complete description of A/UX manuals, see *Road Map to A/UX.* This guide contains descriptions of each A/UX manual and the part numbers and ordering information.

- ◆ *Note:* X11 provides on-line versions of the manual pages presented in *X11 Command Reference for A/UX* and *X11 Programmer's Reference for A/UX.* If you prefer paper copies of these references, you can order the X11 for A/UX Manual Set, which includes all of the X11 documentation. See your Apple A/UX authorized dealer to obtain the X11 for A/UX Manual Set.

- MIT publications

  □ *Athena Widget Set—C Language Interface*

  □ *Xlib—C Language Interface*

  □ *X Toolkit Intrinsics—C Language Interface*

- *X Window System: C Library and Protocol Reference*
- *Inter-Client Communications Conventions Manual*

You can request documents from MIT by contacting

MIT Software Center
Technology Licensing Office
Room E32-300
77 Massachusetts Avenue
Cambridge, MA 02139
MIT Software Center: 617-253-6966
X Ordering Hotline: 617-258-8330

- Third-party publication for X programmers from Digital Press

  - *X Window System C Library and Protocol Reference,* by Robert W. Scheifler, James Gettys, and Ron Newman (1988)

- Third-party publications for X users and programmers from O'Reilly and Associates

  - *X Window System User's Guide* (1990)

  - *X Window System Xlib Programming Manual* (1989)

- Finally, for subscribers to Usenet, the news group `comp.windows.x` provides an online forum for news and discussion of X Window System. Ask your network administrator for more information about Usenet.

Chapter 1 **Introduction to X11**

X11 is a windowing environment for the A/UX® operating system and provides access to Macintosh® computers running A/UX and to other UNIX® systems on a network. Through the use of a high-resolution bitmapped screen, X11 provides windows in which you can run concurrent A/UX and X applications. These windows can be moved, resized, overlapped, or completely hidden from view.

This chapter is an introduction to X11. Topics include

- basic X11 concepts and operations
- the use of X11 on a network
- X11 software architecture
- X11 window hierarchy

What is X11?

X11 lets you display multiple A/UX or X applications in windows on your screen. These applications run independently and simultaneously as A/UX processes. X11 can display windows for X applications running on your computer and on computers connected over a network.

The ability to display several applications at the same time and X11's simple methods for manipulating windows provide features not available from the conventional UNIX console.

△ **Important**   To use X11, you must leave the A/UX Finder™ environment. If you want to work with client applications on your desktop, use the MacX™ application. See *X11 Installation Guide for A/UX* for details. △

The X11 startup sequence (see Chapter 2, "Getting Started") opens several windows. These windows can be moved, resized, stacked in a different order, or hidden according to your needs.

Figure 1-1 shows a sample X11 screen. For more information about the elements on the screen, see "X11 Window Hierarchy," later in this chapter.

- **Figure 1-1** A sample X11 screen
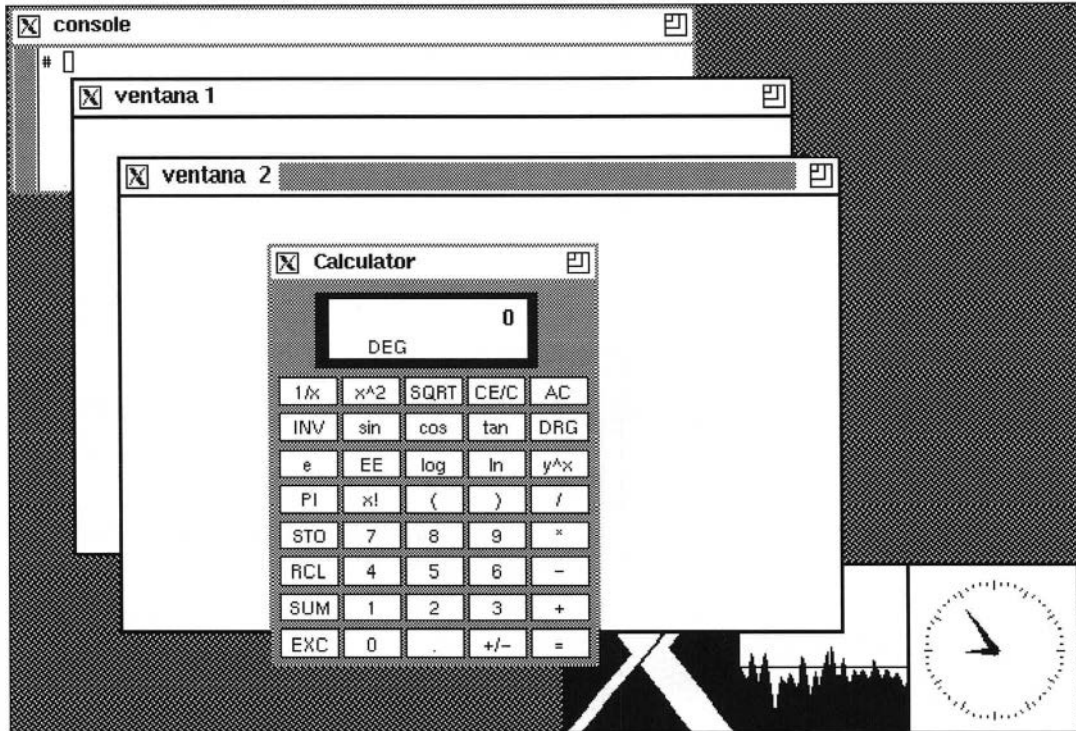
Using X11 on a network

X Window System implements a client-server model to provide windows and services to X applications that interact with the X display server over a network.

A server offers services that can be requested by clients over a network. X client applications request services from the server and rely on the server to supply them. The X server maintains the windowing environment. It controls the graphics drawn on the screen and manages the keyboard and mouse.

In the A/UX environment, the X server runs on your Macintosh. As shown in Figure 1-2, X client applications may run on your computer or on remote computers over a network. Multiple client applications can have connections to a single server, and, theoretically, a single client application may have connections to more than one server.

■ **Figure 1-2** The client-server model

Macintosh II with A/UX running the X server and local client applications

Remote client

Remote client

Remote client

Remote client applications running on other systems in the network
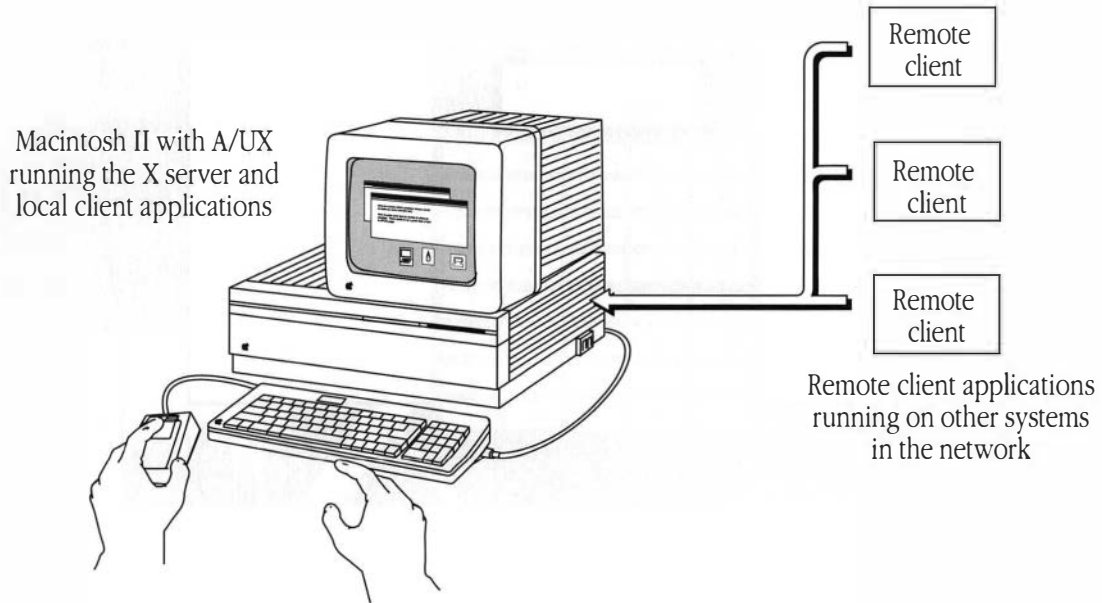
The X server operates transparently across the network, allowing you to run a client application on a remote computer and display the results on your Macintosh computer running A/UX. Network transparency is achieved through the use of the **X protocol,** a set of rules that govern the transmission of information in the X environment. The X protocol promotes device independence across all computers in the network. Any application that follows the X protocol is able to display on any X server. Figure 1-3 illustrates an X11 network configuration.

■ **Figure 1-3** An X11 network configuration

Minicomputer

Macintosh II with A/UX

Client applications coming from other

computers on a network

Mainframe computer

Macintosh II with A/UX
running the X server
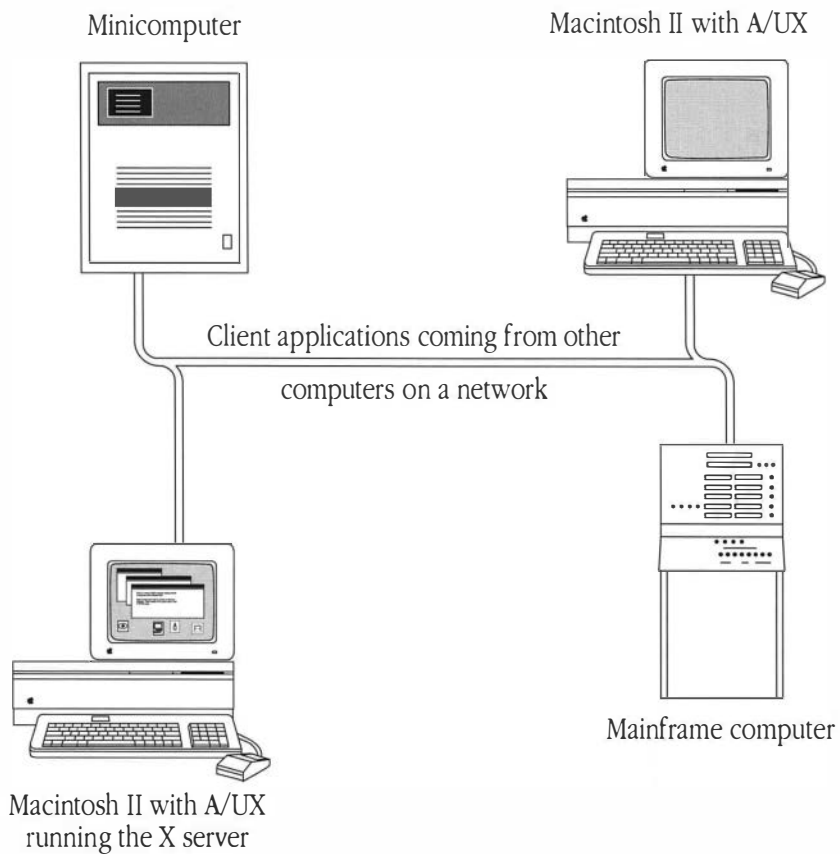
In the next section, you'll learn more about the server and its relationship to client
applications.

X11 software

The X11 distribution consists of the following software:

- X server
- X client applications
- X programming libraries

X server

Unlike file servers or print servers that are accessed by a network of users as remote devices, the X display server runs on your computer. Client applications from your computer or from other computers may request services from this server.

At the request of X client applications, and according to the X protocol, the **X server** performs window-based graphics, such as drawing windows, icons, lines, geometric shapes, and text. The server also routes input events from the keyboard and mouse to the appropriate client applications and provides the resources and mechanisms required by the client applications to implement their user interface.

An **event** is a message from the server that alerts a client application of some activity. The activity can be a press of the mouse button, keyboard input, a window move or resize operation, and so on.

For example, when you press a key on your keyboard, the X server reports this event to the appropriate client application. After interpreting the event, the client application may ignore the event or ask the server to either draw the character on the screen or perform some action.

The server is also responsible for keeping track of the pointer. The **pointer** is a graphics element drawn on the screen that follows mouse movement. The shape of the pointer changes depending on its location on the screen. For example, under the twm window manager the pointer appears as a bold X when it is located on the root (or background) window. When you move the pointer to an xterm window, it changes to an I-beam.

The location of the pointer usually determines which client application receives input events from the server. When you press the mouse button or a keyboard key, the server transmits this information to the appropriate client application. The client application receiving this information is said to have **focus.** The window belonging to the client application with focus is called the **active window.**

◆ *Note:* In the X11 environment, focus follows the location of the pointer. It is possible, however, to change this default assignment and assign focus to a window when you click the window. In this case, pointer position is no longer relevant. See the f.focus function in "Window Manager Functions" in Chapter 5 for details.

You'll learn more about the active window and using the pointer in Chapter 3, "Using twm."

X client applications

X client applications are programs that communicate with the X server and use X windows to interact with users and display information. Some frequently used client applications are a terminal emulator, a window manager, a clock, and window "snapshot" and printing utilities.

X client applications specify graphics features, such as colors, fonts, and window size, which modify the appearance or function of a client application. These features are called **resources.** You can set resource values at the command line with flag options, or you can create several types of resource files. Flag options and resource files let you customize client-application behavior according to your needs and requirements.

For example, by setting a font resource value, you can direct a client application to always use a particular font to display text in its windows. The client application specifies this font resource, and the server uses the requested font when displaying text for the client application.

Two special types of client applications are a window manager and a terminal emulator.

Window manager

A **window manager** provides techniques that you can use to move, arrange, resize, and iconify the windows on your screen.

The default and recommended window manager for X11 is the twm client application. Like all X client applications, twm communicates with and requests services from the X server. In addition, twm has the added responsibility of coordinating certain server operations to allow other X client applications to share the server's facilities.

Because X client applications are designed by independent developers, they differ in how they start up, how they shut down, how they display fonts and icons, and so on. twm ensures uniformity in an otherwise diverse environment by providing windows with certain common features and methods of operation.

Terminal emulator

A **terminal emulator** creates a window that lets you communicate with one of the A/UX shells. Anything you can do at the shell prompt in an A/UX console window, you can also do in a terminal emulator window.

The terminal emulator provided with X11 is the xterm client application. xterm communicates commands to the shell, and the shell processes these commands and sends back responses. By using several xterm windows, you can work independently in different windows—for example, editing text in one window, reading a data file in another window, and displaying graphics in yet another. You also use xterm windows to open other client applications, and client applications that do not create their own windows may use xterm windows to display their results.

xterm offers two terminal emulation modes. In DEC™ VT102 mode, xterm emulates a standard ASCII terminal. In Tektronix 4014 mode, xterm emulates an older-generation graphics terminal, allowing you run and display Tektronix-based UNIX applications on your Macintosh computer running A/UX. You can switch an xterm window from one mode to another, and you can display multiple windows of both types.

In addition to these special client applications, many other client applications, such as text editors and circuit layout systems, have been ported to the X environment. New client applications, such as word processors, spreadsheets, and engineering design tools, are also being developed.

X programming libraries

In addition to providing a user interface, window manager services, and X client applications, X11 also provides a programming interface that developers can use to design new client applications and user interface components.

The X programming interface has two components:

- Xlib is a collection of C-language routines. Xlib implements the details of the X protocol, but the details are hidden. Programmers see only a simple procedure-call interface.

- The X Toolkit, built on Xlib, is a high-level library of user interface tools, or "widgets." Widgets include such items as menus, scroll bars, window frames, panels, and icons.

By combining Xlib routines and widgets from the X Toolkit, programmers can construct a user interface for their client applications.

Xlib is documented in Section (3X) of *X11 Programmer's Reference for A/UX*. The X Toolkit library is documented in Section (3Xt) of *X11 Programmer's Reference for A/UX*. You can also find additional documentation from third-party sources. See "Additional Reading" in the Preface for more information.

X11 window hierarchy

Before you start X11, here's an overview of the elements you will see on the screen.

The startup procedure presents a screen with overlapping windows (see Figure 2-3). These windows are organized in a hierarchy. The **root window** is the first level of the hierarchy and is the foundation on which client application windows are drawn and implemented. The root window is always visible and covers the entire screen. If you connect more than one screen to the server, a root window appears on each screen. (See "Displaying Client Applications on Multiple Screens" in Chapter 2 for details about connecting more than one screen to your server.)

Client application windows are the second level of the hierarchy. These windows appear when you open an X client application. Client application windows can also have subwindows, such as individual commands in a menu, function buttons, and so on. Client application windows may be layered (or stacked) and rearranged according to your needs, but subwindows are subordinate to client application windows and may not be manipulated by window manager operations. (You can learn more about menus and function buttons in Chapter 3, "Using twm.")

Figure 1-4 illustrates a typical X11 window hierarchy.

■ **Figure** 1-4 The X11 window hierarchy
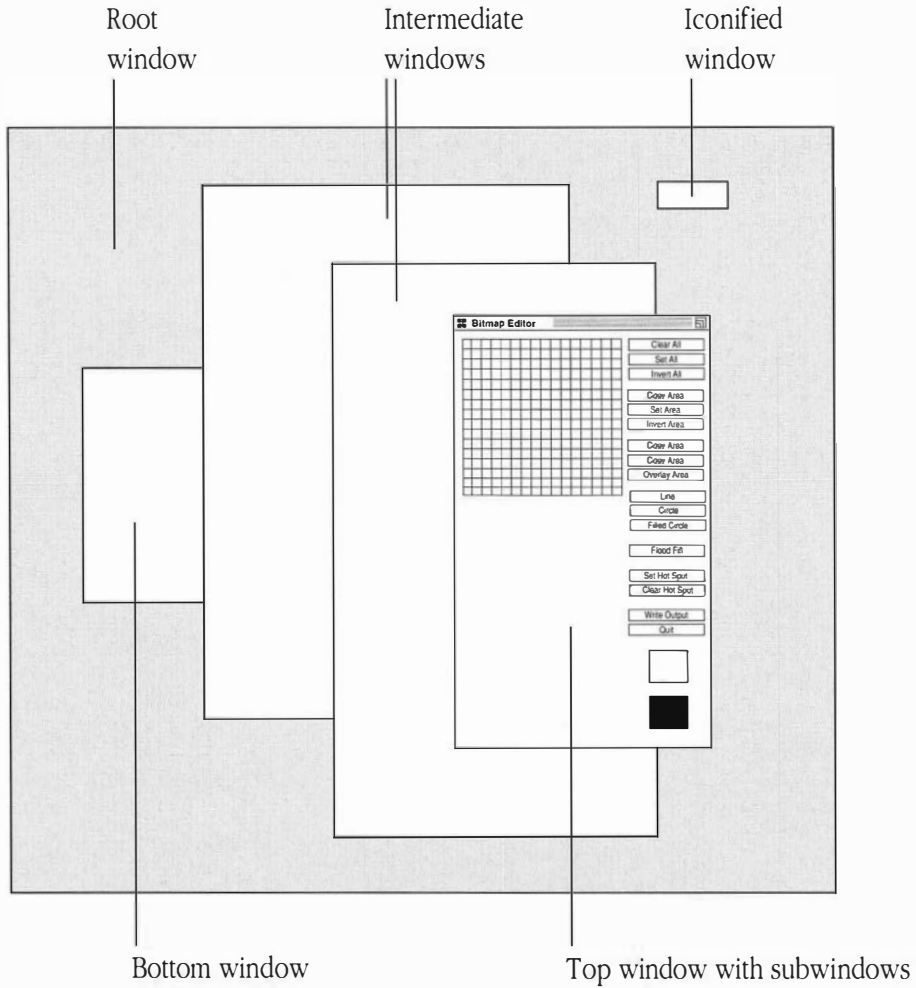
The procedures in the next chapter guide you through the basics of using X11. Follow along with the text at your own pace.

Chapter 2 **Getting Started**

This chapter is an introduction to X11 operations. You can learn how to use X11 by following along with examples in the text. Procedures show you how to

- start X11
- open X client applications
- run A/UX commands in xterm windows
- display client applications on multiple screens
- close client applications
- shut down the X server and exit the X11 environment

△ **Important** Before you begin, make sure the X11 software is installed and configured for your computer. See *X11 Installation Guide for A/UX* for installation information. △

Starting X11

To start X11, you select and open the default X11 environment. You can specify that X11 starts up whenever you log into A/UX. Or if you plan to work in one of the other A/UX startup environments at a later time, you can specify that the X11 environment is opened only for the current work session. (See *A/UX Essentials* for information about the default startup environments.)

Depending on how your A/UX system is set up, A/UX may start automatically when you turn on your Macintosh. If it does, the A/UX login dialog box appears on the screen (see Figure 2-1). If A/UX does not automatically start, see *X11 Installation Guide for A/UX* for information about starting A/UX.

To start X11, follow these steps:

1. **Start your Macintosh.**

   The A/UX login dialog box appears. While the login dialog box is displayed, you can choose items from the Options menu, located in the menu bar at the top of your screen.

■ **Figure 2-1**   The A/UX login dialog box
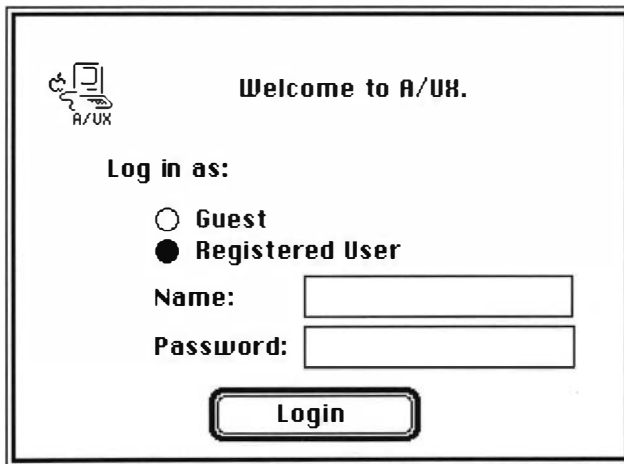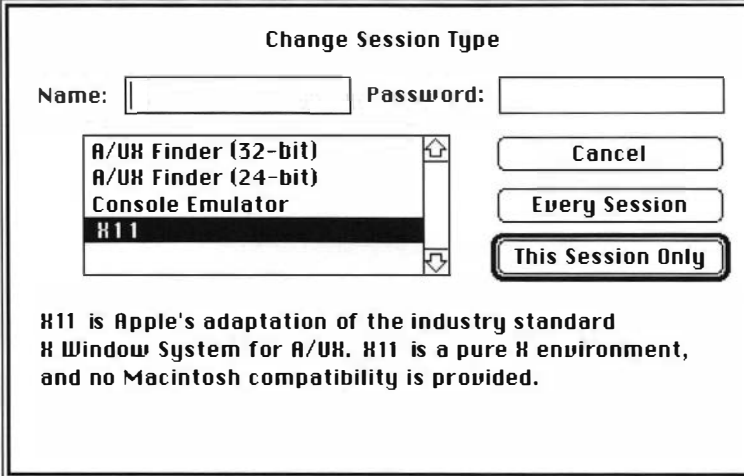
2. **Choose Change Session Type from the Options menu.**

The Change Session Type dialog box appears (see Figure 2-2). You use this dialog box to select your startup environment. You can also use it to enter your A/UX login name and password.

■ **Figure 2-2** The Change Session Type dialog box

3. **Select X11 by clicking it.**
4. **Type your login name in the Name box.**
5. **Press TAB and type your password in the Password box.**

Pressing TAB in a dialog box moves the insertion point to the next text box.

6. **Click Every Session.**

Clicking Every Session specifies that the X11 environment starts up every time you log in to A/UX. The Change Session Type dialog box disappears, and you return to the A/UX login dialog box.

7. **Click Login.**

   After a few moments, the X11 startup screen appears (see Figure 2-3).

   ◆ *Note:* If you specify X11 as your default startup environment, the next time you log in to A/UX, you do not have to choose Change Session Type from the Options menu. Just enter your login name and password in the A/UX login dialog box and click Login. The X11 environment starts automatically.

■ **Figure 2-3** The X11 startup screen
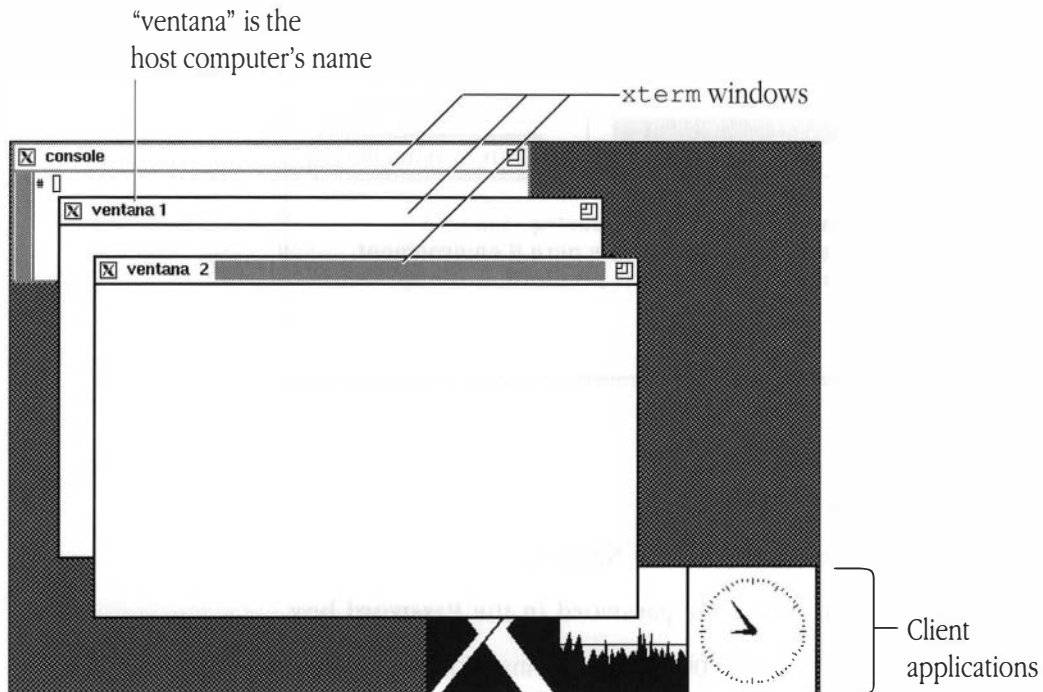
The startup screen displays three xterm windows. The xterm window named "console" is used primarily for printing system messages and for quick global exits from the X11 environment. You can use the other two xterm windows (named "ventana 1" and "ventana 2" in Figure 2-3) to run A/UX commands and to open new X client applications.

◆ *Note:* The titles for xterm windows on the startup screen are specified in the twm window manager configuration file (.twmrc) as either "console" or your host computer's name. The default title for xterm windows is "xterm" (see Figures 3-1 and 4-5). For information about specifying window titles, see the -title flag option in "Using Command-Line Flag Options" in Chapter 5.

In addition to the xterm windows, the startup screen displays three X client applications: xclock is a clock, xload is a system load average meter, and xlogo displays the X logo. You'll find out about these client applications in Chapter 7, "Desktop Client Applications."

△ **Important**   The startup procedure assumes that you are using a monochrome screen. If desired, you can specify screen color depths for color and monochrome. See "Specifying Screen Characteristics" in Chapter 5 for details. △

The next sections show you how you can use the xterm windows to open X client applications and run A/UX commands.

Opening X client applications

You can use the xterm windows on the startup screen to open new X client applications. Follow these steps:

1. **Position the pointer in an xterm window.**

   On the startup screen, the titles of the xterm windows include "console" and your host computer's name.

◆ *Note:* It's not a good idea to open client applications from the xterm window named "console." Reserve the "console" window for global X11 actions, such as exiting X11 and viewing system messages.

The pointer changes to an I-beam, the default xterm pointer, and the window's title bar is highlighted to indicate that the window is active.

2. At the command prompt, enter the name of the X client application you want to open, followed by an ampersand (&).

Be sure to press RETURN.

◆ *Note:* When you open a client application from an xterm window, always run the process in the background by adding an ampersand (&) at the end of the command line. This lets you continue using the xterm window for further processing.

For example, to open another xclock window, enter

xclock &

A rectangular grid, with a corner pointer in the top-left corner, appears on the screen (see Figure 2-4). You use this grid to position new windows on the screen.

3. Position the window grid where you want the new window to appear.

To move the window grid, you move the mouse *without* pressing or holding down the mouse button. The pointer and attached grid follow the movement of the mouse.

4. Click when you reach the desired position.

To click, press and release the mouse button. The window grid disappears, and the new window is drawn on the screen.

◆ *Note:* Whenever you open an X client application that creates its own window, you use the window grid to position the new window on the screen. The size and shape of the window grid vary according to the size and shape of the client application window.

■ **Figure 2-4** The window grid
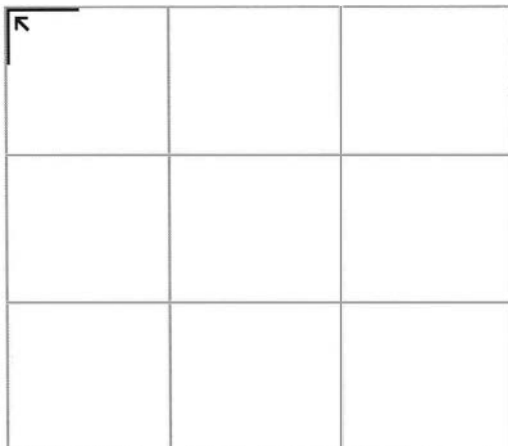
Running A/UX commands

You can also use the xterm windows on the startup screen to run A/UX commands. Follow these steps:

1. Position the pointer in an xterm window.

On the startup screen, the titles of the xterm windows include "console" and your host computer's name.

◆ *Note:* It's not a good idea to open client applications from the xterm window named "console." Reserve the "console" window for global X11 actions, such as exiting X11 and viewing system messages.

The pointer changes to an I-beam, the default xterm pointer, and the window's title bar is highlighted to indicate that the window is active.

2. **At the command prompt, enter the A/UX command you want to run.**

   Be sure to press RETURN.

   For example, to list the contents of the current directory, enter

   ```
   ls
   ```

   The contents of the current directory are listed in the xterm window.

Displaying client applications on multiple screens

You can connect more than one screen to most Macintosh computers and then display client applications on particular screens according to your needs. The X server automatically finds these screens and controls them.

◆ *Note:* You cannot display a window partially on one screen and partially on another. Each window is confined to the screen on which it is created.

The maximum number of screens you can connect depends on the type of Macintosh computer you are using. See your Macintosh owner's guide to find out how many screens you can connect to your computer. For example, with the Macintosh II and IIx computers, you can connect up to six screens (see Figure 2-5).

■ **Figure 2-5** Multiple screens
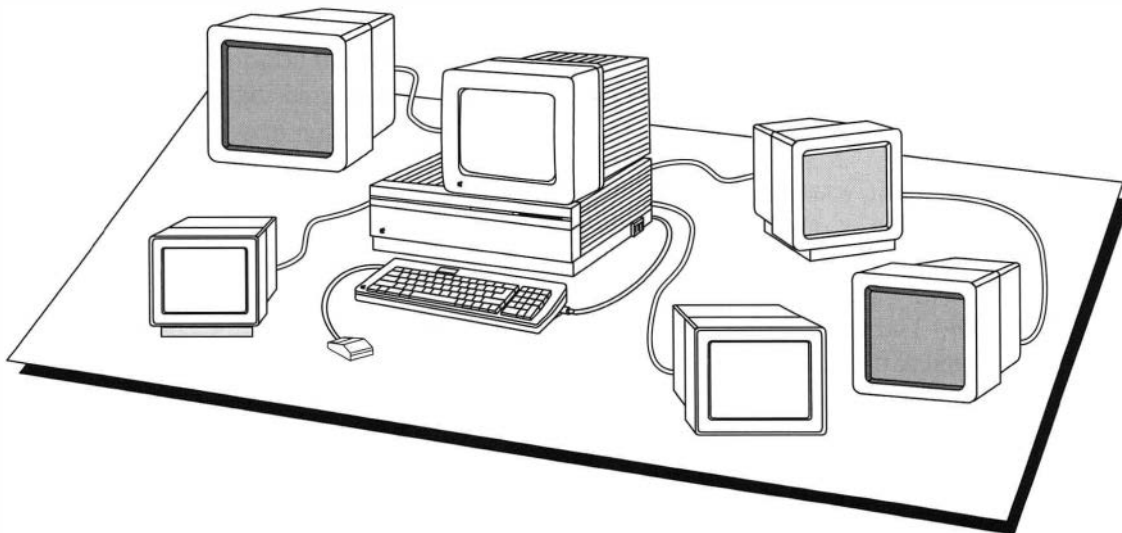
If you want to specify that a window opens on a specific screen, use this syntax when you open the client application:

clientName -display *server*:0.*screen* &

where *clientName* is the name of the client application and *server* is the name of the local *or* remote computer running the X server. The constant 0 designates the default display device, and *screen* is a number that corresponds to the relative position of the screen's video card among all the video cards that are installed in the NuBus™ expansion slots in your Macintosh.

For example, to open an xterm window on a screen whose video card is in relative position 2 in the NuBus with the server residing on a computer named "ventana," enter this command:

xterm -display ventana:0.2 &

This command displays an xterm window on the screen of the computer named "ventana."

See "Using Command-Line Flag Options" in Chapter 5 for more information about the -display flag option.

Closing X client applications

X client applications shut down in different ways depending on how they were designed. Some client applications have close boxes, some have quit buttons, and still others have quit or exit menu commands. Use one of these methods, if available, to shut down a client application.

As a last resort, you can use the A/UX `kill` command to close a window by stopping its process. See `kill`(1) in *A/UX Command Reference* for details.

◆ *Note:* You can also close an X client application by using the Window Operations menu of the `twm` window manager. See "Closing a Client Application and Removing Its Windows" in Chapter 3 for details.

Shutting down X11

You can globally close the X client applications displayed on your screen and shut down the X server. When you shut down X11 in this way, you are not prompted to save your work. Make sure to save your work before you use this global exit.

To shut down X11 and return to the A/UX login dialog box, follow these steps:

▲ **Warning**   If you want to continue with the procedures in Chapter 3, "Using `twm`," don't shut down the server at this time. Wait until you are finished with your work session and then return to this section. ▲

1. **Move the pointer to the `xterm` console window.**

   The pointer changes to an I-beam, the default `xterm` pointer, and the window's title bar is highlighted to indicate that the window is active.

2. At the command prompt, enter `exit`.

Be sure to press RETURN.

All client applications stop running, their windows are removed, the server shuts down, and the A/UX login dialog box returns to your screen.

◆ *Note:* To prevent an accidental shutdown of the server, you can change the `xterm` console window to an icon during your work session. Then you can shut down the server only after changing the window icon back to a full-size window. See Chapter 3, "Using `twm`," for information about changing windows to icons.

The next chapter shows you how to use the `twm` window manager to move and manipulate the windows on your screen.

Chapter 3 Using **twm**

twm is the default window manager for X11. twm is invoked automatically at startup and lets you manipulate windows in many ways.

In windowing environments, you use a window manager to control the size, placement, and behavior of windows on the screen. With twm, you can use mouse and keyboard techniques to manipulate windows, or you can choose commands from a menu to perform the same actions. This chapter shows you how to use both methods to work with windows. Step-by-step instructions show you how to

- make windows active
- move windows
- change windows to icons
- resize windows
- change the stacking order of overlapping windows
- use the Window Operations menu to manipulate windows
- use the New xterms menu to open new xterm windows
- use the Preferences menu to set user preferences

◆ *Note:* The features described in this chapter are controlled by entries in the twm configuration file (.twmrc) and may be changed. Before trying to change your environment, it is recommended that you work with the default features described in this chapter. Later, you can refer to Chapter 5, "Customizing X11," for information about customizing twm.

Working with windows

twm provides a default window that gives X client applications a common appearance. twm decorates each window with visible window controls that you can use to move and manipulate a window on your screen. Figure 3-1 shows these window controls on an xterm window.

△ **Important** This chapter assumes that you have started X11 by using the instructions in Chapter 2, "Getting Started," and that the X11 startup environment, shown in Figure 2-3, is visible on your screen. △

■ **Figure 3-1** The default twm window

Iconify box   Title bar   Size box
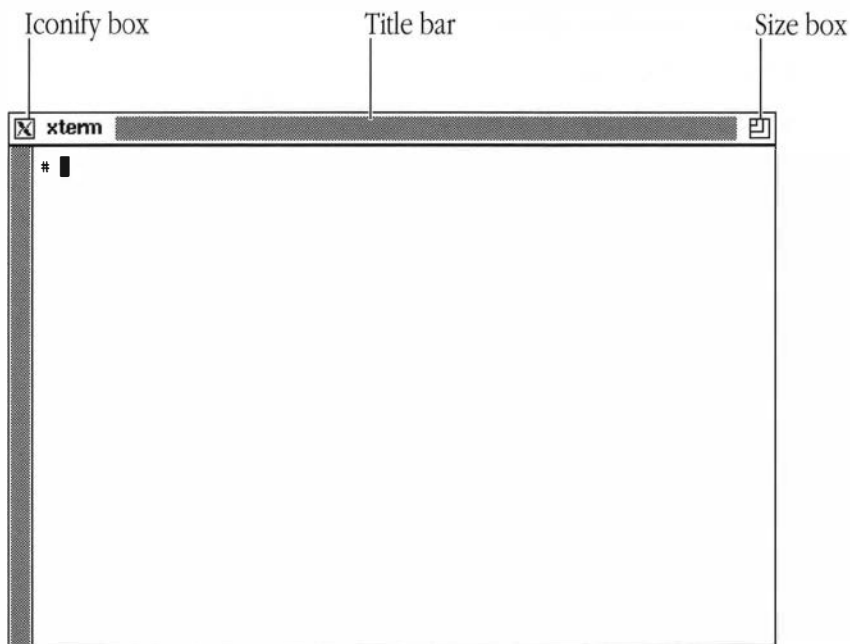
◆ *Note:* You can direct t wm *not* to decorate client application windows with window controls by specifying the NoTitle variable in your .twmrc file (see "Specifying Window Variables" in Chapter 5 for details). To manipulate windows that do not display the t wm window controls, or to use a window's controls when they are hidden under a stack of overlapping windows, you must use the Window Operations menu. See "Window Operations Menu," later in this chapter, for details.

You'll learn how to use these window controls in the following sections.

Making windows active

You must make the window you want to manipulate (move, resize, or iconify) the active window (see Chapter 1, "Introduction to X11," for information about the active window). To assign focus to a window and make it active, follow this step:

■ **Position the pointer anywhere in a window (to make it active).**

The title bar is highlighted, indicating that the window is active.

◆ *Note:* On color screens, when a window is *not* active, special color requirements that may have been specified for the client application might not be honored by X11. As a result, the window contents may appear miscolored and unintelligible. Making the window active restores the special color values to the window.

Moving windows

You move a window by dragging its title bar. Follow these steps:

1. **Position the pointer anywhere on the title bar of the window (except on the iconify or size box), and press and hold down the mouse button.**

   When you press the mouse button, the pointer changes to a compass pointer, and a window grid appears on top of the window you are moving.

2. **Drag the window grid to a new position while holding down the mouse button as you move the mouse.**

   The window stays in its original position on the screen as you drag the window grid.

3. **Release the mouse button.**

   When you release the mouse button, the window grid disappears, and the window moves to its new position.

   ◆ *Note:* To move a window that doesn't have a title bar, or if the title bar is hidden, you must use the Window Operations menu. See "Window Operations Menu," later in this chapter.

Changing windows to icons

When you have many windows displayed on the screen, it is often useful to change a window to an icon, or "iconify" it, and temporarily move the icon out of the way until you need the full-size window again. Changing a window to an icon does not close the client application or terminate an A/UX command that may be running in the window.

The type of icon you see depends on the client application. The icon can be a small box containing the name of the application, or if a specific icon is provided by the client application, the icon can be more elaborate. Figure 3-2 shows a sample X11 screen with icons.

■ **Figure 3-2** A sample X11 screen with icons
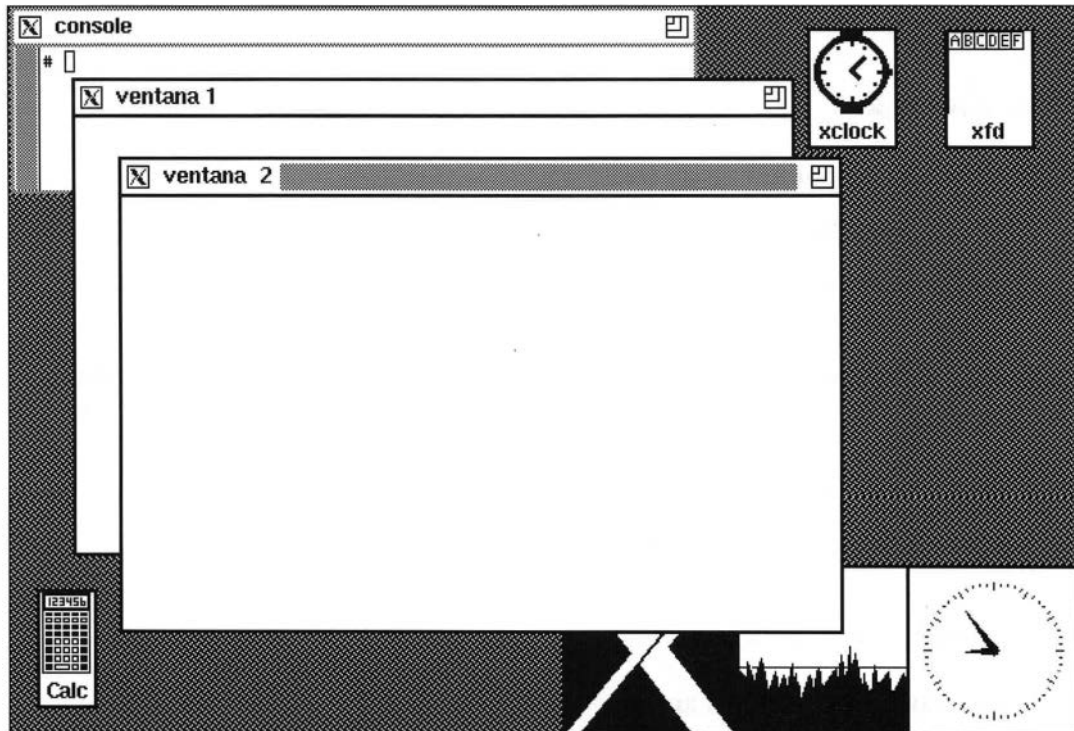
This section shows you how to

■ change a window to an icon

■ move the icon to a different location on the screen

■ change the icon back to a full-size window

To change a window to an icon

1. **Position the pointer on the iconify box of the window.**

2. **Click the iconify box.**

   The window changes to an icon.

◆ *Note:* Notice that when you iconify a window, the icon does not necessarily appear in the same place that the corresponding window occupied on the screen.

To move an icon

1. **Position the pointer on an icon.**

2. **Hold down the LEFT ARROW key.**

   The pointer changes to a compass pointer, and a window grid appears on top of the icon.

3. **Use the mouse to move the window grid to another position on the screen.**

   Do *not* press the mouse button.

4. **Release the LEFT ARROW key.**

   The window grid disappears, and the icon is redrawn in its new position.

To change an icon back to a window

1. **Position the pointer on an icon.**

2. **Click the icon.**

   The window returns to its original size and position on the screen.

◆ *Note:* To iconify a window that doesn't have an iconify box, or if the iconify box is hidden, you must use the Window Operations menu. See "Window Operations Menu," later in this chapter.

Resizing windows

To change the size of a window, follow these steps:

1. **Position the pointer on the size box of the window and hold down the mouse button.**

   When you press and hold down the mouse button, the pointer changes to a compass pointer, a small pop-up window in the top-left corner of the screen displays the current size of the window, and a window grid appears on top of the window.

2. **Drag outside the top, right, left, or bottom border of the window grid.**

   The size of the window is increased or decreased in the direction you drag, and the numeric values in the pop-up window change accordingly.

3. **Release the mouse button.**

   When you release the mouse button, the window grid disappears and the window is redrawn in its new size.

   ◆ *Note:* To resize a window that doesn't have a size box, or if the size box is hidden, you must use the Window Operations menu. See "Window Operations Menu," later in this chapter.

Changing the stacking order of overlapping windows

When you have several windows overlapping on the screen, one or more windows may become hidden from view. You can change the stacking order of overlapping windows by clicking the title bar of the window you want to move.

This section shows you how to

- move a partially obscured window to the top of a stack of overlapping windows
- move a top window to the bottom of a stack of windows in order to make other windows visible

To move a window to the top

1. **Position the pointer anywhere in the title bar of the window you want to move (except on the iconify or size box).**

2. **Click the title bar.**

   When you click the title bar, the window is redrawn in the same position, but on top of the other windows.

To move a top window to the bottom

1. **Position the pointer anywhere in the title bar of the window you want to move (except on the iconify or size box).**

2. **Press the RIGHT ARROW key.**

   When you press the RIGHT ARROW key, the window moves to the bottom of the stack of windows.

Window Operations menu

The Window Operations menu (see Figure 3-3) has a list of commands that you can choose to manipulate the windows on your screen and perform certain global operations that affect the twm client application.

■ **Figure 3-3** The Window Operations menu

```
┌─────────────────────────────┐
│      Window Operations       │
├─────────────────────────────┤
│ Redraw Screen                │
│ Source .twmrc                │
│ twm Version                  │
│ (De)Iconify                  │
│ Move Window                  │
│ Resize Window                │
│ Raise Window                 │
│ Lower Window                 │
│ Destroy Window               │
│ Kill twm                     │
│ Mouse Button Help    ▤       │
└─────────────────────────────┘
```
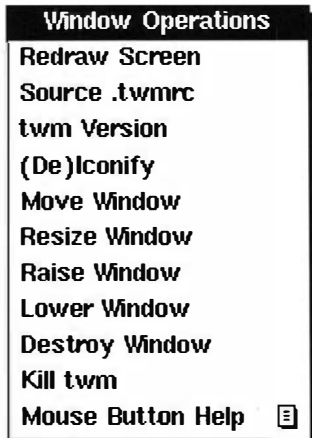
This section shows you how you can use the Window Operations menu to

■ move windows

■ change windows to icons

■ resize windows

■ change the stacking order of overlapping windows

■ display a Mouse Buttons help menu

■ redraw the windows on the screen

■ close a client application and remove its window(s)

■ tell X11 to read your twm configuration file (.twmrc) and reinvoke its values

■ display the twm version number

■ exit twm

Each command in the Window Operations menu is described in one of the following sections.

Displaying the Window Operations menu

You must display the Window Operations menu to choose a command. Follow these steps:

1. **Position the pointer anywhere on the root window.**

   The root window is the background on the screen that is used like a desktop.

2. **Press and hold down the mouse button.**

   The Window Operations menu appears, and the pointer changes to a menu pointer. The menu remains on the screen as long as you continue holding down the mouse button. You can then choose a command by dragging the pointer to it and releasing the mouse button.

   ◆ *Note:* You can also display the Window Operations menu regardless of the pointer's position by pressing and holding down the COMMAND (or UP ARROW) key and then pressing and holding down the mouse button.

Moving windows

To move a window (or icon) by using the Window Operations menu, use the Move Window command. Follow these steps:

1. **Display the Window Operations menu.**

   See "Displaying the Window Operations Menu," earlier in this chapter.

2. **Choose the Move Window command.**

   To choose a command, drag to the command while still holding down the mouse button, and when the command is highlighted, release the mouse button. The Window Operations menu disappears, and the pointer changes to a compass pointer.

3. **Position the pointer in the window (or on the icon) you want to move.**

4. **Press and hold down the mouse button.**

   A window grid appears.

5. Drag the window grid to a new position while holding down the mouse button as you move the mouse.

The window (or icon) stays in its original position on the screen as you drag the window grid.

6. Release the mouse button.

When you release the mouse button, the window grid disappears, and the window (or icon) moves to its new position.

Changing windows to icons

To iconify or deiconify a window by using the Window Operations menu, use the (De)Iconify command.

To change a window to an icon

1. Display the Window Operations menu.

See "Displaying the Window Operations Menu," earlier in this chapter.

2. Choose the (De)Iconify command.

When you release the mouse button, the Window Operations menu disappears and the pointer changes to a circle pointer.

3. Click the window you want to change to an icon.

The window changes to an icon.

◆ *Note:* Notice that when you iconify a window, the icon does not necessarily appear in the same place that the corresponding window occupied on the screen.

To change an icon back to a window

1. Display the Window Operations menu.

2. **Choose the (De)Iconify command.**

   The Window Operations menu disappears, and the pointer changes to a
   circle pointer.

3. **Click the icon you want to change back to a window.**

   The icon changes back to a full-size window in its original position.

Resizing windows

To resize a window by using the Window Operations menu, use the Resize Window command.
Follow these steps:

1. **Display the Window Operations menu.**

   See "Displaying the Window Operations Menu," earlier in this chapter.

2. **Choose the Resize Window command.**

   When you release the mouse button, the Window Operations menu disappears and
   the pointer changes to a compass pointer.

3. **Position the pointer in the window you want to resize.**

4. **Press and hold down the mouse button.**

   A window grid appears on top of the window, and a pop-up window in the top-left
   corner of the screen displays the current size of the window.

5. **Drag outside the top, right, left, or bottom border of the window grid.**

   The size of the window is increased or decreased in the direction you drag, and the
   numeric values in the pop-up window change accordingly.

6. **Release the mouse button.**

   When you release the mouse button, the window grid disappears, and the window is
   redrawn in its new size.

Changing the stacking order of overlapping windows

To change the order of a stack of overlapping windows by using the Window Operations menu, use the Raise Window and Lower Window commands. Follow these steps:

1. **Display the Window Operations menu.**

    See "Displaying the Window Operations Menu," earlier in this chapter.

2. **Choose the Raise Window (or Lower Window) command.**

    When you release the mouse button, the Window Operations menu disappears and the pointer changes to a circle pointer.

3. **Click the window you want to move to the top (or bottom) of the stack.**

    The window is redrawn in the same position, either on the top or bottom of the other windows.

    ◆ *Note:* When you use the Window Operations menu commands to change the stacking order of overlapping windows, window focus does *not* change. If the top window is the active window, it remains active when it is moved to the bottom.

Displaying the Mouse Buttons help menu

The Mouse Buttons help menu is a quick reference to the three-button mouse equivalents and Meta key assignments for the Macintosh (see Figure 3-4).

■ **Figure 3-4** The Mouse Buttons help menu

```
       Mouse Buttons
Left Button = Mouse Button
Middle Button = Left Arrow
Right Button = Right Arrow
Meta = Open Apple/Up Arrow
Control = Control/Down Arrow
```

◆ *Note:* For more information about the three-button mouse equivalents for the Macintosh, see Figure P-1 in the Preface.

To display the Mouse Buttons help menu, follow these steps:

1. **Display the Window Operations menu.**

   See "Displaying the Window Operations Menu," earlier in this chapter.

2. **Drag to the Mouse Button Help item so it is highlighted and continue dragging to the right (in the direction of the arrow).**

   When the pointer is on the arrow, the Mouse Buttons help menu appears.

3. **Release the mouse button when you're finished.**

Redrawing the screen

To redraw the contents of the windows on your screen, use the Redraw Screen command. Follow these steps:

1. **Display the Window Operations menu.**

   See "Displaying the Window Operations Menu," earlier in this chapter.

2. **Choose the Redraw Screen command.**

   When you release the mouse button, all the windows are redrawn.

   ◆ *Note:* You can also redraw the screen with the xrefresh client application. See xrefresh(1X) in *X11 Command Reference for A/UX* for details.

Closing a client application and removing its windows

You can close a client application and remove its windows from the screen by using the Destroy Window command. Follow these steps:

▲ **Warning**   Make sure to save your work before you use the Destroy Window command. ▲

1. **Display the Window Operations menu.**

   See "Displaying the Window Operations Menu," earlier in this chapter.

2. **Choose the Destroy Window command.**

   When you release the mouse button, the Window Operations menu disappears, and the pointer changes to a skull-and-crossbones pointer.

3. **Click the window you want to remove.**

   The window and its subwindows are removed from the screen. All processes associated with the window are stopped.

Reading the **twm** configuration file

The Source .twmrc command tells `twm` to read your `twm` configuration file (.twmrc) and reinvoke its values. This command is useful for testing any changes you've made to your .twmrc file. Follow these steps:

1. **Display the Window Operations menu.**

   See "Displaying the Window Operations Menu," earlier in this chapter.

2. **Choose the Source .twmrc command.**

   When you release the mouse button, the Window Operations menu disappears, and the .twmrc file is read and its values reinterpreted.

   ◆ *Note:* This command is for experienced users. See the `f.source` and `f.twmrc` functions in "Window Manager Functions" in Chapter 5 for details.

Displaying the `twm` version number

To display the `twm` version number in the title bar of the `xterm` console window, follow these steps:

1. **Display the Window Operations menu.**

    See "Displaying the Window Operations Menu," earlier in this chapter.

2. **Choose the twm Version command.**

    When you release the mouse button, the Window Operations menu disappears and the `twm` version number appears in the title bar of the `xterm` console window.

    ◆ *Note:* The version number disappears when you press the mouse button, LEFT ARROW, or RIGHT ARROW, or when you move the pointer to another window.

Exiting `twm`

The Kill twm command closes the `twm` client application and disables its window manipulation capabilities. To close the `twm` window manager, follow these steps:

▲ **Warning** If you want to continue working with the procedures in this chapter, do *not* use this command. This command is for experienced users only. ▲

1. **Display the Window Operations menu.**

    See "Displaying the Window Operations Menu," earlier in this chapter.

2. **Choose the Kill twm command.**

    When you release the mouse button, the Window Operations menu disappears, the windows on your screen no longer display the default window controls, and you are no longer able to access the `twm` menus.

New xterms menu

twm provides a menu that lets you quickly open an xterm window in one of three sizes: 12 lines, 24 lines, or 30 lines. Figure 3-5 shows the New xterms menu.

■ **Figure 3-5**   The New xterms menu

```
New xterms
xterm 24 lines
xterm 30 lines
xterm 12 lines
```
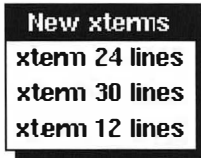
To display the New xterms menu and open a new xterm window, follow these steps:

1. **Position the pointer anywhere on the root window.**

    Do *not* press the mouse button.

2. **Press and hold down the LEFT ARROW key.**

    The New xterms menu appears, and the pointer changes to a menu pointer. The menu remains on the screen as long as you continue pressing the LEFT ARROW key.

3. **Choose a command for the desired window size.**

    When you release the LEFT ARROW key, the New xterms menu disappears, and after several seconds, a window grid with a corner pointer in its top-left corner appears on the screen. You use this window grid to position the new window on the screen.

4. **Position the window grid where you want the new window to appear.**

    Do *not* press the mouse button. To move the window grid, you move the mouse. The pointer and attached grid follow the movement of the mouse.

5. **Click when you reach the desired position.**

   The window grid disappears, and the new xterm window is drawn on the screen, displaying your host computer's name in the title bar.

   ◆ *Note:* You can also display the New xterms menu regardless of the pointer's position by pressing and holding down the COMMAND (or UP ARROW) key and then pressing and holding down the LEFT ARROW key.

Preferences menu

You use the Preferences menu (see Figure 3-6) to choose features that customize the X11 console. The features you can customize include auto-repeat, mouse tracking, and the screen saver.

◆ *Note:* The features you can set with the Preferences menu may also be set with the xset client application. See "Customizing the Console With xset" in Chapter 5 for more information.

■ **Figure 3-6** The Preferences menu


```
        Preferences
 Autorepeat On
 Autorepeat Off
 Fast Mouse Tracking
 Slow Mouse Tracking
 Fast Screen Saver
 Slow Screen Saver
```

Each command in the Preferences menu is described in one of the following sections, but first you'll find out how to display the Preferences menu.

Displaying the Preferences menu

You must display the Preferences menu to choose a command. Follow these steps:

1. Position the pointer on the root window.

Do *not* press the mouse button.

2. Press and hold down the RIGHT ARROW key.

The Preferences menu appears, and the pointer changes to a menu pointer. The menu remains on the screen as long as you continue pressing the RIGHT ARROW key. Then you can choose one of the commands, as described in the following procedures.

◆ *Note:* You can also display the Preferences menu regardless of the pointer's position by pressing and holding down the COMMAND (or UP ARROW) key and then pressing and holding down the RIGHT ARROW key.
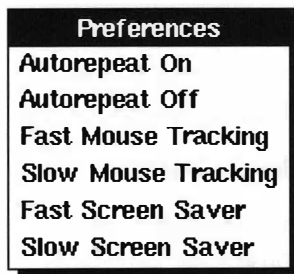
Turning auto-repeat off or on

When you press a key on the keyboard and hold it down, the X server automatically repeats that key's character until the key is released. This is called **auto-repeat.**

By default, auto-repeat is turned on when you start X11. To turn off auto-repeat, choose Autorepeat Off from the Preferences menu. To turn auto-repeat back on, choose Autorepeat On. Follow these steps:

1. Display the Preferences menu.

See "Displaying the Preferences Menu," earlier in this chapter.

2. Choose the Autorepeat Off or Autorepeat On command to change the auto-repeat setting.

When you release the RIGHT ARROW key, the Preferences menu disappears.

Setting the mouse tracking

When you start to move the mouse, it moves the pointer at a slow, precise rate across the screen until some distance has been covered. Then the pointer increases speed and moves at a faster rate across the screen. This is called **mouse tracking.**

When you choose Fast Mouse Tracking from the Preferences menu, after you move the pointer two pixels in any direction, the pointer begins to move across the screen four times faster than the default speed. To return mouse tracking to its default setting, choose Slow Mouse Tracking. Follow these steps:

1. **Display the Preferences menu.**

   See "Displaying the Preferences Menu," earlier in this chapter.

2. **Choose the Fast Mouse Tracking or Slow Mouse Tracking command to change the mouse tracking speed.**

   When you release the RIGHT ARROW key, the Preferences menu disappears.

Setting the screen saver

The screen saver displays variations of the X logo pattern on the screen. The screen saver is activated when your computer is left idle, and it protects your screen from burn-in.

By default, the X11 screen saver takes control of the screen after 500 seconds (8 minutes, 20 seconds) of inactivity and leaves each screen-saver pattern on the screen for 500 seconds before displaying a new pattern.

When you choose Fast Screen Saver, the screen saver takes control of the screen after 50 seconds of inactivity and leaves each screen-saver pattern on the screen for 50 seconds before displaying a new pattern. To return the screen saver to the default setting, choose Slow Screen Saver. Follow these steps:

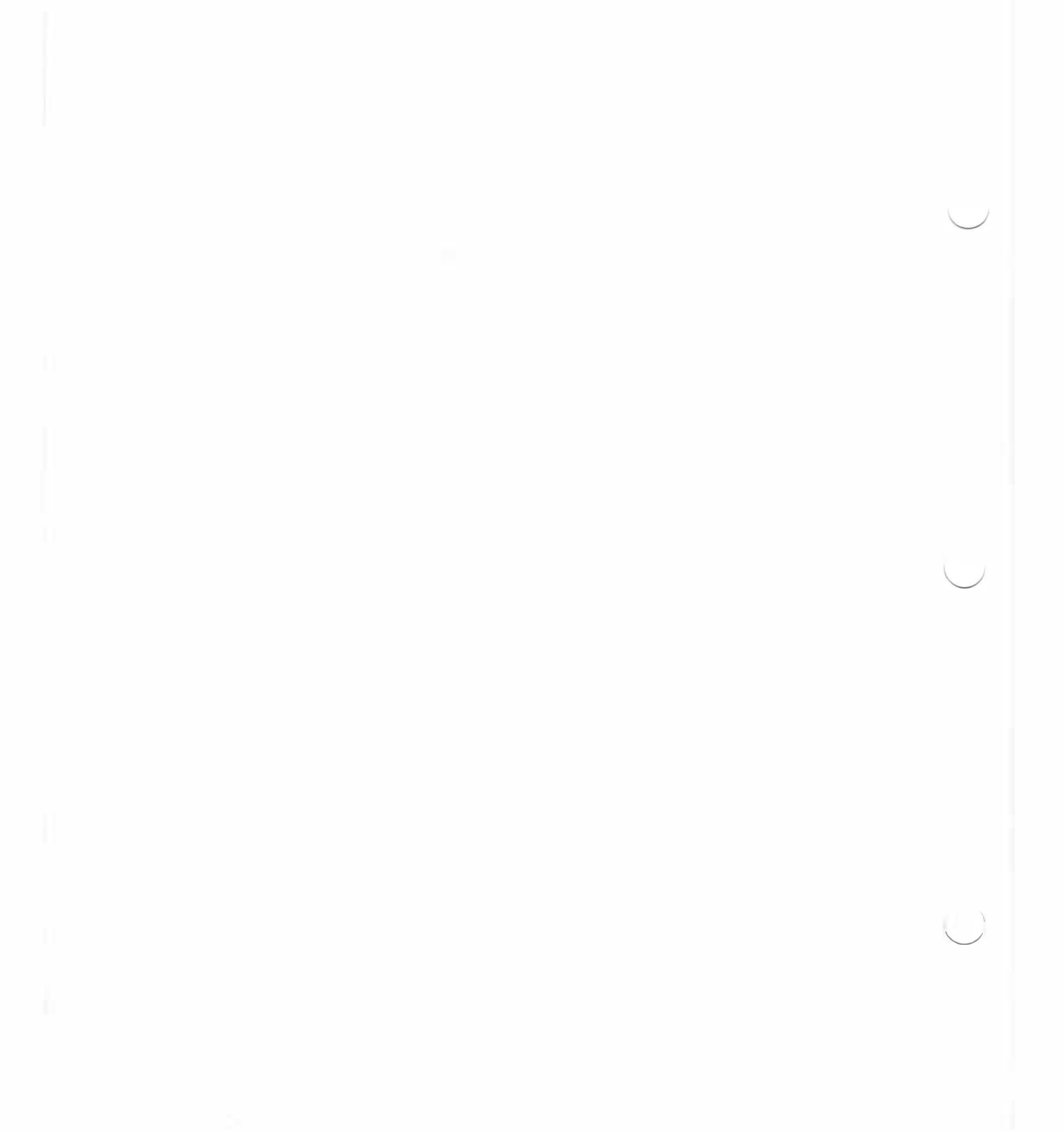1. **Display the Preferences menu.**

   See "Displaying the Preferences Menu," earlier in this chapter.

2. **Choose the Fast Screen Saver or Slow Screen Saver command to change the amount of time before which the screen saver becomes active.**

   When you release the RIGHT ARROW key, the Preferences menu disappears.

The next chapter shows you how to use the xterm terminal emulator.

Chapter 4 **Using xterm**

This chapter describes the xterm client application, the default terminal emulator for X11. xterm creates a window that you can use to open other X client applications and to run A/UX commands.

xterm provides two terminal emulation modes. In VT102 mode, the default setting, xterm emulates a standard ASCII character terminal. In Tektronix mode, xterm emulates an older-generation graphics terminal, letting you display graphics applications that were produced on a Tektronix 4014 terminal.

This chapter tells you how to use xterm and gives you an overview of basic xterm commands and operations. Topics include

- using menus to adjust terminal settings
- using the scroll bar
- copying text between xterm windows
- changing xterm defaults

Using `xterm` menus

`xterm` has four menus that you can use to control your `xterm` windows:

- Main Options menu
- VT Options menu
- VT Fonts menu
- Tek Options menu

The Main Options menu allows you to set options that apply to both VT102 and Tektronix windows. The VT Options menu and the Tek Options menu adjust terminal settings for VT102 and Tektronix windows, respectively. The VT Fonts menu allows you to modify the fonts used in a VT102 window.

When you display an `xterm` menu, the menu choices you see apply only to the active `xterm` window (see "Making Windows Active," in Chapter 3). To choose menu items for a different window, you must first make the window active and then display the menu.

Unlike the `twm` window manager menus, `xterm` menus may not be customized. However, you can invoke many of the features displayed in the menus by specifying command-line flag options when you open `xterm`, or by specifying `xterm` resources. Then you can use a menu item to override or disable the feature. See "Changing `xterm` Defaults," later in this chapter, for more information.

Most of the `xterm` menus are divided into three sections by horizontal lines. The top section contains terminal operation modes that toggle on and off. The middle section contains terminal commands; choosing one of these commands performs a specific action. The bottom section contains commands that allow you to manipulate Textronix and VT102 windows; you can use the VT Options menu to switch between the two modes of windows.

Each `xterm` menu is described in one of the following sections.

Main Options menu

You can use the Main Options menu (see Figure 4-1) to set options that apply to both VT102 and Tektronix windows.

■ **Figure 4-1** The Main Options menu

```
┌─────────────────────────────┐
│       Main Options          │
├─────────────────────────────┤
│  Secure Keyboard            │
│  Allow SendEvents           │
│  Log to File                │
│  Redraw Window              │
├─────────────────────────────┤
│  Send STOP Signal           │
│  Send CONT Signal           │
│  Send INT Signal            │
│  Send HUP Signal            │
│  Send TERM Signal           │
│  Send KILL Signal           │
├─────────────────────────────┤
│  Quit                       │
└─────────────────────────────┘
```

To display the Main Options menu and choose a menu item, follow these steps:

1. **Position the pointer in an xterm window.**

2. **Hold down the CONTROL (or DOWN ARROW) key.**

3. **Hold down the mouse button.**

   The pointer changes to a menu pointer, and the Main Options menu appears.

   After the Main Options menu appears, you can release the CONTROL (or DOWN ARROW) key, but you must continue holding down the mouse button.

4. **Drag to the desired menu item.**

   The command is highlighted.

5. **Release the mouse button.**

   When you release the mouse button, the menu disappears and the chosen menu item is executed.

   ◆ *Note:* The menu items you choose apply only to the active window.

Main Options menu modes

The following list describes the menu items in the modes section of the Main Options menu. With one exception (Redraw Window), these menu items toggle on and off. A checkmark next to a menu item indicates that it is active.

Secure Keyboard   Directs keyboard input to the active `xterm` window, ensuring that what you type in the window remains private and secure. For example, if you run an application that asks you for a password, you can choose Secure Keyboard from the menu, enter your password, and only the active client application is able to read the data. Note that the Secure Keyboard mode is automatically disabled when the `xterm` window is changed to an icon or is otherwise unmapped.

Allow SendEvents   Instructs `xterm` to accept SendEvent key and button events from the X server. By default, this option is disabled and `xterm` does not accept such events. This option should be enabled only by experienced users.

Log to File   Records the commands entered and responses received in an `xterm` window and saves the information in a log file in your home directory. By default, the log file is named `XtermLog`. *PID*, where *PID* is a number that uniquely identifies the file.

Redraw Window   Redraws the contents of an `xterm` window.

◆   *Note:* You can also use the `xrefresh` client application or the Redraw Screen `twm` menu command to redraw an `xterm` window. See "Window Operations Menu" in Chapter 3 and `xrefresh`(1X) in *X11 Command Reference for A/UX* for more information.

Main Options menu commands

The following list describes the menu items in the commands section of the Main Options menu. You can use these menu items to send signals to the process group running under xterm, usually one of the A/UX shells (see signal(3) in *A/UX Programmer's Reference* for information about signals). With one exception (Send CONT Signal), choosing one of these commands terminates the active xterm window.
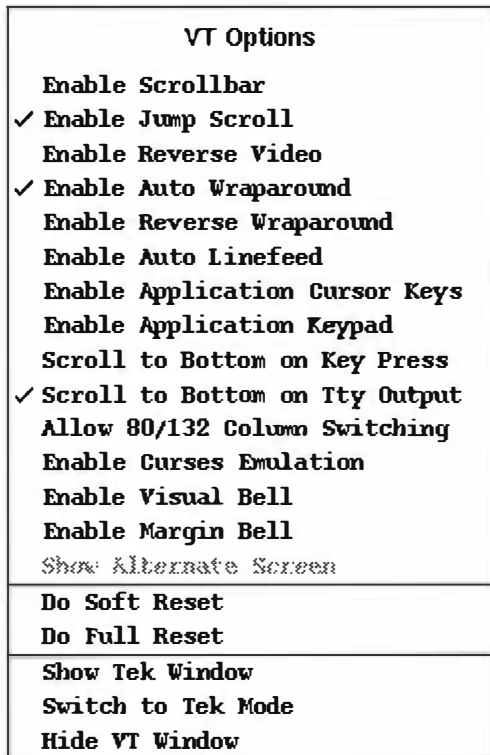
Send STOP Signal   Sends a SIGSTOP signal and suspends the current process.

Send CONT Signal   Sends a SIGCONT signal and causes a suspended or interrupted process to resume.

Send INT Signal   Sends a SIGINT signal and interrupts the current process.

Send HUP Signal   Sends a SIGHUP signal and causes the current process to disconnect.

Send TERM Signal   Sends a SIGTERM signal and terminates the current process.

Send KILL Signal   Sends a SIGKILL signal and causes an interrupt that removes the process completely.

Quit   Closes an xterm window. Note that choosing this command is the same as entering exit at the command prompt.

VT Options menu

You can use the VT Options menu (see Figure 4-2) to adjust terminal settings for VT102 windows.

■ **Figure 4-2** The VT Options menu

```
                VT Options
   Enable Scrollbar
 ✓ Enable Jump Scroll
   Enable Reverse Video
 ✓ Enable Auto Wraparound
   Enable Reverse Wraparound
   Enable Auto Linefeed
   Enable Application Cursor Keys
   Enable Application Keypad
   Scroll to Bottom on Key Press
 ✓ Scroll to Bottom on Tty Output
   Allow 80/132 Column Switching
   Enable Curses Emulation
   Enable Visual Bell
   Enable Margin Bell
   Show Alternate Screen
   Do Soft Reset
   Do Full Reset
   Show Tek Window
   Switch to Tek Mode
   Hide VT Window
```

To display the VT Options menu and choose a menu item, follow these steps:

1. **Position the pointer in an xterm VT102 window.**

2. **Hold down the CONTROL (or DOWN ARROW) key.**

3. **Hold down the LEFT ARROW key.**

   The pointer changes to a menu pointer, and the VT Options menu appears.

   After the VT Options menu appears, you can release the CONTROL (or DOWN ARROW) key, but you must continue holding down the LEFT ARROW key.

4. **While holding down the LEFT ARROW key, position the pointer on the desired menu item.**

   Make sure that you move the pointer to the desired menu item *without* pressing or holding down the mouse button.

5. **Release the LEFT ARROW key.**

   When you release the LEFT ARROW key, the VT Options menu disappears and the selected menu item is executed.

   ◆ *Note:* The menu items you choose apply only to the active window.

VT Options menu modes

The following list describes the menu items in the modes section of the VT Options menu. These menu items toggle on and off. A checkmark next to a menu item indicates that it is active. Unless otherwise indicated, the default setting for these menu items is off.

Enable Scrollbar   Places a scroll bar on the left side of a VT102 window. See "Using the Scroll Bar," later in this chapter, to find out how to use the scroll bar.

Enable Jump Scroll

    Allows text to scroll as fast as xterm receives it, producing a jump scroll rather than a smooth, one-line-at-a-time scroll. The default setting is on.

Enable Reverse Video

    Reverses the foreground and background colors.

Enable Auto Wraparound

    Allows you to continue typing when you reach the end of a line, automatically moving the insertion point to start the next line. The default setting is on.

Enable Reverse Wraparound

> Allows you to backspace from the beginning of one line to the end of the previous line.

Enable Auto Linefeed

> Adds an extra line when you press the RETURN key.

Enable Application Cursor Keys

> Enables the VT102 arrow keys to move the cursor.

> ◆ *Note:* Because X11 uses arrow keys as substitutes for certain mouse button presses, you must hold down the OPTION key and then press the desired arrow key if you want to use an arrow key explicitly as an arrow key.

Enable Application Keypad

> Enables the VT102 ten-key keypad on the keyboard.

Scroll to Bottom on Key Press

> Repositions the cursor at the bottom of the scroll region when a key is pressed. This menu item is only available when a scroll bar is present.

Scroll to Bottom on Tty Output

> Repositions the cursor at the bottom of the scroll region when output is displayed in a window. This menu item is only available when a scroll bar is present.

Allow 80/132 Column Switching

> Switches between 80 columns and 132 columns on the screen and resizes the window accordingly. The default is 80 columns.

Enable Curses Emulation

> Causes xterm to work around the curses bug that causes the more program to display lines incorrectly if the lines are the exact width of the window and are followed by a line that begins with a tab character (the leading tabs are not displayed). See more(1) in *A/UX Command Reference* and curses(3) in *A/UX Programmer's Reference* for more information.

Enable Visual Bell   Causes the window to flash instead of sounding the audible system bell.

Enable Margin Bell Rings a bell when you type near the end of a line.

Show Alternate Screen

> Restores the contents of a VT102 window as they were before you opened and closed a text editor. For example, if you use vi(1) and then close it, the previous contents of the screen, including messages and commands that were previously on the screen, are restored.

VT Options menu commands

The following list describes the menu items in the commands section of the VT Options menu. Choosing one of these commands performs the designated action in the active VT102 window.

Do Soft Reset   Resets the scroll region from a portion of the window to the full window. It is most useful when a program resets the scroll region incorrectly.

Do Full Reset   Clears the screen, resets tabs to every eight columns, and after xterm finishes processing a command, resets terminal modes to initial states.

Show Tek Window

> Displays a Tektronix window that is invisible (see Hide Tek Window, in the section "Tek Options Menu," later in this chapter, for information about making a Tektronix window invisible).

Switch to Tek Mode

> Displays a Tektronix window.

Hide VT Window   Makes a VT102 window invisible. The window is not destroyed but stays invisible until you choose Show VT Window from the Tek Options menu (described later in this chapter).

◆ *Note:* If you switch between VT102 mode and Tektronix mode, you may need to change the setting of your TERM environment variable. See the manual page for your default shell in *A/UX Command Reference* for more information.

VT Fonts menu

You can use the VT Fonts menu (see Figure 4-3) to modify the size of the fonts used in the active `xterm` VT102 window.

■ **Figure 4-3** The VT Fonts menu

```
┌──────────────────────┐
│      VT Fonts        │
│ ✓ Default            │
│   Tiny               │
│   Small              │
│   Medium             │
│   Large              │
│   Escape Sequence    │
│   Selection          │
└──────────────────────┘
```
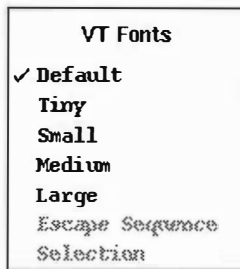
To display the VT Fonts menu and choose a menu item, follow these steps:

1. **Position the pointer in an `xterm` VT102 window.**

2. **Hold down the CONTROL (or DOWN ARROW) key.**

3. **Hold down the RIGHT ARROW key.**

   The pointer changes to a menu pointer, and the VT Fonts menu appears.

   After the VT Fonts menu appears, you can release the CONTROL (or DOWN ARROW) key, but you must continue holding down the RIGHT ARROW key.

4. **While holding down the RIGHT ARROW key, position the pointer on the desired menu item.**

   Make sure that you move the pointer to the desired menu item *without* pressing or holding down the mouse button.

5. **Release the RIGHT ARROW key.**

   When you release the RIGHT ARROW key, the VT Fonts menu disappears and the selected menu item is executed.

   ◆ *Note:* The menu items you choose apply only to the active window.

VT Fonts menu modes

The following list describes the menu items in the modes section of the VT Fonts menu. These menu items toggle on and off. A checkmark next to a menu item indicates that it is active; only one item is active at a time. Unless otherwise indicated, the default setting for these menu items is off.

Default
: Sets the font size currently being used in the active VT102 window to the default size specified in your `/usr/lib/X11/app-defaults/XTerm` resource file.

Tiny
: Sets the font size currently being used in the active VT102 window to a very small size and reduces the window to accommodate the smaller font size.

Small
: Sets the font size currently being used in the active VT102 window to a small size and reduces the window to accommodate the smaller font size.

Medium
: Sets the font size currently being used in the active VT102 window to a medium size and resizes the window to accommodate the new font size.

Large
: Sets the font size currently being used in the active VT102 window to a large size and enlarges the window to accommodate the larger font size.

The characteristics associated with the items in the VT Fonts menu, such as point size and font family, are determined by corresponding resources in the resource file `/usr/lib/X11/app-defaults/XTerm`. See xterm(1X) in *X11 Command Reference for A/UX* for information about setting resources for the VT Fonts menu.

VT Fonts menu commands

The following list describes the menu commands in the VT Fonts menu. Choosing one of these commands performs the designated action in the active VT102 window.

Escape Sequence   Sets the font size currently being used in the active VT102 window to the font identified by an OSC 50 Escape Sequence that has been sent to the window. (If the window has not received such an Escape Sequence, this menu item is disabled and appears dimmed in the menu.)

Selection   Interprets the value of the PRIMARY selection as the name of a font. That font is then used for the active VT102 window, and the window is resized accordingly.

Tek Options menu

You can use the Tek Options menu (see Figure 4-4) to adjust terminal settings for Tektronix windows.

■ **Figure 4-4** The Tek Options menu

```
┌─────────────────────────┐
│      Tek Options        │
│  Large Characters       │
│  #2 Size Characters     │
│✓ #3 Size Characters     │
│  Small Characters       │
├─────────────────────────┤
│  PAGE                   │
│  RESET                  │
│  COPY                   │
├─────────────────────────┤
│  Show VT Window         │
│  Switch to VT Mode      │
│  Hide Tek Window        │
└─────────────────────────┘
```
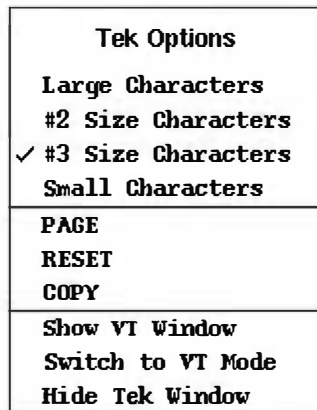
To display the Tek Options menu and choose a menu item, follow these steps:

1. **Position the pointer in an `xterm` Tektronix window.**

2. **Hold down the CONTROL (or DOWN ARROW) key.**

3. **Hold down the LEFT ARROW key.**

   The pointer changes to a menu pointer, and the Tek Options menu appears.

   After the Tek Options menu appears, you can release the CONTROL (or DOWN ARROW) key, but you must continue holding down the LEFT ARROW key.

4. **While holding down the LEFT ARROW key, position the pointer on the desired menu item.**

   Make sure that you move the pointer to the desired menu item *without* pressing or holding down the mouse button.

5. **Release the LEFT ARROW key.**

   When you release the LEFT ARROW key, the Tek Options menu disappears and the selected menu item is activated.

   ◆ *Note:* The menu items you choose apply only to the active window.

Tek Options menu modes

The following list describes the menu items in the modes section of the Tek Options menu. These menu items toggle on and off. A checkmark next to an item indicates that it is active. Unless otherwise indicated, the default setting for these menu items is off.

Large Characters — Uses the largest character set available in Tektronix mode. This is the default character set.

#2 Size Characters — Uses a smaller character set than Large Characters.

#3 Size Characters — Uses a smaller character set than #2 Size Characters.

Small Characters — Uses the smallest character set available in Tektronix mode.

◆ *Note:* Large Characters, #2 Size Characters, #3 Size Characters, and Small Characters are mutually exclusive. You may choose only one of these menu items at a time.

Tek Options menu commands

The following list describes the menu items in the commands section of the Tek Options menu. Choosing one of these commands performs the designated action in the active Tektronix window.

PAGE   Clears the Tektronix window.

RESET   Resets the Tektronix terminal, emulating an actual Tektronix reset button.

COPY   Places any Tektronix plotting commands needed to reproduce the current screen in a file named COPY-*yy-mm-dd.hh.mm.ss*.

Show VT Window   Displays the VT102 window you made invisible by choosing Hide VT Window from the VT Options menu.

Switch to VT Mode
                        Switches to VT102 mode.

Hide Tek Window
                        Makes the Tektronix window invisible. The window is not destroyed but stays invisible until you choose Show Tek Window from the VT Options menu.

◆ *Note:* If you switch between Tektronix mode and VT102 mode, you may need to change the setting of your TERM environment variable. See the manual page for your default shell in *A/UX Command Reference* for more information.

Using the scroll bar

You can display your VT102 windows with a scroll bar (see Figure 4-5). By default, the scroll bar is placed on the left side of the window, and 64 lines of text are saved.

■ **Figure 4-5** An xterm window with a scroll bar
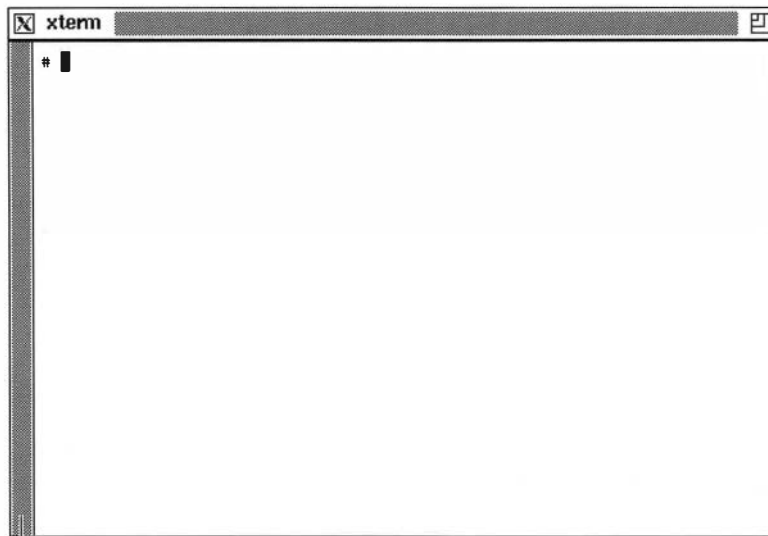
Scroll bar

There are several ways to display an xterm window with a scroll bar (see Figure 4-5). You can

■ choose Enable Scrollbar from the VT Options menu

■ open an xterm window with the command

```
xterm -sb &
```

- specify this resource string in a resource file:

```
xterm*scrollBar: True
```

◆ *Note:* You can change the number of lines saved for scrolling. See the `saveLines` resource and the `-sl` flag option in `xterm`(1X) in *X11 Command Reference for A/UX* for details.

The `xterm` scroll bar is a gray bar that slides within a rectangular scrolling region. The scrolling region is the same length as the window. The scroll bar changes in length within the scrolling region depending on how much text is visible in the window relative to the amount of text that has been saved. For example, when the scroll bar first appears, it fills the entire scrolling region. As more text is saved, the scroll bar decreases in length within the scroll bar region.

When you position the pointer in the scrolling region, the pointer changes to an arrow that points up or down, depending on the direction in which scrolling can occur. If you position the pointer somewhere in the middle of the saved text, the pointer appears as a two-headed arrow, indicating that you can scroll in either direction.

The following sections show you how to use the scroll bar.

Scrolling to the beginning of the saved text

1. **Position the pointer in the scrolling region.**

2. **Press the RIGHT ARROW key.**

    The text in the window scrolls down, toward the beginning of the saved text.

Scrolling to the end of the saved text

- **Position the pointer in the scrolling region and click it.**

    The text in the window scrolls up, toward the end of the saved text.

Scrolling to a specific place in the saved text

1. **Position the pointer in the scrolling region.**

2. **Press the LEFT ARROW key.**

   The scroll bar moves to the position of the pointer, and the text in the window scrolls to the corresponding position in the saved text.

Dragging the scroll bar within the scrolling region

1. **Position the pointer in the scrolling region.**

2. **Hold down the LEFT ARROW key.**

3. **Drag the scroll bar to the desired position in the scrolling region.**

   When you drag the scroll bar up, the text in the window scrolls down, toward the beginning of the saved text. When you drag the scroll bar down, the text in the window scrolls up, toward the end of the saved text.

Copying text between windows

With xterm you can select, copy, and paste text within a single VT102 window or between two VT102 windows. By cutting and pasting, you can build new command lines, share text between documents, or take output from a program and insert it in a document.

To select text for copying and then paste it, follow these steps:

1. **Position the pointer at the beginning of the text selection.**

2. **Drag until you reach the end of your selection.**

   The text becomes highlighted as you drag through it.

As a convenience, you can also double-click to select a word and triple-click to select a line. You can extend a selection by positioning the pointer where you want the selection extended to and then pressing the RIGHT ARROW key. You can click anywhere in the text to deselect the highlighted text.

◆ *Note:* Even when the text is no longer highlighted, the selection is retained in memory. It remains available (for pasting) until you make another selection.

3. **Position the pointer where you want to paste the text.**

   If you want to paste the text in a different VT102 window, position the pointer in that window.

4. **Press the LEFT ARROW key.**

   The text is pasted in the new location.

◆ *Note:* You can also use the `xclipboard` and `xcutsel` client applications to copy and paste text between client application windows. See `xclipboard`(1X) and `xcutsel`(1X) in *X11 Command Reference for A/UX* for more information.

Changing `xterm` defaults

You can specify many of the menu items on the `xterm` menus by specifying new values for `xterm` defaults in resource files or as flag options when you open an `xterm` window. This permits you to set defaults that may not otherwise be in effect during a work session. Then, where appropriate, you can use the `xterm` menus to override these defaults.

Features you can customize include window size and position, fonts used to display text, scrolling properties, bell properties, and icon size.

To specify default settings that go into effect each time you use `xterm`, use the resource manager (`xrdb`) or an X defaults file to change `xterm` resource values (see "Working With Resources" in Chapter 5 for details). These settings are then in effect each time you open an `xterm` window.

To specify options for a particular xterm window, use a flag option on the command line when opening the xterm client application.

For more information about specifying xterm resources and flag options, see xterm(1X) in *X11 Command Reference for A/UX*.

In the next chapter you'll learn more about using flag options and resources to customize client applications.

Chapter 5 **Customizing X11**

You can customize many aspects of the X11 environment and user interface to suit your individual needs and preferences. This chapter shows you how to

- modify the window-manipulation techniques, menus, and mouse-button assignments of the t wm window manager

- use command-line options and X resources to modify the behavior and appearance of client applications

- change the appearance of the root window with the xsetroot client application

- set preferences for the screen, keyboard, and mouse with the xset client application

- limit user access to the X server with the xhost client application

▲ **Warning**   This chapter is for advanced users. If you are a beginner, it is recommended that you become familiar with the default configuration before trying to customize X11. ▲

Customizing `twm`

The `twm` window manager defines many aspects of the X11 user interface. You can modify this user interface in a number of ways. You can modify the style of the default window (see Figure 3-1), change the fonts used to display text in windows and menus, use different pointer shapes for window-manipulation operations, change the contents of the default menus, define new menus, and so on.

Each of the features, techniques, and menus described in Chapter 3, "Using `twm`," is controlled by entries in the `twm` configuration file, `.twmrc`, and may be changed according to your needs. You can change the default configuration by changing the entries in your `.twmrc` file and restarting X11.

Two sample window manager configuration files are included with X11 in the `/usr/lib/X11` directory.

■ `/usr/lib/X11/Sample.twmrc` is the default `.twmrc` file. When you install the X11 software, a copy of this file is put in your home directory and is named `.twmrc` (see *X11 Installation Guide for A/UX* for installation information). When you start X11, the `twm` window manager looks for this file in your home directory and uses the entries in this file to create your environment.

■ `/usr/lib/X11/Expert.twmrc` is similar to the file `Sample.twmrc`, but includes an icon manager window. You can use the icon manager window to keep track of iconified windows and eliminate the clutter of icons on your screen (see "Using the Icon Manager," later in this chapter, for details). To use `Expert.twmrc` as your `.twmrc` file and invoke the icon manager, copy the contents of `Expert.twmrc` to your `.twmrc` file and restart X11.

△ **Important** Before you use `Expert.twmrc` or make any changes to your `.twmrc` file, make sure you're familiar with the default configuration described in Chapter 3, "Using `twm`." △

The `.twmrc` file

When you modify your `.twmrc` file, follow these steps:

1. **Make a backup copy of your `.twmrc` file.**

   You can use this command:

   ```
   cp .twmrc .twmrc.bak
   ```

2. **Make the desired changes to `.twmrc` by using an A/UX text editor such as `vi(1)` to edit the file.**

3. **Restart X11.**

Structure and syntax

The `.twmrc` file has three sections: a window-variables section, a mouse-button section, and a menu section.

■ The window-variables section specifies variables that set such things as fonts, system bell volume, and screen-saver behavior. Window-variable specifications are in the first part of the `.twmrc` file.

■ The mouse-button section specifies the keyboard keys and mouse-button presses that you use to invoke window manager operations. This includes the methods for displaying menus and moving windows. Mouse-button specifications are in the second part of the `.twmrc` file, following the window-variables section.

■ The menu section specifies the contents of the menus. Menu specifications are in the last part of the `.twmrc` file, following the mouse-button section.

△ **Important**   Specifications in the `.twmrc` file must be entered in the order indicated. △

Entries in .twmrc files follow these syntax rules:

■ Comments are preceded by a number symbol:

```
# This is a comment
```

■ Strings of characters are enclosed in double quotation marks:

"*string*"

■ A list of one or more variables is enclosed in braces:

```
{
list-item1
list-item2

}
```

■ Unless otherwise noted, variables and keywords are case sensitive.

Each part of the .twmrc file is discussed in detail in one of the following sections.

Default **.twmrc** file

Before trying to change your twm configuration, it's helpful to study the defaults specified in Sample.twmrc. The following is a complete listing of Sample.twmrc.

```
#******************************************
#   Sample.twmrc
#******************************************

AutoRaise
{
}

BorderWidth 2

Color
{
  BorderColor "black"
  IconForeground "black"
  IconBackground "cadetblue"
  IconBorderColor "black"
```

```
    MenuForeground "black"
    MenuBackground "wheat"
    MenuTitleForeground "black"
    MenuTitleBackground "wheat"
    TitleForeground "white"
    TitleBackground "cadetblue"
}

Monochrome
{
    MenuTitleForeground "white"
    MenuTitleBackground "black"
}

ForceIcons

IconDirectory      "~/icons"
Icons
{
 "xterm"  "xterm.icon" # obtained from /usr/include/X11/bitmaps
 "xfd"    "xfd_icon"  # obtained from /usr/include/X11/bitmaps
}

IconFont  "variable"

MenuFont  "variable"

NoTitle
{
 "xclock"
 "xload"
 "xlogo"
}

NoHighlight
{
    "xclock"
```

```
    "xload"
    "xlogo"
}


NoSaveUnder

NoTitleFocus

ResizeFont    "vgb-25"
TitleFont     "variable"
#WarpCursor
DefaultFunction f.menu "default-menu"


#Button = KEYS : CONTEXT : FUNCTION
#---------------------------------
Button1 =    : root   : f.menu "button1"
Button2 =    : root   : f.menu "button2"
Button3 =    : root   : f.menu "button3"
Button1 = m  : window : f.menu "buttonl"
Button2 = m  : window : f.menu "button2"
Button3 = m  : window : f.menu "button3"
Button1 = m  : title  : f.menu "button1"
Button2 = m  : title  : f.menu "button2"
Button3 = m  : title  : f.menu "button3"
Button1 =    : title  : f.function "move-or-raise"
Button2 =    : title  : f.raiselower
Button3 =    : title  : f.lower
Button1 =    : icon   : f.function "deiconify-raise"
Button2 =    : icon   : f.move
Button3 =    : icon   : f.lower


Function "move-or-raise"
{
  f.move
  f.deltastop
  f.raise
}
```

```
Function "deiconify-raise"
{
  f.deiconify
  f.raise
}

menu "button1"
{
 "Window Operations"   f.title
 "Redraw Screen"       f.refresh
 "Source .twmrc"       f.twmrc
 "twm Version"         f.version
 "(De)Iconify"         f.iconify
 "Move Window"         f.move
 "Resize Window"       f.resize
 "Raise Window"        f.raise
 "Lower Window"        f.lower
 "Destroy Window"      f.destroy
 "Kill twm"         f.quit
 "Mouse Button Help"   f.menu "help"
}

menu "default-menu"
{
 "Default Menu"     f.title
 "Redraw Screen"    f.refresh
 "Source .twmrc"    f.twmrc
 "twm Version"      f.version
 "(De)Iconify"      f.iconify
 "Move Window"      f.move
 "Resize Window"    f.resize
 "Raise Window"     f.raise
 "Lower Window"     f.lower
 "Destroy Window"   f.destroy
```

```
 "Kill twm"         f.quit
 "Mouse Button Help" f.menu "help"
}

menu "button2"
{
 "New xterms"  f.title
 "xterm 24 lines" !"xterm -T `hostname` -geometry 80x24 &"
 "xterm 30 lines" !"xterm -T `hostname` -geometry 80x30 &"
 "xterm 12 lines" !"xterm -T `hostname` -geometry 80x12 &"
}

menu "button3"
{
 "Preferences"        f.title
 "Autorepeat On"       !"xset r on &"
 "Autorepeat Off"      !"xset r off &"
 "Fast Mouse Tracking"  !"xset m 4 2 &"
 "Slow Mouse Tracking"  !"xset m &"
 "Fast Screen Saver"    !"xset s 50 50 &"
 "Slow Screen Saver"    !"xset s 500 500 &"
}

menu "help"
{
 "Mouse Buttons"            f.title
 "Left Button = Mouse Button"   f.nop
 "Middle Button = Left Arrow"   f.nop
 "Right Button = Right Arrow"   f.nop
 "Meta = Open Apple/Up Arrow"   f.nop
 "Control = Control/Down Arrow"  f.nop
}
```

Specifying window variables

Window variables include items such as border width, window background colors, and the fonts used to display text in title bars, icons, and menus. Window variables are initialized once, when twm first starts up.

◆ *Note:* You must enter window-variable specifications in the first part of the .twmrc file.

The following is an alphabetical list of the most commonly used twm window variables.

AutoRaise (*list*)   Lists window names that automatically move to the top of a stack of overlapping windows when the pointer enters the window's borders. The names in the list are the first characters the system looks for in a window name. For example, specifying

```
AutoRaise
{
 "xterm"
 "bitmap"
}
```

moves any window whose name starts with xterm or bitmap to the top of the stack when the pointer resides within its borders.

BorderColor "*string*"
   Sets the color of window borders. You must specify this variable in a Color or Monochrome variable list.

BorderWidth *pixels*
   Sets the width, in pixels, of window borders.

`Color { `*`list`*`}`   Lists color assignments for color display devices. Note that you may assign colors to specific elements in a window. For example, specifying

```
Color
{
 BorderColor "red"
 TitleForeground "yellow"
 TitleBackground "blue"
}
```

adorns windows with red borders and draws their title bars with yellow letters on a blue background.

`DontMoveOff`   Prevents windows from being moved off the screen.

`ForceIcons`   Causes the bitmaps in the `Icons` variable list to override any icons provided by client applications.

`IconBackground "`*`string`*`"`
Sets the background color of icons. You must specify this variable in a `Color` or `Monochrome` variable list.

`IconBorderColor "`*`string`*`"`
Sets the color of icon borders. You must specify this variable in a `Color` or `Monochrome` variable list.

`IconDirectory "`*`string`*`"`
Specifies the directory to be searched for icon bitmap files. If this variable is set, `twm` looks in the specified directory for the bitmap file. If that file doesn't exist or if the `IconDirectory` variable isn't set, the default `/usr/include/X11/bitmaps` directory is searched.

`IconFont "`*`string`*`"`
Specifies the font used to display a client application's name in a window icon.

`IconForeground "`*`string`*`"`
Sets the foreground color of window icons. You must specify this variable in a `Color` or `Monochrome` variable list.

Icons { *list* }   Lists window names and bitmap files to be used as icons when the iconify operation is invoked. For example,

```
Icons
{
 "xbiff" "xbiff_icon"
}
```

iconifies windows named `xbiff` by using the bitmap pattern specified in the file `xbiff_icon`.

MenuBackground "*string*"

Sets the background color of a menu. You must specify this variable in a `Color` or `Monochrome` variable list.

MenuFont "*string*"

Specifies the font used in menus.

MenuForeground "*string*"

Sets the foreground color of menus. You must specify this variable in a `Color` or `Monochrome` variable list.

MenuShadowColor "*string*"

Sets the color for menu drop shadows. You must specify this variable in a `Color` or `Monochrome` variable list.

MenuTitleBackground "*string*"

Sets the background color of menu titles. You must specify this variable in a `Color` or `Monochrome` variable list.

MenuTitleForeground "*string*"

Sets the foreground color of menu titles. You must specify this variable in a `Color` or `Monochrome` variable list.

`Monochrome {` *list* `}`

> Lists color assignments for monochrome screens. For example,

```
Monochrome
{
 BorderColor "black"
 TitleForeground "black"
 TitleBackground "white"
}
```

> sets the window border and title-bar foreground to black, and the title-bar background to white.

`NoRaiseOnDeiconify`

> Specifies that windows are not to be moved to the top of the stack of overlapping windows when they are deiconified.

`NoRaiseOnMove`

> Specifies that windows are not to be moved to the top of the stack of overlapping windows following a move operation.

`NoRaiseOnResize`

> Specifies that windows are not to be moved to the top of a stack of overlapping windows following a resize operation.

`NoTitle {` *list* `}`   Lists windows that do not have title bars. For example, specifying

```
NoTitle
{
 "xclock"
 "xlogo"
 "xload"
}
```

> draws windows without title bars for windows whose names begin with `xclock`, `xlogo`, or `xload`.

`NoTitleFocus`   Specifies that input focus is not to be directed to a window when the pointer is in its title bar.

`ResizeFont "`*string*`"`
> Specifies the font to be displayed in the window-dimensions pop-up window at the top-left of the screen during window resize operations.

`ReverseVideo` Reverses a window's background and foreground colors.

`TitleBackground "`*string*`"`
> Sets the background color of title bars. You must specify this variable in a `Color` or `Monochrome` variable list.

`TitleFont "`*string*`"`
> Specifies the font used to display text in title bars. Note that because the default title bar is 17 pixels high, the font you choose for window titles may not exceed 15 pixels in height.

`TitleForeground "`*string*`"`
> Sets the foreground color of title bars. You must specify this variable in a `Color` or `Monochrome` variable list.

`UnknownIcon "`*string*`"`
> Specifies the filename of a bitmap file to be used for the default icon. This bitmap is used as the icon for any client application that does not provide its own icon bitmap.

`WarpCursor` Causes the pointer to jump from a window icon to the middle of the window after it is deiconified.

`Zoom` Draws a series of window outlines when a window is changed to or from an icon.

For more information about window variables, see `twm`(1X) in *X11 Command Reference for A/UX*.

Specifying mouse-button functions

You use mouse-button functions to define the action performed when a mouse button or an equivalent key is pressed in a certain location on the screen, such as a client application's window, the root window, an icon, or a title bar. (See "Window Manager Functions," later in this chapter, for a complete list of the available twm window manager functions.)

◆ *Note:* You must enter mouse-button specifications in the second part of the .twmrc file, following the window-variables section.

Mouse-button functions are assigned using this format:

Button*n* = *keys* : *context* : *function*

These keywords are defined as follows:

- The placeholder Button*n* must be replaced by Button1, Button2, or Button3 on a three-button mouse. The three-button mouse equivalents for using X11 on a Macintosh are as follows:
  - □ Button1 is the mouse button.
  - □ Button2 is the LEFT ARROW key.
  - □ Button3 is the RIGHT ARROW key.
- The placeholder *keys* specifies modifier keys. These values are as follows:
  - □ s is for SHIFT.
  - □ c is for CONTROL (or DOWN ARROW).
  - □ m is for Meta (COMMAND or UP ARROW).
- The placeholder *context* determines what type of function is invoked according to where a specified Button*n* (Button1, Button2, or Button3) is pressed:
  - □ Functions in icon context are invoked when a button is pressed while the pointer is within the boundaries of an icon.
  - □ Functions in root context are invoked when Button*n* is pressed while the pointer is in the root window.
  - □ Functions in title context are invoked when Button*n* is pressed while the pointer is in a window's title bar.

□ Functions in `window` context are invoked when `Button`*n* is pressed while the pointer is in a window.

□ Functions in `iconmgr` context are invoked only when the icon manager has been specified (see "Using the Icon Manager," later in this chapter).

■ The placeholder *function* specifies the window manager action to be taken when the appropriate button (and modifier, if any) is pressed in the specified context (see "Window Manager Functions," later in this chapter).

The following is a sample button-function specification:

```
Button1 =    : root : f.menu "pop-up1"
Button2 = m  : window: f.menu "pop-up2"
Button3 = m  : title : f.menu "pop-up3"
Button1 =    : title : f.move
Button1 =    : icon : f.move
```

In this example, the strings have the following meaning:

■ Line 1 specifies that when the mouse button (`Button1`) is pressed while the pointer is in the root window (`root`), `twm` displays the menu `pop-up1`.

■ Line 2 specifies that when the LEFT ARROW key (`Button2`) is pressed with the Meta (`m`) modifier (COMMAND or UP ARROW) while the pointer is in any window (`window`), `twm` displays the menu `pop-up2`.

■ Line 3 specifies that when the RIGHT ARROW key (`Button3`) is pressed with the Meta (`m`) modifier (COMMAND or UP ARROW) while the pointer is in a title bar (`title`), `twm` displays the menu `pop-up3`.

■ Line 4 specifies that when the mouse button (`Button1`) is pressed while the pointer is in a title bar (`title`), `twm` executes the `f.move` function for that window.

■ Line 5 specifies that when the mouse button (`Button1`) is pressed while the pointer is on an icon (`icon`), `twm` executes the `f.move` function for that icon.

For more information about button functions, see `twm`(1X) in *X11 Command Reference for A/UX*.

Specifying menu functions

You use menu functions to define the contents of menus. The next section, "Window Manager Functions," has a complete list of the available twm window manager functions.

◆ *Note:* You must enter menu-function specifications in the last part of the .twmrc file, following the button-function section.

Menu functions are assigned using this format:

```
menu "menu name"
{
  string function
  string function
        .
        .
        .
  string function
}
```

These keywords are defined as follows:

- menu is a keyword that is read when twm starts up.
- The placeholder *menu name* is the menu title. It should be identical to the string in the f.menu function. Note that *menu name* is case sensitive and distinguishes between uppercase and lowercase characters.
- The placeholder *string* is the text that appears in a menu.
- The placeholder *function* is one of the entries described in the next section, "Window Manager Functions."

The following is a sample menu-function specification:

```
Button1 = : root  : f.menu "pop-up1"
        .
        .
        .
menu "pop-up1"
```

```
{
"Window Operations"        f.title
"Redraw Screen"            f.refresh
"Source .twmrc"            f.twmrc
"twm Version"              f.version
}
```

In this example, the strings have the following meanings:

- `Button1` is defined in the button-function section of the `.twmrc` file as `f.menu "pop-up1"`. This matches the menu keyword `pop-up1` in the menu section of `.twmrc`.

- `Window Operations` is designated as the title of the menu and is tied to the `f.title` function.

- `Redraw Screen` is the first menu command and is tied to the `f.refresh` function.

- `Source .twmrc` is the second menu command and is tied to the `f.twmrc` function.

- `twm Version` is the third menu command and is tied to the `f.version` function.

For more information about menu functions, see `twm`(1X) in *X11 Command Reference for A/UX*.

Window manager functions

The following is an alphabetical list of the `twm` window manager functions you use to define menus and mouse-button specifications. See the preceding sections "Specifying Mouse-Button Functions" and "Specifying Menu Functions" for information about how to assign these functions.

`f.circledown`   Moves the top window in a stack of overlapping windows to the bottom of the stack.

`f.circleup`   Moves the bottom window in a stack of overlapping windows to the top of the stack.

`f.cutfile`   Interprets the contents of the server's cut buffer as a filename, and reads the contents of that file into the server's cut buffer.

`f.deltastop`   Allows a user-defined function to be discontinued if the pointer has been moved more than *MoveDelta* pixels.

`f.destroy`	Removes a client application window. If invoked from a menu, the pointer changes to a skull-and-crossbones pointer, and the next window you click is destroyed.
`f.file` *"string"*	Reads *string* into the server's cut buffer.
`f.focus`	Assigns focus to the mouse so that keyboard input follows the mouse.
`f.iconify`	Iconifies/deiconifies a window. If invoked from a menu, the pointer changes to a circle pointer, and the next window (or icon) you click is changed to an icon (or window).
`f.lower`	Moves the selected window to the bottom of a stack of overlapping windows. If invoked from `title`, `window`, or `icon` context, the action is immediate. If invoked from `root` context or from a menu, the pointer changes to a circle pointer, and the next window you click is moved to the bottom of the stack.
`f.menu` *"string"*	Assigns *string* to the specified mouse button. If used as a menu entry, it assigns a submenu to the menu entry; the submenu is displayed when the appropriate menu item is chosen.
`f.move`	Displays a grid over the active window. This grid can then be dragged to another location. Double-clicking causes the pointer to jump to the top-left corner of the grid after the window is moved. If invoked from the title bar, the action is immediate. If invoked from the root window or from a menu, the pointer changes to a compass pointer, and the next window you click is moved.
`f.nop`	Specifies a null operation that can be used for menu commands that provide information, rather than perform an action.
`f.quit`	Exits `twm`.
`f.raise`	Moves the selected window to the top of a stack of overlapping windows. If invoked from `title`, `window`, or `icon` context, the action is immediate. If invoked from `root` context or from a menu, the pointer changes to a circle pointer, and the next window you click is moved to the top of the stack.

`f.refresh`	Redraws the contents of the windows on the screen.
`f.resize`	Resizes a window. If invoked from `title`, `window`, or `icon` context, the action is immediate. If invoked from `root` context or from a menu, the pointer changes to a compass pointer, and the next window you click is resized.
`f.source "file"`	Reads and processes the named *file* as the `twm` configuration file. This function is used to rebuild menus and redefine button operations; it does not change `twm` variables.
`f.title "string"`	Specifies a menu title that is centered and outlined with a border. This function can be used more than once in a menu.
`f.twmrc`	Specifies that the `.twmrc` file be reread. This function is useful when you make changes to `.twmrc` and you want them to take effect immediately. This function performs the same action as `f.source`, except that you do not have to specify the filename.
`f.unfocus`	Assigns focus to the root window and implements mouse focus rather than keyboard focus.
`f.version`	Displays the current version number of the `twm` window manager in the title bar of the `xterm` console window. The version number disappears when you press the mouse button, LEFT ARROW, or RIGHT ARROW, or when you move the pointer to another window.
`f.winrefresh`	Similar to the `f.refresh` function, but allows you to redraw the contents of a single window. The pointer changes to a circle pointer, and the next window you click is redrawn.

! string	Sends the specified string to an A/UX shell for execution, allowing you to invoke any client application or A/UX command from a menu or mouse button. For example,

```
!"xterm -geometry 80x24-10-10 &"
```

opens an `xterm` window that is 80 characters wide and 24 characters high, with the bottom-right corner of the window 10 pixels to the left and 10 pixels above the bottom-right corner of the screen. Note that the ampersand (`&`) at the end of the command line is mandatory.

^ string	Places the specified string and a newline character into the window's cut buffer.

Using the icon manager

The icon manager displays the TWM Icon Manager window, which lists open and iconified client applications. When you use the icon manager, `twm` no longer displays icons to represent iconified windows. Instead, when you iconify a window, the icon manager simply places a mark (an iconify box) next to the client application's name in the TWM Icon Manager window. You use this window to keep track of iconified client applications and eliminate the clutter of icons on your screen.

To display the TWM Icon Manager window, use `Expert.twmrc` as your `.twmrc` file, and restart X11. The window appears in the top-right corner of your screen.

◆ *Note:* For more information about setting up your `.twmrc` file, see the section "Setting Up X11" in Chapter 2 of *X11 Installation Guide for A/UX*.

Figure 5-1 shows the TWM Icon Manager window. Several default startup client applications are listed in the window, and as new client applications are opened, their names are added to the list. The window increases in size, as necessary, to display the names.

■ **Figure 5-1**   The TWM Icon Manager window
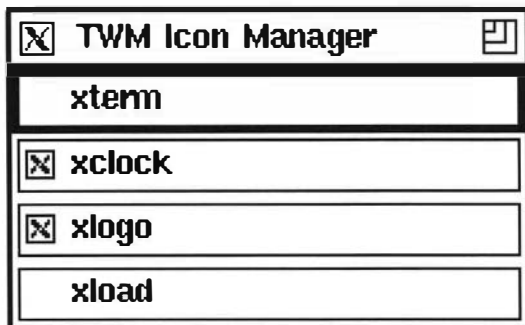
To iconify a client application window, position the pointer on the client application's name in the TWM Icon Manager window and click. The client application window disappears, and an iconify box, representing the iconified window, is placed next to the client application's name. In Figure 5-1, xclock and xlogo are iconified.

To deiconify a client application window, click the client application's name again. The iconify box is removed, and the window returns to its full size and original position on the screen. The name of the client application, without the iconify box, remains in the TWM Icon Manager window to show that the client application is open. In Figure 5-1, xterm and xload are open client applications that are not iconified.

Using command-line flag options

You can use command-line flag options to customize many aspects of a client application's behavior and appearance. With flag options you can specify window borders, background and foreground colors, window size, the fonts used to display text and menu items, and so on.

◆ *Note:* In most cases, the features you can modify with flag options may also be modified by setting resources. See "Working With Resources," later in this chapter, for details.

You can specify a flag option at the command line when you open a client application, or you can enter flag options in a startup file. For example, when you start X11, the startup sequence opens the `xterm`, `xclock`, `xload`, and `xlogo` client applications. This startup configuration is specified in the `.x11start` startup file (in your home directory) as follows:

```
xclock -analog -update 60 -geometry 96x96-2-2 &
xload -geometry 96x96-100-2 &
xlogo -fg grey -bg maroon -geometry 96x96-2-2 &
xterm -geometry 80x24+40+88 -title "`hostname` 1" &
xterm -geometry 80x24+40+88 -title "`hostname` 2" &
```

Notice that each command in the example specifies the `-geometry` flag option. The `-geometry` flag option, which specifies window size and positioning on the screen, is an X Toolkit option. X Toolkit options are a group of flag options that specify basic windowing features that are available to all client applications.

The following is a list of the most commonly used X Toolkit options:

`-bg` *color*   Specifies the color to be used for the window background.

`-display` *server*:0.*screen*
                     Specifies the display device on which to open a client application.

`-fg` *color*   Specifies the color to be used for the window foreground.

`-fn` *fontName*   Specifies a font name.

`-font` *fontName*   Specifies a font name.

`-geometry` *WxH+X+Y*
                     Specifies the size and positioning of a window, calculated from the top-left corner of the screen.

`-geometry` *WxH-X-Y*
                     Specifies the size and positioning of a window, calculated from the bottom-right corner of the screen.

`-rv`   Specifies reverse video, swapping the background and foreground colors.

`+rv`	Disables reverse video.
`-title` *name*	Specifies the string used as the window's title in the title bar.
`-xrm` *resource*	Specifies a resource name and value. This flag option is most commonly used to override defaults or to specify features that do not have explicit command-line options.

The following sections give details and examples about how to use some of these options. For more information about the X Toolkit options, see X(1X) and the individual client application's manual page in *X11 Command Reference for A/UX*.

◆ *Note:* Many client applications also have their own set of flag options, tailored to the client application's individual features and characteristics. For a list of these flag options, see the client application's manual page in *X11 Command Reference for A/UX*.

Specifying colors

If you have a color monitor, you can specify different colors for your windows. For example, to specify an `xterm` window with a white foreground and a blue background, use this command:

```
xterm -fg white -bg blue &
```

You specify a color with its common name, such as `blue`, `red`, `white`, or `black`. When a client application requests a color, the server searches the X11 color database (`/usr/lib/X11/rgb.txt`) and translates the color's name into the numeric value found in the color database.

See X(1X) in *X11 Command Reference for A/UX* for more information about specifying colors.

Specifying fonts

The `-fn` (or `-font`) flag option lets you specify fonts that can be used to display text in windows and menus. For example, to specify a font in an `xterm` window, use the command

`xterm -fn` *fontName* `&`

where *fontName* follows standard X11 font-naming conventions. See Chapter 6, "Working With Fonts and Graphics," for more information.

◆ *Note:* Use the `xlsfonts` client application to find out the names of the fonts available on your server. See Chapter 6, "Working With Fonts and Graphics," and `xlsfonts`(1X) in *X11 Command Reference for A/UX* for details.

For example, to specify the Courier 10 font in a client window, enter

client `-fn adobe-courier-medium-r-normal--10-100-75-75-m-60-iso8859-1`

Or to specify the same font in a resource file, enter

client`*font: adobe-courier-medium-r-normal--10-100-75-75-m-60-iso8859-1`

To eliminate the need to type long, complicated font names, you can abbreviate these names with wildcard characters. For example, using wildcard characters, you can specify the Courier 10 font in this way:

`'*-courier-medium-r-normal--10-*-75-*'`

◆ *Note:* Make sure to use single quotation marks to enclose a font name that uses wildcard characters—for example,

`xterm -fn '*-courier-medium-r-normal--10-*-75-*' &`

The X11 distribution provides font database files to assist the server in finding requested fonts. The font database file is a cross-referenced list consisting of the full name of the font and its filename. When the server receives a font request, it searches the database and then goes to the indicated file to find the font data. See Chapter 6, "Working With Fonts and Graphics," and `mkfontdir`(1X) in *X11 Command Reference for A/UX* for details about the X11 font database.

◆ *Note:* The xfontsel client application provides a point-and-click interface for viewing font files. You can examine a sample of a specific font, and copy and paste the string identifying the sample font into an application or resource file. See Chapter 6, "Working With Fonts and Graphics," for information about using xfontsel. For additional information, see xfontsel(1X) in *X11 Command Reference for A/UX.*

For more information about using fonts in the X11 environment, see X(1X) in *X11 Command Reference for A/UX.*

Specifying window geometry

Window geometry refers to the size and placement of a window on the screen. You can specify window geometry with the syntax

client -geometry *W*x*H*+*X*+*Y* &

where *W* is the width of the window, *H* is the height, and +*X*+*Y* specifies the pixel location of the top-left corner of the window relative to the top-left corner of the screen.

Some client applications measure *W*x*H* in characters, not pixels. For example,

xterm -geometry 80x24+40+20 &

specifies an xterm window that is 80 characters wide and 24 characters high, with its top-left corner placed 20 pixels to the right and 40 pixels below the top-left corner of the screen.

You can also use negative values to specify window placement. For example, this syntax

client -geometry *W*x*H*-*X*-*Y* &

specifies where the bottom-right corner of the window is placed relative to the bottom-right corner of the screen. For example,

xclock -geometry 96x96-2-2 &

specifies an xclock window that is 96 pixels wide and 96 pixels high, with its bottom-right corner placed 2 pixels to the left and 2 pixels above the bottom-right corner of the screen.

Specifying display devices

You can use the `-display` flag option to designate a specific screen on which a client application window will appear. Use this syntax:

client `-display` *server:* `0.` *screen* `&`

where *server* is the name of the local *or* remote computer running the X server, 0 designates the default screen, and *screen* is a number that corresponds to the relative position of the screen's video card among all the video cards installed in the NuBus expansion slots in the Macintosh.

For example, to open an `xterm` window on a screen controlled by the video card in relative position 3, enter this command:

```
xterm -display ventana:0.2 &
```

This command displays an `xterm` window on the specified screen of the computer named "ventana."

When *server* is the local server, you can simplify the syntax by omitting the server component of the argument. You can use this syntax:

client `-display` `:0.` *screen*

◆ *Note:* Facing the back of the Macintosh, video cards in NuBus slots are ordered right to left, beginning with 0. By default, all windows are drawn on screen 0.0, controlled by the first video card in the NuBus.

Specifying screen characteristics

The X11 startup sequence (see Chapter 2, "Getting Started") assumes that the screen(s) connected to the X server are monochrome screens. You can specify color characteristics for a screen by creating a file named `.X11` in your home directory. Each time you start X11, the startup sequence looks for this file for any special startup instructions.

For example, to specify screen characteristics for monochrome and color screens, enter this command line in .X11:

```
X -screen 0 -depth 8 -screen 1 -depth 1
```

This specifies that the screen controlled by the video card in relative position 1 in the NuBus (screen 0) operates at a color depth of 8 bits per pixel, indexing a color map of 256 entries, and the screen controlled by the video card in relative position 2 (screen 1) operates at a color depth of 1 bit per pixel, or monochrome.

◆ *Note:* The default display device is numbered 0, and the default color depth is the least depth supported by the video card (usually 1 bit per pixel).

You may also enter the x command at the command prompt in the A/UX System Console (console emulator window). This also starts the server and invokes the startup procedure. To start X11 using the x command, you must choose Console Emulator as your startup environment in the Change Session Type dialog box. See *A/UX Essentials,* Chapter 2, "Getting Started," and X(1X) in *X11 Command Reference for A/UX* for more information.

▲ **Warning**   Do not enter the x command in a CommandShell window in the A/UX Finder environment. ▲

Specifying resources on the command line

You can use the -xrm flag option to specify resources on the command line. For example, this command opens xclock with a red foreground:

```
xclock -xrm "Foreground: red" &
```

See the next section, "Working With Resources," and the client application's manual page in *X11 Command Reference for A/UX* for more information about resources.

Working with resources

A resource is a characteristic, such as a color, font, or window size, that affects the behavior or appearance of a client application. All client applications have certain predefined resource values. You can customize the behavior of client applications by assigning new values to these resources. Understanding the concepts of resource management greatly increases your ability to customize your configuration according to your needs.

◆ *Note:* For a complete list of the resources that can be set for a client application, see the client application's manual page in *X11 Command Reference for A/UX*.

X11 obtains resource values from the following sources:

- global resources stored in the server
- application-specific resources stored in the `/usr/lib/X11/app-defaults` directory
- user-specific resources stored in a file whose name is set in the XENVIRONMENT variable
- temporary resources specified at the command line with the `-xrm` flag option

Global resources

Global resources are stored directly on the server, where they are available to all client applications that communicate with the server.

The X protocol provides a mechanism that allows you to use the `xrdb` client application to load a global resource file into the server's resource database. When a client application needs the value of a resource, it searches this database for the needed resource. If the needed values are found in the resource database, these values override any predefined resource values.

◆ *Note:* In older versions of X Window System, global resources were stored in a file named .Xdefaults. For compatibility, if you do not use xrdb to load global resources in the server, client applications will look in your home directory for .Xdefaults. Using the .Xdefaults file is similar to using the resource database, but it is less flexible because client applications running on remote hosts may not have access to a local .Xdefaults file. When you use xrdb, resource values are available to client applications running on both local and remote computers.

Application-specific resources

Application-specific resources set defaults for client applications running on a particular host computer. Each client application may have a resource file in the /usr/lib/X11/app-defaults directory. When a client application needs the value of a resource, it searches this directory for a file with its name. For example, the pathname of an application-specific resource file for xterm is /usr/lib/X11/app-defaults/XTerm.

If the client application finds an application-specific resource file, it searches this file for the needed resource. If the needed values are found, these values override any predefined resource values and values specified in the resource database.

◆ *Note:* Because application-specific resource files reside in the /usr/lib directory, they affect all users of the host computer.

User-specific resources

User-specific resources are similar to application-specific resources, but they allow users to set defaults that won't affect other users. You can specify resource values for a client application in a file and then set the XENVIRONMENT variable to point to that file. When a client application needs the value of a resource, it searches the specified file for the needed resource. If the needed values are found, these values override any predefined resource values and values specified in the resource database and /usr/lib/X11/app-defaults directory.

Temporary resources

Temporary resources affect a single invocation of a client application. You use the -xrm command-line flag option to specify temporary resources. Resources specified at the command line override any previously set resource values.

Setting resources

Resource specifications follow this syntax:

clientApplication * *resource* : *value*

where *value* can be a string value (such as foreground: red), a numeric value (borderwidth: 1), or a Boolean value (scrollBar: on).

The following examples specify resource values for the xterm client application:

```
xterm*foreground: red
xterm*borderwidth: 1
xterm*scrollBar: on
```

◆ *Note:* For details about the syntax of resource specifications, see X(1X) in *X11 Command Reference for A/UX.*

Client applications determine resource values in the following order, with later values overriding earlier values:

- global resources
- application-specific resources
- user-specific resources
- temporary resources

To set global resources

1. **Create a file with an A/UX text editor and specify the desired resources.**

2. **Give the file a name that identifies it as an X resource file.**

   For example, you can use `.Xres` as the filename.

3. **Use the xrdb client application to load the resource file into the server.**

   For example, use this command:

   ```
   xrdb -load .Xres
   ```

Assume the following is part of a global resource file:

```
XTerm*scrollBar: on
```

Unless overriding values have been set, whenever `xterm` opens on this server, it will display a scroll bar.

For more information about `xrdb`, see `xrdb`(1X) in *X11 Command Reference for A/UX*.

To set application-specific resources

1. **Create a file with an A/UX text editor and specify the desired resources for a client application.**

2. **Give the file the same name as the client application and store it in the `/usr/lib/X11/app-defaults` directory.**

   For example, this is part of an application-specific resource file for `xclock`:

   ```
   xclock*digital:on
   ```

Unless overriding values have been set, whenever a user opens the `xclock` client application on this host computer, the clock displays the time in digital format.

To set user-specific resources

1. **Create a file with an A/UX text editor and specify the desired resources.**

2. **Give the file a name that identifies it as an X environment file.**

   For example, you can use `.Xenv` as the filename.

3. **Depending on which shell you use, do the following:**

   □ If you use the C shell, add the `XENVIRONMENT` variable to your `.cshrc` file to point to the pathname of your environment file.

   For example, add this line to your `.cshrc` file:

   ```
   setenv XENVIRONMENT $HOME/.Xenv
   ```

   □ If you use the Bourne shell or Korn shell, add these lines to your `.profile` file:

   ```
   XENVIRONMENT=$HOME/.Xenv
   export XENVIRONMENT
   ```

Values in the environment file override any previously set values.

To set temporary resources

You can use the `-xrm` flag option when you open a client application at the command line to temporarily override default resources. For example, this command opens `xclock` with a blue foreground and a yellow background:

```
xclock -xrm "Foreground: blue" -xrm "Background: yellow" &
```

△ **Important**   When you create or modify resource files, make sure you follow the format and syntax specified on the client application's manual page in *X11 Command Reference for A/UX*. △

Customizing the root window with `xsetroot`

You can use the `xsetroot` client application to modify the appearance of the root window. The features you can modify include:

- pointer shape
- window color
- window pattern

◆ *Note:* You can change the color *or* pattern of the root window; you may not change both color and pattern at the same time.

Changing the root window pointer

When the pointer is on the root window, it appears as a bold X. To specify a different pointer shape, use the command

```
xsetroot -cursor pointerFile maskFile
```

where *pointerFile* contains a bitmap file to be used as the pointer shape and *maskFile* contains a bitmap pattern to be placed behind *pointerFile* to highlight the pointer against the window background. Note that both the *pointerFile* and *maskFile* bitmaps must be the same shape, but the *pointerFile* bitmap should be at least one pixel narrower than the *maskFile* bitmap to set off the *pointerFile* bitmap correctly.

You can use bitmaps that you create with the bitmap client application, or you can use bitmaps from the bitmap directory (/usr/include/X11/bitmaps).

◆ *Note:* Until you are familiar with using and creating bitmaps and masks, it is recommended that you use a black mask. See Chapter 6, "Working With Fonts and Graphics," and bitmap(1X) in *X11 Command Reference for A/UX* for more information.

Changing the root window color

If you have a color monitor, you can change the background color of the root window. Use the command

```
xsetroot -solid color
```

where *color* is any color listed in the color database (/usr/lib/X11/rgb.txt).

You can also specify different background and foreground colors. For example, the command

```
xsetroot -fg white -bg black
```

produces a white foreground and a black background. Note that this is the same as entering

```
xsetroot -rv
```

This command specifies the default gray background:

```
xsetroot -gray
```

Changing the root window pattern

Two flag options are available to change the pattern of the root window:

- `-bitmap` specifies a bitmap pattern that is repeatedly tiled over the root window.
- `-mod` specifies variations of a grid pattern.

When you use the `-bitmap` flag option, you can create your own designs for a single tile with the `bitmap` client application (see Chapter 6, "Working With Fonts and Graphics," for information about the `bitmap` client application). You can also specify a bitmap pattern from the bitmap directory (`/usr/include/X11/bitmaps`).

To tile the pattern across your screen, use this syntax:

```
xsetroot -bitmap bitmapFile
```

`xsetroot` fills the root window with repeated tiles of the bitmap stored in *bitmapFile*. For example,

```
xsetroot -bitmap /usr/include/X11/bitmaps/dot
```

produces tiled dots on the root window.

The `-mod` flag option specifies variations of a grid pattern on your screen. Use this syntax:

```
xsetroot -mod x y
```

The arguments x and y are integers between 1 and 16. A value of 1 produces the smallest pattern, and a value of 16 produces the largest pattern. By supplying different values for x and y, you can produce many variations of the grid pattern. For example, this command

```
xsetroot -mod 5 12
```

produces a plaid-like grid on your screen.

To return the default pointer shape, color, and background pattern of the root window, enter

```
xsetroot -def
```

For more information about xsetroot, see xsetroot(1X) in *X11 Command Reference for A/UX.*

Customizing the console with **xset**

You can use the xset client application to customize characteristics of the X11 console. The features you can modify include

- the system bell
- auto-repeat
- mouse tracking
- the screen saver

To find out the current settings of your computer, enter

```
xset q
```

in an xterm window. A display like this appears in the window:

```
Keyboard Control Values:
    auto repeat:  on     key click percent:  0     LED mask:   00000000
    auto repeating keys:      0000000000000000
                              0000000000000000
                              0000000000000000
                              0000000000000000
    bell percent:  50        bell pitch:  400        bell duration:  100
Pointer Control:
    acceleration:  4/1      threshold:  2
Screen Saver:
    prefer blanking: yes    allow exposures: yes
    time-out: 600     cycle: 600
Colors:
        default colormap:    0x27   BlackPixel:  1        WhitePixel:   0
Font Path:
/usr/lib/X11/fonts/misc/,/usr/lib/X11/fonts/75dpi/,/usr/lib/X11/fonts/100dpi/
Bug Mode: compatibility mode is enabled
```

◆ *Note:* Some of these features may also be set from the t wm Preferences menu. See "Preferences Menu" in Chapter 3 for details.

Adjusting the bell

A client application may ask X11 to ring the A/UX "system bell" when, for some reason, the client application requires your attention. You can modify the behavior of the bell by using the b flag option to xset.

When you start X11, the system bell is turned on. To turn the bell off, enter

```
xset -b
```

or

```
xset b off
```

To turn the bell back on, enter

```
xset b
```

or

```
xset b on
```

You can also use the b option to specify the length or duration of the audible bell tone. For example, this command specifies that the duration of each bell tone is 100 milliseconds (the default for the Macintosh):

```
xset b 0 0 100
```

◆ *Note:* The zeros are placeholders for values that are not currently supported by X11. These placeholder values are required.

Setting auto-repeat

When you press a keyboard key, and the X server automatically repeats that key's character until you release the key, auto-repeat is enabled. To turn auto-repeat off, enter

```
xset -r
```

or

```
xset r off
```

To turn auto-repeat back on, enter

```
xset r
```

or

```
xset r on
```

Controlling mouse tracking

The mouse starts moving the pointer at a slow, precise rate across the screen until the pointer suddenly increases speed and moves a certain number of pixels in one direction during a single mouse action. The increase in speed is the acceleration rate, and the pixel value is the acceleration threshold. These values determine the speed at which the mouse moves the pointer across the screen and permit precision placement of the pointer at slow speeds as well as rapid movement of the pointer across the screen.

You use xset with the m flag option to control mouse movement. This command

```
xset m 4 8
```

specifies that the pointer travels four times faster than the default setting after it is moved more than 8 pixels in one direction.

The first argument is always the rate, and the second argument is always the threshold. If you omit the acceleration threshold argument, the default (2) is used.

Setting the screen saver

The screen saver displays variations of the X logo pattern on the screen. By default, the screen saver is on when you start X11, and it is activated when the system is left idle. This protects your screen from burn-in.

To turn the screen saver off, enter

```
xset -s
```

or

```
xset s off
```

To turn the screen saver back on, enter

```
xset s
```

or

```
xset s on
```

You can also specify a required period of screen inactivity, as well as the length of time each X logo pattern remains on the screen. For example,

```
xset s 300 120
```

specifies an inactivity period of 300 seconds (5 minutes), with each pattern remaining on the screen for 120 seconds (2 minutes) before changing to the next pattern.

You can also use the s flag option to set window exposure. Window exposure is an event in which the server can freely discard window contents to activate and run the screen saver. When window exposure is enabled, the server is not responsible for replacing the contents that appeared on the screen before the screen saver was activated; rather, the various client applications are responsible for restoring the contents after you turn the screen saver off.

By default, window exposure is off when you start X11. The screen saver runs only if the server can regenerate the contents of all windows currently on the screen. To turn on window exposure, enter

```
xset s expose
```

To return window exposure to the default condition, enter

```
xset s noexpose
```

For more information about `xset`, see `xset`(1X) in *X11 Command Reference for A/UX*.

Controlling access to the server with `xhost`

The `xhost` client application controls access to the server. You can run `xhost` either from an `xterm` window or from your `.x11start` startup file.

To display the current list of host computers, enter

```
xhost
```

in an `xterm` window. The message

```
access control enabled
```

appears in the window, along with the current list of computers that have permission to connect to the server.

Initially, the server only allows connections to client applications on the local computer or to computers listed in a host access file. To add to the host access list, enter

```
xhost  +hostname
```

For example, this command

```
xhost +ventana
```

adds the computer named `ventana` to the host access list. You see this message:

```
ventana being added to access control list
```

To remove a computer from the host access list, enter

```
xhost  -hostname
```

For example, this command

```
xhost -ventana
```

removes access privileges for the computer named `ventana`. The message

```
ventana being removed from access control list
```

is displayed. If the computer being removed from the list is currently connected to the server, the connection is not broken. However, future attempts to connect are not permitted.

◆ *Note:* If you accidentally remove your own computer's name from the host access list, you are not able to connect to the local server until you reset and break all connections, thus reestablishing local connections.

To check the host access list to make sure your changes were made, enter

```
xhost
```

The newly updated list appears in the xterm window.

You can also override the host access list. To turn access control off, giving all computers access to the server, enter

```
xhost +
```

Access is granted to all computers and all users in the network. If you then enter

```
xhost
```

you see the message

```
Host Access Control disabled
```

To turn access control on again, enter

```
xhost -
```

Access to the server is then granted to the computers whose names appear in the host access list.

For more information about xhost, see xhost(1X) in *X11 Command Reference for A/UX.*

The next chapter shows you how to use the X client applications that let you work with fonts, bitmap patterns, and screen dumps.

Chapter 6 **Working With Fonts and Graphics**

This chapter discusses the client applications you can use to work with fonts, graphics, and screen dumps. Here is a list of the client applications that are described:

- `bdftosnf` converts fonts from bitmap distribution format to server natural format.
- `showsnf` lists the contents of a font file in server natural format.
- `mkfontdir` creates a font database file.
- `xfd` displays in a window the characters in a font.
- `xfontsel` provides a point-and-click interface for viewing and selecting font names.
- `xlsfonts` lists the server's available fonts.
- `bitmap` provides a bitmap editor.
- `xmag` magnifies a screen image in a window.
- `xwd` dumps window images in a special format.
- `xwud` displays a window dump file on the screen.
- `xpr` formats a window dump file for printing on a PostScript® printer.
- `xdpr` sends a window dump directly to a PostScript printer.

For general information about opening and closing client applications and positioning new windows on the screen, see Chapter 2, "Getting Started." For details about customizing client-application behavior and appearance with flag options and resources, see Chapter 5, "Customizing X11."

bdftosnf

The `bdftosnf` command allows you to convert a font file in bitmap distribution format (BDF) to server natural format (SNF). Converting font files from BDF to SNF allows the X server to recognize and use fonts it wouldn't otherwise be able to use.

To convert a BDF font to an SNF font, enter

`bdftosnf` *fontname.bdf* > *fontname.snf*

in an `xterm` window. Type the name of the BDF font file you want converted in place of *fontname.bdf.* In place of *fontname.snf,* type the name of the font file you're converting and replace `.bdf` with `.snf`. The `bdftosnf` command compiles the selected font file and writes the output into the file *fontname.snf.* To ensure that the SNF font file was compiled properly, you can use the `showsnf` command described in the following section.

For more information about this client application, see `bdftosnf`(1X) in *X11 Command Reference for A/UX.*

showsnf

You can use `showsnf` to display the characters in an SNF font file that you produce using the `bdftosnf` command. You usually use the `showsnf` command to verify that a font file has been compiled correctly.

To display the contents of an SNF font file, enter

`showsnf` *fontname.snf*

in an `xterm` window. Type the name of the SNF file you want to display in place of *fontname.snf.* The `showsnf` command displays the selected SNF file.

For more information about this client application, see `showsnf`(1X) in *X11 Command Reference for A/UX.*

mkfontdir

When you add a font to your system, you must run the `mkfontdir` command to update the font database file (see X(1X) in *X11 Command Reference for A/UX* for information about the font database).

To create a font database file, enter

```
mkfontdir directory
```

in an `xterm` window, where *directory* is the name of the directory that contains the font files you want to add to your font database file. The `mkfontdir` command creates a font database file, called `fonts.dir`, in the directory you specified.

After creating a new font database file, make sure to reset the font path as follows:

```
xset -fp rehash
```

This command notifies the server that the font database file has been updated.

For more information about this client application, see `mkfontdir`(1X) in *X11 Command Reference for A/UX*.

◆ *Note:* Font database files are included in the X11 distribution. You only have to run `mkfontdir` if you add new fonts to your system.

xfd

The `xfd` client application displays the characters in a font.

To open `xfd`, enter

```
xfd -fn fontName &
```

in an xterm window. The xfd client application creates its own window (see Figure 6-1) and displays the characters in *fontName*. For example, Figure 6-1 displays the characters in Courier font.

■ **Figure 6-1**   The xfd window
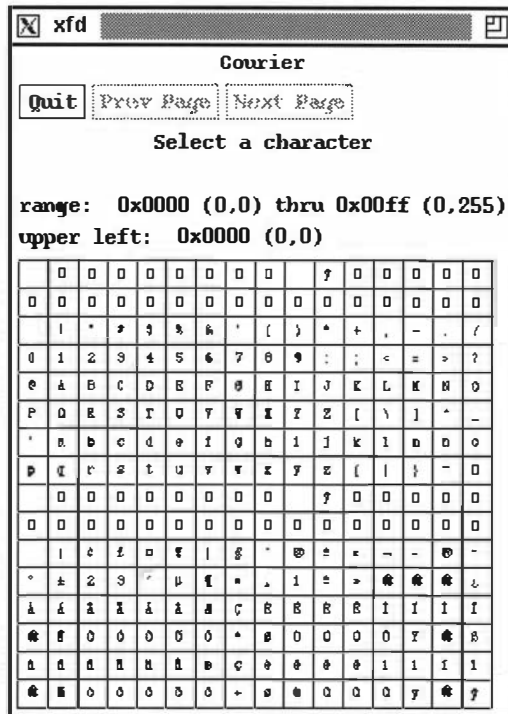
◆ *Note:* You can manipulate the xfd window by using the techniques described in Chapter 3, "Using twm."

If the full set of characters does not fit in a single window, position the pointer in the window and press the RIGHT ARROW key to see the next window of characters. To return to the previous window of characters, press the mouse button while the pointer is in the window. To display a character's decimal and hexadecimal values, position the pointer on the character and press the LEFT ARROW key. The character's decimal and hexadecimal values are displayed at the bottom of the window. To close the window and stop the client application's process, click Quit.

You can use the -verbose flag option to display additional information about a font, such as character width, ascent and descent, and left and right bearing. Position the pointer on a character and click it. You can then enter a greater-than sign (>) to display the maximum bounds of the character, or a less-than sign (<) to display the minimum bounds of the character.

For more information about this client application, see xfd(1X) in *X11 Command Reference for A/UX.*

xfontsel

The xfontsel client application provides a point-and-click interface for viewing the X11 fonts available on a server. You can use xfontsel to examine a sample of a particular font, and you can copy and paste the string that identifies the sample into an application or a resource file. This allows you to customize the font used by a client application without having to look up and enter the font's full name manually.

To open the xfontsel application, enter

xfontsel

in an xterm window. A window similar to the one shown in Figure 6-2 appears on your screen.

■ **Figure 6-2** The xfontsel client application


```
┌─────────────────────────────────────────────────────────────────────┐
│ ⊠  xfontsel ▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓   ⊡ │
│ ┌────┐┌──────┐                                          2 fonts match│
│ │quit││select│                                                       │
│ └────┘└──────┘                                                       │
│ -fndry-fmly-wght-slant-sWdth-adstyl-pxlsz-ptSz-resx-resy-spc-avgWdth-rgstry-encdng│
│              -*-New Century Schoolbook-*-R-Normal-*-18-180-75-75-P-*-*-*  ■│
│                                                                       ■│
│                                                                       │
│                                                                       │
│          ABCDEFGHIJKLMNOPQRSTUVWXYZ                                    │
│          abcdefghijklmnopqrstuvwxyz                                    │
│          0123456789                                                   │
│                                                                       │
│                                                                       │
└─────────────────────────────────────────────────────────────────────┘
```
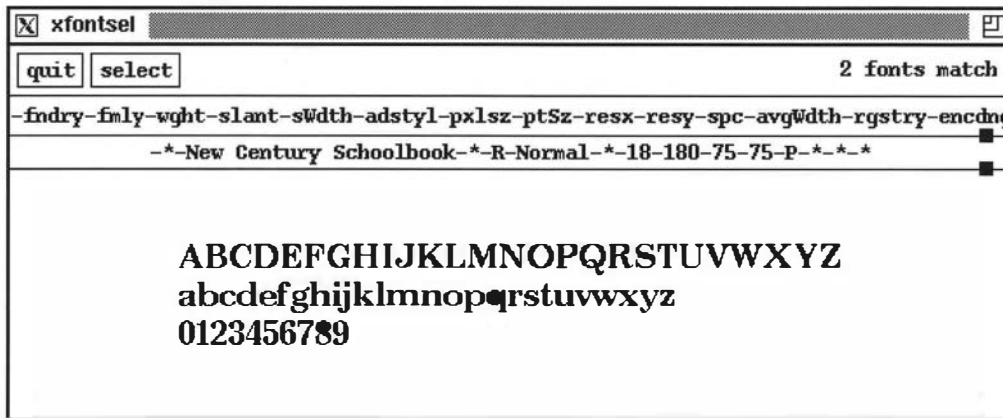
Figure 6-2 shows a sample of the New Century Schoolbook font with certain characteristics specified, such as point size (18) and style (Normal). To modify the sample, you can choose different characteristics from the pop-up menus in the field beginning with -fndry-fmly. For example, position the pointer on -fmly and press the mouse button. A pop-up menu appears displaying the font family names available for viewing. While holding down the mouse button, move the pointer to highlight the name of a font family and release the mouse button. The sample displays the font family you've chosen. You can use the same method to choose characteristics from the other pop-up menus in the field. As you choose different characteristics, xfontsel displays the string identifying the sample you're viewing. In Figure 6-2, the full name of the font is

```
-*-New Century Schoolbook-*-R-Normal-*-18-180-75-75-P-*-*-*
```

Click the "select" button to make the font's full name the PRIMARY selection. You can then paste the font's full name into an application or resource file. To close the window and stop the client application's process, click the "quit" button.

For more information about this client application, see xfontsel(1X) in *X11 Command Reference for A/UX.*

xlsfonts

The `xlsfonts` client application displays a list of the fonts available on your server. These font names can then be used in specifications for fonts used to display text in title bars, menus, and client application windows.

To display the list, enter

```
xlsfonts
```

in an `xterm` window. The server's available fonts are displayed in the window.

Flag options permit you to specify how the list of fonts is displayed. See `xlsfonts`(1X) in *X11 Command Reference for A/UX* for details.

◆ *Note:* For details about specifying fonts and about X11 font-naming conventions, see "Specifying Fonts" in Chapter 5 and X(1X) in *X11 Command Reference for A/UX*.

bitmap

The `bitmap` client application is a bitmap editor that lets you create bitmap patterns that you can use for icons, pointer shapes, or other graphics elements.

To open `bitmap`, enter

```
bitmap filename &
```

in an `xterm` window, where *filename* corresponds to a bitmap pattern in the X11 bitmap directory (`/usr/include/X11/bitmaps`), to a bitmap pattern previously created with the `bitmap` client application, or to a new bitmap pattern of your design. `bitmap` creates a window similar to the one shown in Figure 6-3.

■ **Figure 6-3** The bitmap window

The Bitmap Editor window showing a grid drawing area on the left and buttons on the right: Clear All, Set All, Invert All, Clear Area, Set Area, Invert Area, Copy Area, Move Area, Overlay Area, Line, Circle, Filled Circle, Flood Fill, Set Hot Spot, Clear Hot Spot, Write Output, Quit.
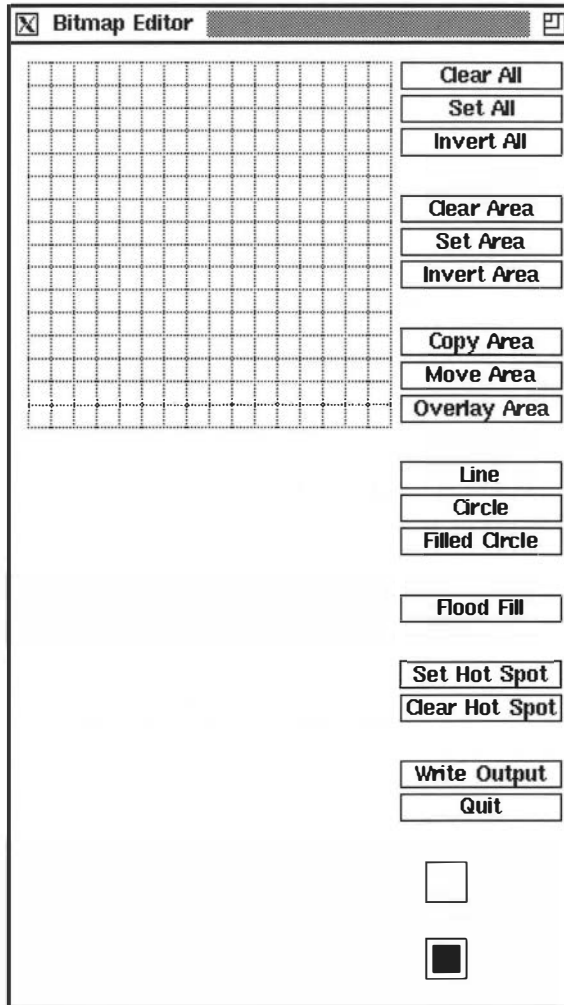
◆ *Note:* You can manipulate the bitmap window by using the techniques described in Chapter 3, "Using twm."

A bitmap is a rectangular grid of pixels, made up of 0 and 1 bits. Each pixel may be set to black or white, representing a bit that is set to either 0 or 1. The squares in the top-left grid in the Bitmap Editor window represent pixels in a bitmap.

You use the mouse and keyboard keys to choose command buttons that set, clear, or invert individual squares to create the desired pattern in the bitmap. The two small subwindows at the bottom of the window display the pattern you're working on in its actual size, in both normal and inverted view. Whenever the grid changes, the images in these subwindows also change.

- When you *clear* a square, the square changes to the background color and the bit is set to 0. You position the pointer on a square and press the RIGHT ARROW key to clear a square.

- When you *set* a square, the square changes to the foreground color and the bit is set to 1. You position the pointer on a square and click it to set a square.

- When you *invert* a square, a 0 changes to a 1 and a square in the background color is displayed in the foreground color, and vice versa. You position the pointer on a square and press the LEFT ARROW key to invert a square.

To work with a group of squares, drag across the desired squares. You can also use the command buttons to clear, set, and invert squares, and to move or copy selected pieces of the bitmap from one part of the grid to another.

For more information about this client application, see bitmap(1X) in *X11 Command Reference for A/UX.*

◆ *Note:* The bmtoa and atobm client applications are filters that convert bitmap files to and from ASCII strings. These converted files can then be used in C programs to declare the size and contents of pointers, icons, and so on. For information about atobm and bmtoa, see bitmap(1X) in *X11 Command Reference for A/UX.*

xmag

The `xmag` client application magnifies sections of the screen. You can use it to examine and modify graphics.

To open `xmag`, enter

```
xmag &
```

in an `xterm` window. The `xmag` pointer, a crosshair surrounded by a blinking box, appears on the screen. Position the pointer on a section of the screen you want to magnify, and click.

After you select the image you want to magnify, a window grid appears. Move the window grid to the desired position and click. The `xmag` window appears and displays the magnified image. Figure 6-4 shows the Magnifying Glass window displaying a close-up view of several lines of text appearing in an `xterm` window.

- **Figure 6-4**   The `xmag` window
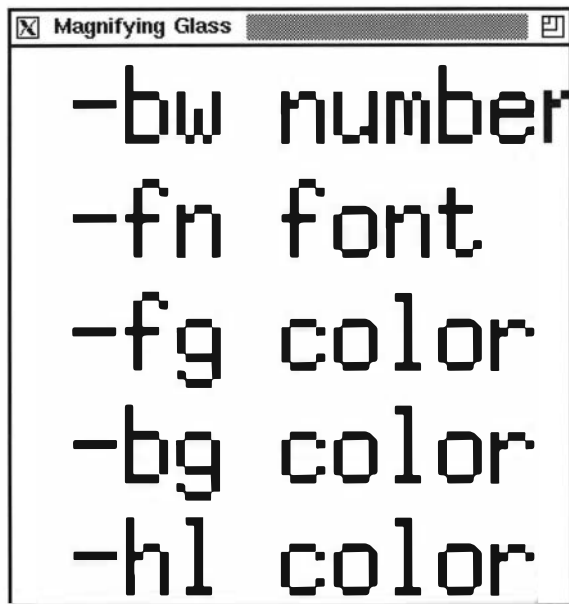
◆ *Note:* You can manipulate the xmag window by using the techniques described in Chapter 3, "Using twm."

If you position the pointer on a pixel in the xmag window and hold down the mouse button, a small subwindow appears that displays the position, number, and color value of the selected pixel in the magnified image.

To clear the xmag window, press the Space bar. The enlarged image disappears, and you can select a new section of the screen to magnify.

For more information about this client application, see xmag(1X) in *X11 Command Reference for A/UX.*

xwd

The xwd client application dumps an image of a window to a specially formatted window dump file. This file can then be read by other client applications for printing, redisplaying, editing, and formatting.

To create a window dump file, enter

```
xwd -out filename
```

in an xterm window. The pointer changes to a crosshair. Position the pointer in the window you want to dump, and click. The console bell rings once when the operation begins and twice when it is completed. The contents of the window are saved in the specified file.

By default, xwd takes a snapshot of the entire window, including the window borders. If you want to use the window snapshot in a document, you might not want the borders to be included in your snapshot. For example, this command

```
xwd -nobdrs -out ventana.1
```

specifies that an image of a window be taken, without window borders, and saved in the file ventana.1.

For more information about this client application, see xwd(1X) in *X11 Command Reference for A/UX.*

xwud

The xwud client application redisplays an image that was captured with xwd. The window image appears at the same location on the screen as the original window and looks exactly like a functioning window.

For example, to specify that the file ventana.1 be displayed, enter

```
xwud -in ventana.1
```

For more information about this client application, see xwud(1X) in *X11 Command Reference for A/UX.*

xpr

The xpr client application formats a window dump file that you produced with xwd for printing on a PostScript printer.

Flag options allow you to add headers and trailers, specify margins, adjust the scale and orientation, and append multiple windows to a single output file. For example,

```
xpr ventana.1 -header "User Guide"
```

formats the file ventana.1 for printing and puts the header "User Guide" at the top.

The command

```
xpr ventana.1 -scale 3
```

displays the window with a magnification factor of 3.

For more information about this client application, see xpr(1X) in *X11 Command Reference for A/UX*.

---

xdpr

The xdpr client application dumps the contents of a window directly to a PostScript printer. First, it runs xwd to dump the window, then it runs xpr to format the contents of the dump file for a PostScript printer, and finally it runs the A/UX line-printer spooler (lpr) to do the actual printing.

◆ *Note:* For information about setting up the A/UX line-printer spooler, see *A/UX Local System Administration* and lpr(1) in *A/UX Command Reference*.

For more information about this client application, see xdpr(1X) in *X11 Command Reference for A/UX*.

In the next chapter, you'll learn about the X client applications that are similar to Macintosh desk accessories.

Chapter 7 **Desktop Client Applications**

This chapter describes desktop client applications, which create their own windows on your desktop. The client applications listed below are similar to Macintosh desk accessories.

- xbiff displays a graphics mail notification program.
- xcalc displays a calculator.
- xclock displays the time of day in a square analog window.
- oclock displays the time of day in a circular analog window.
- xditview allows you to preview device-independent troff files.
- xedit allows you to edit text files.
- xload provides a load-average meter.
- xlogo displays the X logo on the screen.
- xman provides a manual-page browser.
- puzzle displays a puzzle game.

This chapter also describes these graphics demonstration programs:

- maze displays an automated maze demonstration program.
- muncher draws variations of a bitmap pattern.
- plaid draws a plaid pattern.
- ico draws an animated polyhedron.
- xeyes displays a pair of eyes that move with the pointer.

Each of these client applications creates its own window that you can manipulate using the techniques described in Chapter 3, "Using twm."

xbiff

The xbiff client application is a graphics mail notification program. xbiff displays a graphics image of a mailbox. When your mailbox contains no mail, the mailbox flag is down. When mail arrives, the flag goes up, the window changes color, and the mailbox beeps.

To open xbiff, enter

```
xbiff &
```

in an xterm window. The xbiff window, shown in Figure 7-1, appears.

■ **Figure 7-1**   The xbiff window

You can specify an update interval with the -update flag option. For example,

```
xbiff -update 1800 &
```

updates the xbiff window every half hour (1800 seconds).

For more information about this client application, see xbiff(1X) in *X11 Command Reference for A/UX*.

xcalc

The `xcalc` client application is a scientific calculator that emulates a TI-30 scientific calculator.

To open `xcalc`, enter

`xcalc &`

in an `xterm` window. The `xcalc` window, shown in Figure 7-2, appears.

■ **Figure 7-2** The `xcalc` window
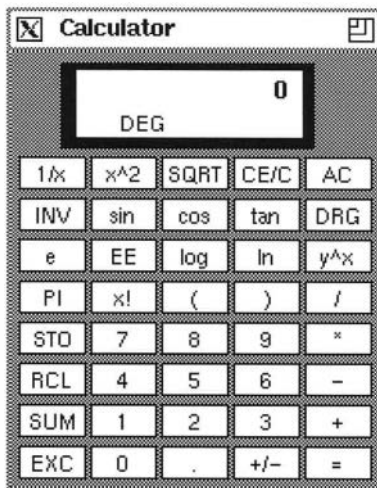
You operate the calculator by clicking the desired calculator control buttons. The number keys, the +/– key, and the numeric operators (+, –, *, /, and =) perform their normal functions and follow standard precedence rules. For example, entering

`3+4*5=`

results in 23, not 35.

Parentheses can be used to override precedence; entering

`(1+2+3)*(4+5+6)=`

results in

`6*15=90`

The `xcalc` keys do the following:

1/x	Replaces the number in the display with its reciprocal.
x^2	Squares the number in the display.
SQRT	Calculates the square root of the number in the display.
CE/C	Clears the display. Clicking this button once clears the number in the display, allowing you to reenter a new number and continue calculating. Double-clicking the button clears the display and terminates the current calculation.
AC	Clears everything in the display and terminates the current calculation.
INV	Inverts the meaning of the function keys.
sin	Computes the sine of the number in the display according to the current DRG mode (see DRG below). If inverted, it computes the arcsine.
cos	Computes the cosine of the number in the display. If inverted, it computes the arccosine.
tan	Computes the tangent of the number in the display. If inverted, it computes the arctangent.
DRG	Changes the DRG mode, as indicated by DEG, RAD, or GRAD at the bottom of the display. In DEG mode, the numbers are in degrees. In RAD mode, the numbers are in radians. In GRAD mode, the numbers are in gradians. When inverted, the DRG key converts degrees to radians to gradians, and vice versa. For example, entering 45 in DEG mode results in .785398, which is 45 degrees converted to radians.
e	Specifies the constant e (2.7182818...).

E E	Specifies exponents. For example, for the value 10^4, select this combination of keys:
	1 0 EE 4
log	Calculates the log (base 10) of the number in the display. When inverted, it raises 10.0 to the power of the number in the display. For example, for the value of 1000, select this combination of keys:
	3 INV log
ln	Calculates the log (base e) of the number in the display. When inverted, it raises e to the power of the number in the display.
y^x	Raises the power of the number on the left to the number on the right. For example, selecting this combination of keys
	2 y^x 3 =
	results in 8, which is 2^3.
P I	Specifies the constant pi (3.145927...).
x!	Computes the factorial of the number in the display. The number in the display must be an integer between 0 and 50.
STO	Copies the number in the display into the memory register of xcalc.
RCL	Recalls the number stored in the memory register of xcalc to the display.
SUM	Adds the number in the display to the number in the memory register of xcalc.
EXC	Swaps the number in the display with the number in the memory register of xcalc.

To "turn off" the calculator and close the xcalc window, position the pointer on the AC key and press the RIGHT ARROW key.

For more information about this client application, see xcalc(1X) in *X11 Command Reference for A/UX*.

xclock

The xclock client application displays the time on your screen. You can specify that the time be displayed in analog or digital form (see Figures 7-3 and 7-4); the time is updated at an interval you choose.

To open xclock, enter

xclock &

in an xterm window. The xclock window appears, as shown in Figure 7-3.

■ **Figure 7-3**   The xclock analog window
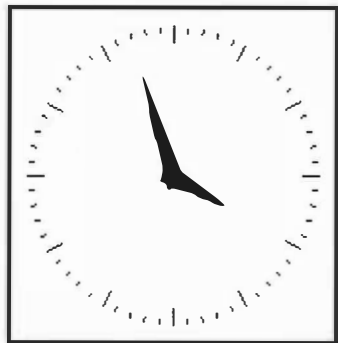
◆ *Note:* Because the .twmrc file specifies no title bar for xclock, you must use the commands in the Window Operations menu to manipulate the xclock window(s). See Chapter 3, "Using twm," and the NoTitle variable in "Specifying Window Variables" in Chapter 5 for more information.

You can modify `xclock` by specifying flag options. For example,

```
xclock -chime &
```

adds a chime to your clock that rings once on the half hour and twice on the hour.

To specify an update interval, use the `-update` option. For example, the command

```
xclock -update 10 &
```

updates the clock every 10 seconds.

With analog clocks, you can also specify the color of the hands on the clock. For example, the following command specifies an analog clock with white hands:

```
xclock -analog -hd white &
```

To open the digital clock, enter

```
xclock -digital &
```

The `xclock` digital window shown in Figure 7-4 appears.

■ **Figure 7-4** The `xclock` digital window

```
Tues Oct 17 17:04:00 1989
```

For more information about this client application, see `xclock`(1X) in *X11 Command Reference for A/UX.*

oclock

The oclock client application displays the time in a circular analog clock on your screen.

To open oclock, enter

oclock &

in an xterm window. The oclock application appears on your screen, as shown in Figure 7-5.

- **Figure 7-5** The oclock client application
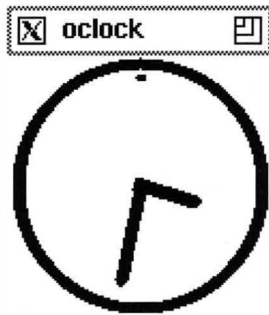
You can specify certain options to oclock on the command line, such as size and color of the clock. You can also set the interval at which the time is updated. See the section "xclock" earlier in this chapter for information about other options you can use with oclock.

For more information about this client application, see oclock(1X) in *X11 Command Reference for A/UX*.

xditview

The xditview client application allows you to preview a file formatted with the A/UX troff text processor. You create a text file using a text editor, such as vi, format and process the file using troff, and use xditview to see how your file will look when you print it.

For example, to process a file named resume.ms, enter

```
troff -ms -TX75 resume.ms > resume
```

in an xterm window. This creates a troff output file named resume. To preview resume, enter

```
xditview resume
```

in the xterm window. The file resume appears in the xditview window exactly as it will look when you print it. You can modify the behavior of xditview by using flag options, such as -page, which specifies a particular page of a file to preview.

For more information about this client application, see xditview(1X) in *X11 Command Reference for A/UX.*

xedit

The xedit client application is a simple mouse-oriented text editor that lets you enter and edit text; cut, copy, and paste text elements; search and replace; and scroll through a file.

To open xedit, enter

```
xedit &
```

in an xterm window. The xedit window, shown in Figure 7-6, appears.

■ **Figure 7-6** The xedit window
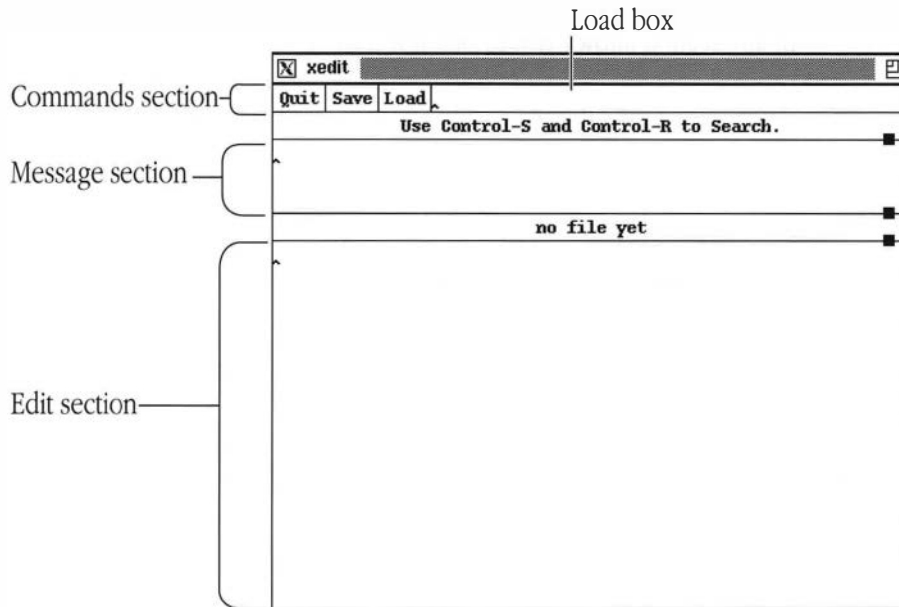
The xedit window consists of three sections. The commands section displays buttons that you can use to operate on your file and includes three text boxes where you can enter strings. The message section displays xedit messages and may also be used as a scratch pad. The edit section displays the text you are editing or creating. You use these buttons in the commands section as follows:

Quit   Quits the current editing session.

Save   Saves the file, putting the original, unedited file in *filename*.BAK.

Load   Loads the file you specify in the Load box and displays it in the editing window.

To edit an existing file, use the command

xedit *filename* &

For more information about this client application, see xedit(1X) in *X11 Command Reference for A/UX*.

xload

The xload client application displays a graph that monitors the system load average, showing how heavily the CPU is being used.

To open xload, enter

```
xload &
```

in an xterm window. The xload window, shown in Figure 7-7, appears.

■ **Figure 7-7** The xload window
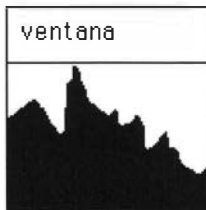
◆ *Note:* Because the .twmrc file specifies no title bar for xload, you must use the commands in the Window Operations menu to manipulate the xload window. See Chapter 3, "Using twm," and the NoTitle variable in "Specifying Window Variables" in Chapter 5 for more information.

You can specify an update interval with the `-update` flag option. For example, the command

```
xload -update 10 &
```

specifies that the display is updated every 10 seconds.

For more information about this client application, see `xload`(1X) in *X11 Command Reference for A/UX.*

`xlogo`

The `xlogo` client application displays the X logo on the screen.

To open `xlogo`, enter

```
xlogo &
```

in an `xterm` window. The `xlogo` window, shown in Figure 7-8, appears.

■ **Figure 7-8**   The `xlogo` window

◆ *Note:* Because the `.twmrc` file specifies no title bar for `xlogo`, you must use the commands in the Window Operations menu to manipulate the `xlogo` window. See Chapter 3, "Using `twm`," and the `NoTitle` variable in "Specifying Window Variables" in Chapter 5 for more information.

For more information about this client application, see `xlogo`(1X) in *X11 Command Reference for A/UX*.

---

xman

The `xman` client application provides a browser for A/UX and X11 manual pages. It allows you to easily access and view manual pages.

To open `xman`, enter

```
xman &
```

in an `xterm` window. The window shown in Figure 7-9 appears on your screen.

◆ *Note:* If the `xman` client application does not appear on your screen and you see a message about setting the MANPATH environment variable, see `xman`(1X) in *X11 Command Reference for A/UX*.

■ **Figure 7-9** The `xman` client application

```
┌─────────────────────────┐
│   Manual Browser        │
│ ┌─────────┐ ┌─────────┐ │
│ │  Help   │ │  Quit   │ │
│ └─────────┘ └─────────┘ │
│ ┌─────────────────────┐ │
│ │    Manual Page      │ │
│ └─────────────────────┘ │
└─────────────────────────┘
```
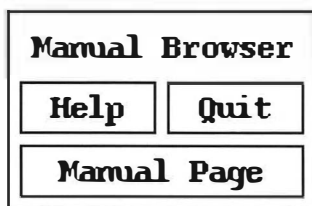
You click the Manual Page button to open a window that displays a manual page. For example, if you click Manual Page in the window shown in Figure 7-9, a window appears that contains the manual page for the xman command. To view a different manual page, choose a section of commands from the pop-up Sections menu. (To choose a section from the Sections menu, click the Sections button in the Manual Page window and drag to the name of a section.) A list of the commands that have available manual pages appears. Select the name of a command whose manual page you want to view. The manual page for that command appears in the window.

For example, if you choose (1) User Commands from the Sections pop-up menu, and then select the mkfontdir command, the window in Figure 7-10 appears on your screen.

■ **Figure 7-10** The manual page for the mkfontdir command

7-14 X11 User's Guide for A/UX
030-1732-B
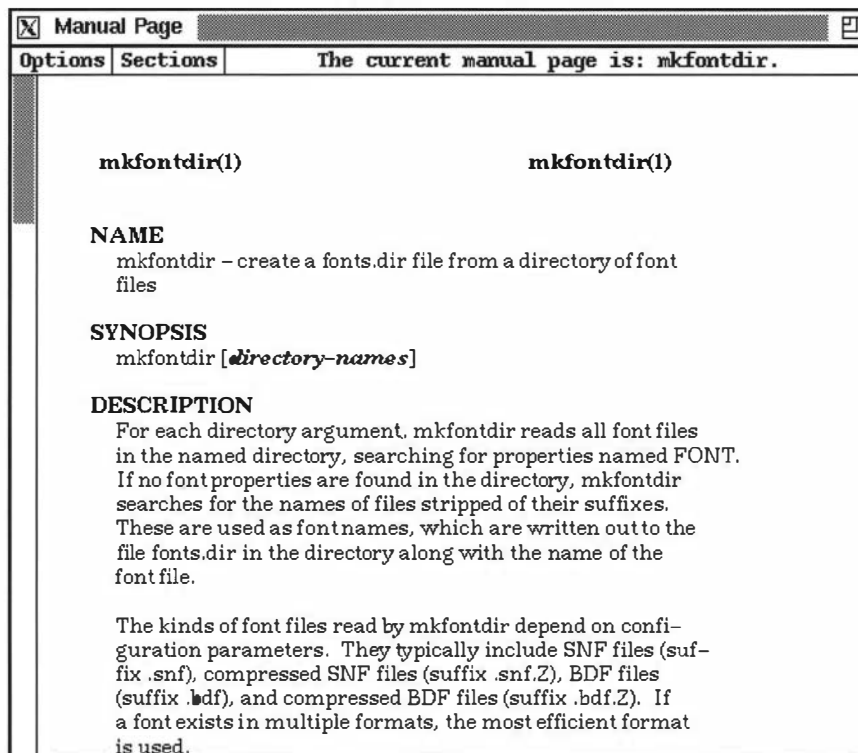
You can use buttons and the scroll bar to view the contents of the mkfontdir manual page. To view another manual page, choose a section of commands from the Sections pop-up menu, and then choose the name of the command whose manual page you want to view.

For more information about this client application, see xman(1X) in *X11 Command Reference for A/UX*.

puzzle

The puzzle client application draws a 15-piece puzzle in a window with numbered tiles arranged four by four.

To open puzzle, enter

puzzle &

in an xterm window. The puzzle window, shown in Figure 7-11, appears.

■ **Figure 7-11** The puzzle window

Control bar
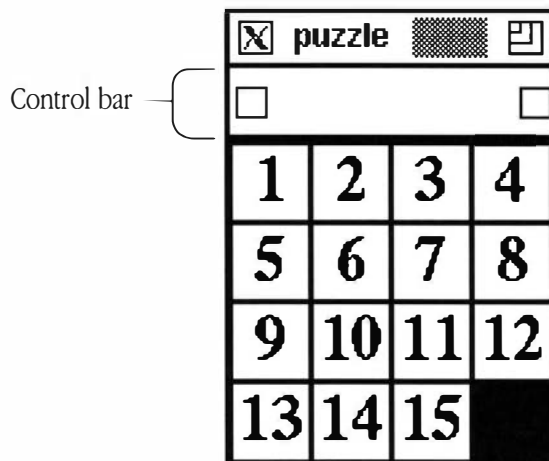
Use the following operations to manipulate the pieces:

- Click the left box under the title bar to scramble the numbered tiles.
- Click the right box under the title bar to solve the puzzle.
- Click a numbered tile to shift that tile into the adjacent empty space; the tile you click must be in the same row or column as the empty space.
- Position the pointer in the control bar and press the LEFT ARROW key to exit.

Several flag options are provided with `puzzle`. For example, the command

```
puzzle -speed 5 &
```

specifies that the tiles move at a rate of five tiles per second when the puzzle is being solved.

You can also change the size of the puzzle:

```
puzzle -size 8x8 &
```

draws a puzzle that has the tiles arranged eight by eight instead of the default four by four.

For more information about this client application, see `puzzle`(1X) in *X11 Command Reference for A/UX*.

maze

The `maze` client application draws and solves a random maze.

To open `maze`, enter

```
maze &
```

in an `xterm` window. The window shown in Figure 7-12 appears on your screen.

■ **Figure 7-12** The maze client application
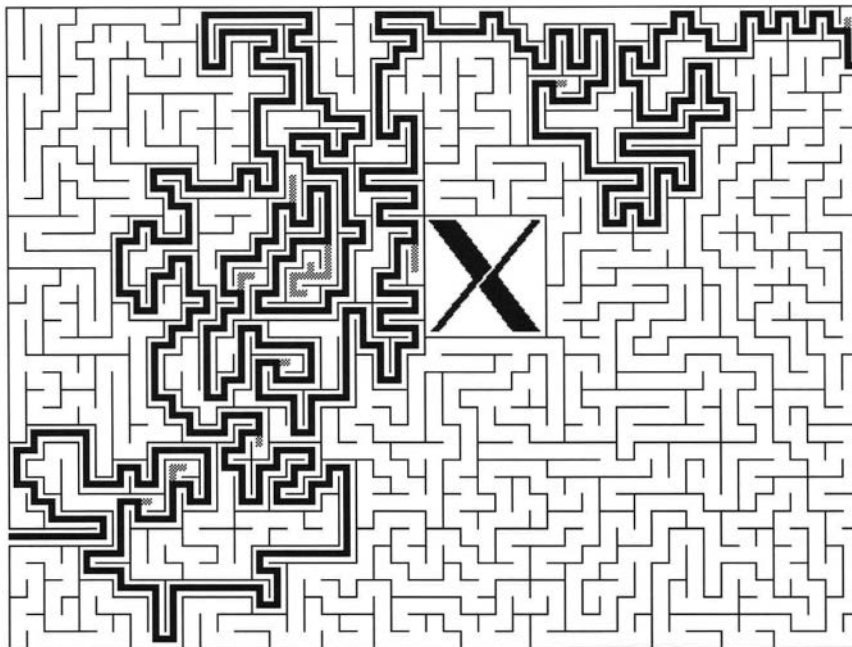
The application automatically begins solving the current maze by drawing lines. You can clear the window and restart the maze by clicking the maze window.

For more information about this client application, see maze(1X) in *X11 Command Reference for A/UX*.

muncher

The muncher client application draws variations of a bitmap pattern in a window.

To open muncher, enter

muncher &

in an xterm window. The muncher window, shown in Figure 7-13, appears.

■ **Figure 7-13** The muncher window
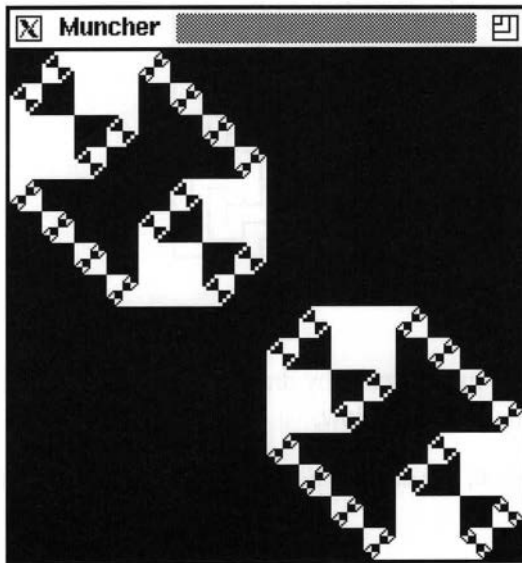
For more information about this client application, see muncher(1X) in *X11 Command Reference for A/UX*.

plaid

The plaid client application draws a continuously changing plaid pattern in a window.

To open plaid, enter

plaid &

in an xterm window. The plaid window, shown in Figure 7-14, appears.

■ **Figure 7-14** The plaid window
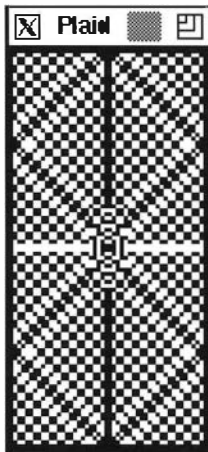
For more information about this client application, see plaid(1X) in *X11 Command Reference for A/UX*.

ico

The `ico` client application draws an animated polyhedron in a window.

To open `ico`, enter

```
ico &
```

in an `xterm` window. The `ico` window, shown in Figure 7-15, appears.

■ **Figure 7-15** The `ico` window
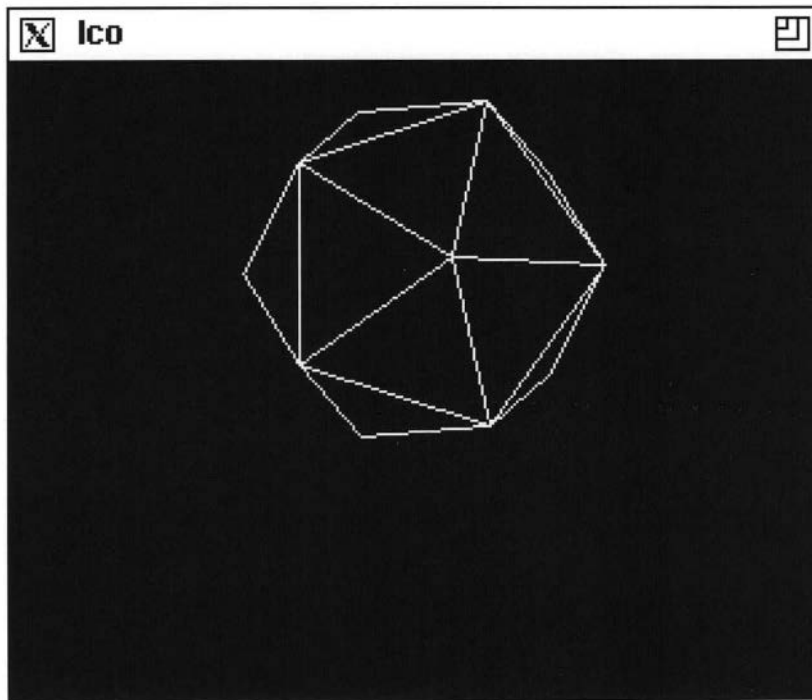
For more information about this client application, see `ico`(1X) in *X11 Command Reference for A/UX*.

xeyes

The xeyes client application displays a pair of "eyes" in a window. These eyes follow the pointer and make it look as if someone is watching the pointer move.

To open xeyes, enter

xeyes &

in an xterm window. The xeyes window, shown in Figure 7-16, appears.

■ **Figure 7-16** The xeyes window
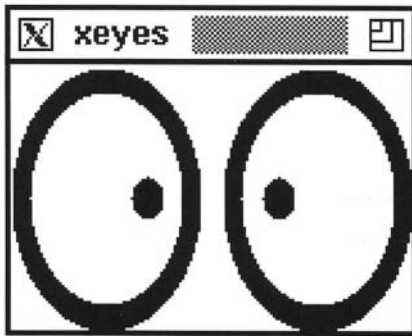
For more information about this client application, see xeyes(1X) in *X11 Command Reference for A/UX.*
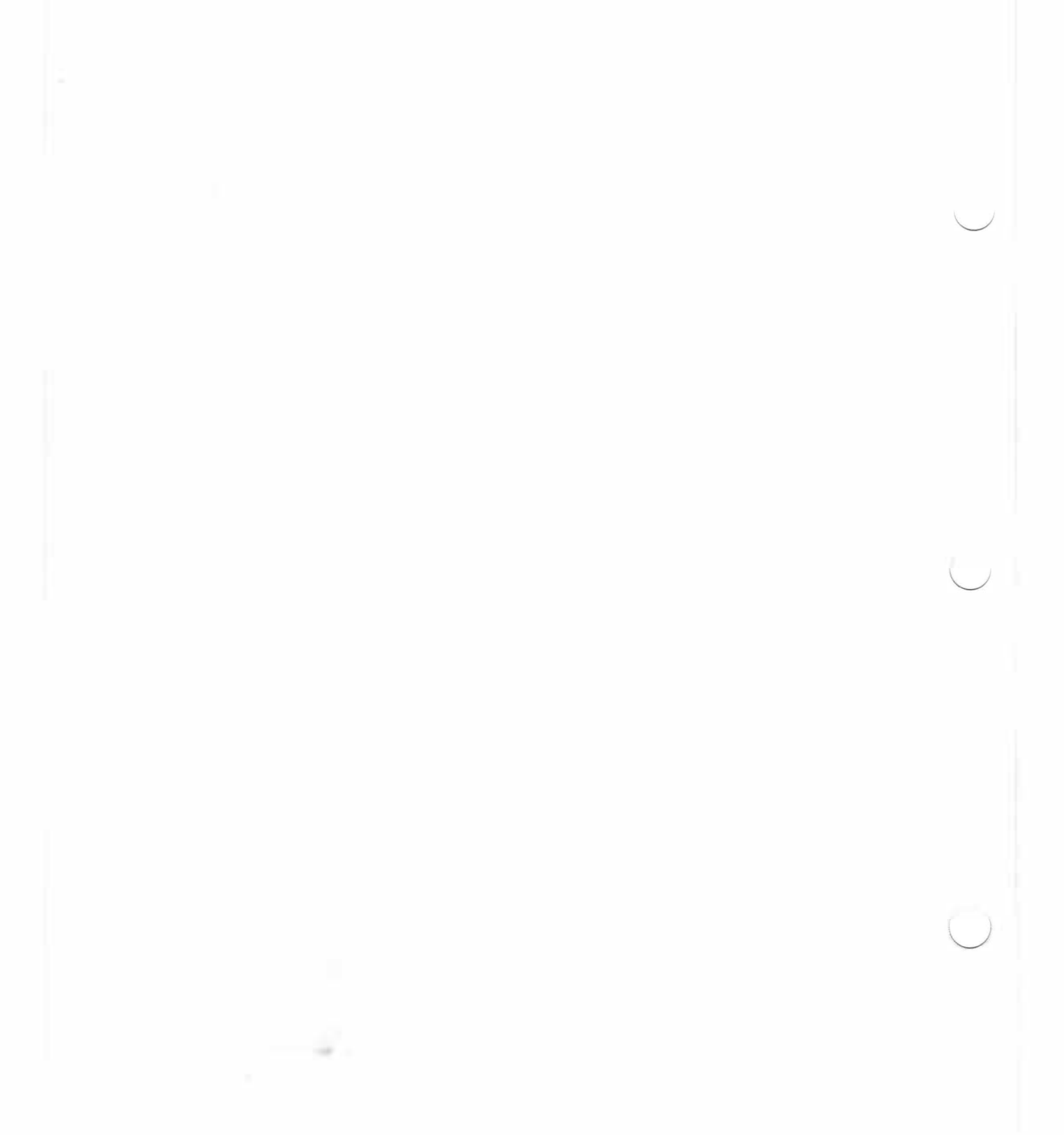
Chapter 8 **Advanced User and Programmer Tools**

This chapter presents client applications and commands primarily for advanced users and programmers. If you're creating new X applications and user interfaces, the client applications and commands described in this chapter are invaluable. If you are an advanced user, such as an administrator of an X network, you also will find many of the commands in this chapter useful.

Read this chapter to learn about the X11 client applications and commands that allow you to

- work with windows
- manage X servers
- develop X applications

Working with windows

Here is a list of client applications and commands that you can use to work with windows:

■ xkill destroys windows.

■ xlswins lists a window's hierarchy.

■ xprop lists window and font properties.

■ xwininfo displays general information about windows.

xkill

You can use the xkill client application to close a client application. If you enter

xkill

in an xterm window, the cursor changes, and the following message appears:

Select the window whose client you wish to kill with button 1...

Click any window, and the server terminates its connection to the client application that created that window. The client application exits, and each related window disappears.

◆ *Note:* You should only use xkill to close a client application when other methods are not successful.

For more information about this client application, see xkill(1X) in *X11 Command Reference for A/UX.*

xlswins

The xlswins client application lists the window hierarchy, or window tree. For example, to display a list of the windows comprising the window tree descending from the root window, enter

```
xlswins
```

in an xterm window. A listing with entries for each window on the desktop appears. For example, you may see a listing like this:

```
0x300090 ()
 0x300091 ()
  0x300094 ()
  0x300096 ()
  0x300098 ()
 0x50000f (ventana 2)
  0x500012 ()
```

You can also specify a window ID number (use the xwininfo client application to determine a window ID).

For example, this command

```
xlswins 0x300090
```

displays the window tree for the window with ID number 0x300090.

For more information about this client application, see xlswins(1X) in *X11 Command Reference for A/UX*.

xprop

The `xprop` client application displays window and font properties.

There are several ways to display a window's properties. One way is to enter

```
xprop
```

in an `xterm` window. The pointer changes to a crosshair, and the following message appears:

```
Please select the window about which you would like information by
clicking the mouse in that window.
```

When you click in the desired window, a listing such as the following appears in the window:

```
WM_STATE(WM_STATE):
                Top-level client window state is NormalState.
                window manager's icon window id #: 0x0
WM_COMMAND(STRING) = { "xterm", "-geometry", "80x24+60+108", \
        "-title", "ventana 2" }
WM_CLIENT_MACHINE(STRING) = "ventana"
WM_CLASS(STRING) = "xterm", "XTerm"
WM_NORMAL_HINTS(WM_SIZE_HINTS):
                user specified location: 81, 43
                user specified size: 514 by 407
                program specified minimum size: 4 by 4
                program specified resize increment: 6 by 13
WM_HINTS(WM_HINTS):
                Client accepts input or input focus: True
                Initial state is Normal State.
WM_ICON_NAME(STRING) = "xterm"
WM_NAME(STRING) = "ventana 2"
```

To display the properties of the root window, enter

```
xprop -root
```

You can specify a window by name. For example, this command

```
xprop -name console
```

displays information about the xterm window named "console." Note that the window name must be unique on the current screen.

You can also specify a window by its window ID (use the xwininfo client application to find out the window ID):

```
xprop -id windowID
```

You can also use xprop to display font properties. For example, the command

```
xprop -font 0x45
```

displays information about font 0x45:

```
FONT: font id # 0x45
POINT_SIZE = 130
WEIGHT = 177
RESOLUTION = 107
X_HEIGHT = 10
QUAD_WIDTH = 6
```

For more information about this client application, see xprop(1X) in *X11 Command Reference for A/UX.*

xwininfo

The xwininfo client application displays information about windows. This includes window ID and window height, width, and depth. To run xwininfo, enter

```
xwininfo
```

in an xterm window. The pointer changes to a crosshair, and the following message appears:

```
xwininfo ==> Please select the window about which you
........ ==> would like information by clicking the
........ ==> mouse in that window.
```

Click a window, and information about that window is displayed—for example,

```
xwininfo ==> Window id: 0x300090 (has no name)

      ==> Upper left X: 81
      ==> Upper left Y: 24
      ==> Width: 316
      ==> Height: 426
      ==> Depth: 1
      ==> Border width: 2
      ==> Window class: InputOutput
      ==> Colormap: 0x80065
      ==> Window Bit Gravity State: ForgetGravity
      ==> Window Window Gravity State: NorthWestGravity
      ==> Window Backing Store State: NotUseful
      ==> Window Save Under State: no
      ==> Window Map State: IsViewable
      ==> Window Override Redirect State: no
      ==> Corners: +81+24 -239+24 -239-26 +81-26
```

◆ *Note:* Depth is expressed as bits per pixel; all other numeric data is measured in pixels.

Without options, general descriptive and numerical information about the location and appearance of a window is displayed. Flag options permit you to request more specific information.

For example, to display a window's event masks, enter

```
xwininfo -events
```

in an xterm window and click in any window. Information like the following appears:

```
xwininfo ==> Window id: 0x300090 (has no name)

      ==> Someone wants these events:
        ==> ButtonPress
        ==> ButtonRelease
        ==> EnterWindow
        ==> LeaveWindow
```

```
==> VisibilityChange
==> SubstructureRedirect
==> Do not propagate these events:
==> Override redirection?: No
```

To display the root, parent, and child window IDs, enter

```
xwininfo -tree
```

and click in any window. Information like the following appears:

```
xwininfo ==> Window id: 0x300090 (has no name)

       ==> Root window id: 0x8006b (has no name)
       ==> Parent window id: 0x8006b (has no name)
       ==> Number of children: 2
         ==> Child window id: 0x50000f (ventana 2)
         ==> Child window id: 0x300091 (has no name)
```

The -root flag option displays information about the root window. For example, enter

```
xwininfo -tree -events -root
```

to display tree and event information for the root window.

Note that you can select any window you desire by clicking in it or by specifying its window ID or the window name as an option.

For more information about this client application, see xwininfo(1X) in *X11 Command Reference for A/UX*.

Working with X servers

These commmands allow you to manage X servers:

- xauth manages the authorization information used in connecting to an X server.
- xdm manages X servers running on a collection of computers, both local and remote.
- xdmshell starts xdm directly from the shell command line.

- `xdpyinfo` displays information about the server.
- `XmacII` is the Macintosh II server for X11.
- `xinit` starts the X server.

---

xauth

The `xauth` command allows you to manage authorization credentials used by client applications when connecting to an X server. By default, the `XmacII` server does not check a client application's authorization credentials. (You must specify the `XmacII` flag option `-auth` when you start up X11 in order to enable a credential check.) When authorization is enabled, each client application that attempts to connect to the X server must provide a copy of valid authorization credentials. Credentials for a client application reside in the file `.Xauthority`, which must appear in the home directory on the computer, local or remote, on which the client application runs. The `xauth` command reads and writes credentials to and from the `.Xauthority` file.

For example, to extract the authorization credentials from the X server running on your local computer, enter

```
xauth extract - $DISPLAY > /tmp/authority
```

in an `xterm` window. In this example, `xauth` copies credentials from your `.Xauthority` file into a file named `/tmp/authority`. Next, you copy the information in `/tmp/authority` to a remote computer. To complete the procedure, you run `xauth` to merge your authorization information into the X server's authority files on the remote computer. You can then run client applications on the remote computer while displaying the applications on your local computer screen.

For more information about this command, see `xauth`(1X) in *X11 Command Reference for A/UX*.

xdm

The xdm command allows you to manage a collection of X servers, both local and remote. xdm is used primarily for providing network-based services for X servers. You can use xdm to run a standard login session, including prompts for login name, password, and user authentication. You can customize xdm to provide a specific interface to the login session. In addition, xdm allows you to determine which client applications initially appear on your computer screen.

◆ *Note:* The xdm command is used mainly by programmers and system administrators.

For more information about this command, see xdm(1X) in *X11 Command Reference for A/UX*.

xdmshell

The xdmshell command allows you to start an X server under the control of xdm from the A/UX System Console (console emulator window). To get to the A/UX System Console, choose Console Emulator from the Change Session Type dialog box when you log in to A/UX. See *A/UX Essentials* for more information about logging in to the A/UX System Console.

To run xdmshell, enter

xdmshell

in the A/UX System Console. This runs xdm and starts a standard login session.

◆ *Note:* Make sure you run the xdmshell command from the A/UX System Console window, *not* from an xterm window.

For more information about this command, see xdmshell(1X) in *X11 Command Reference for A/UX*.

xdpyinfo

The xdpyinfo command displays information about the X server. This includes the number of screens connected to the server and other numeric data regarding the dimensions and properties of your screen(s).

To run xdpyinfo, enter

xdpyinfo

in an xterm window. A listing like the following appears:

```
name of display:  unix:0.0
version number:  11.0
vendor string:  Apple Computer, Inc.
vendor release number:  4
maximum request size: 16384 longwords (65536 bytes)
motion buffer size: 0
bitmap unit, bit order, padding:  32, MSBFirst, 32
image byte order:  MSBFirst
keycode range:  minimum 8, maximum 134
default screen number:  0
number of screens:  1

screen #0:
 dimensions:  640x480 pixels (225x169 millimeters)
 resolution:  72x72 dots per inch
 root window id:  0x8006b
 depth of root window:  1 plane
 number of colormaps:  minimum 1, maximum 1
 default colormap:  0x80065
 default number of colormap cells:  2
 preallocated pixels:  black 1, white 0
 options:  backing-store YES, save-unders YES
 current input event mask:  0x10800d
  KeyPressMask     ButtonPressMask     ButtonReleaseMask
  ExposureMask     SubstructureRedirectMask
 number of visuals:  1
 default visual id: 0x80064
```

```
visual:
 visual id:  0x80064
 class:  StaticGray
 depth:  1 plane
 size of colormap:  2 entries
 red, green, blue masks:  0x0, 0x0, 0x0
 significant bits in color specification:  1 bits
```

For more information about this command, see xdpyinfo(1X) in *X11 Command Reference for A/UX*.

XmacII

XmacII is the X11 server for Macintosh II–family computers running the A/UX operating system. The server normally is started by the shell script X.

For more information about the X11 server, see XmacII(1X) in *X11 Command Reference for A/UX*.

---

xinit

If you want greater control over your startup environment, you may prefer to start X11 manually by using the xinit client application. xinit starts the server and opens one xterm window. You can then use this xterm window to open additional client applications.

◆   *Note:* If you close the first xterm window opened by xinit, xinit automatically shuts down the X server.

For more information about this client application, see xinit(1X) in *X11 Command Reference for A/UX*.

Using advanced programming tools

You can use these commands and client applications to develop new X applications:

- `appres` lists the resources that an application loads.
- `editres` allows you to view and edit the full widget hierarchy of an X Toolkit client application.
- `imake` provides a C preprocessor interface to the `make` utility.
- `listres` generates a list of a widget's resource database.
- `makedepend` creates dependencies in makefiles.
- `mkdirhier` creates a directory hierarchy.
- `showrgb` displays the contents of an RGB database.
- `viewres` displays a browser that shows the widget class hierarchy of the Athena Widget Set.
- `xev` displays event information.
- `xlsatoms` lists atoms interned on a server.
- `xlsclients` lists the client applications displayed on an X server.
- `xmkmf` generates an X11 makefile from an Imakefile by running the `imake` command.
- `Xrepair` resets the A/UX console screen.
- `xmodmap` and `xprkbd` facilitate keyboard remapping.
- `xstdcmap` defines colormap properties.

---

appres

The `appres` command allows you to list the resources used by a particular application.

For example, to display the resources that the client application `xload` uses, enter

`appres XLoad`

in an `xterm` window. A list of the resources used by `xload` appears in the window.

◆ *Note:* Make sure you type an uppercase x and L when you enter this example at the command prompt.

For more information about this command, see `appres`(1X) in *X11 Command Reference for A/UX.*

---

editres

The `editres` client application allows you to view and edit the resources used by a particular application. You can use `editres` to construct resource specifications, apply the resource to an application, view the results dynamically, and append the resource to a user's X resources file.

To open `editres`, enter

```
editres
```

in an `xterm` window. A window similar to the one shown in Figure 8-1 appears on your screen; don't worry if you don't see any information in the window right away.

■ **Figure 8-1** The editres client application
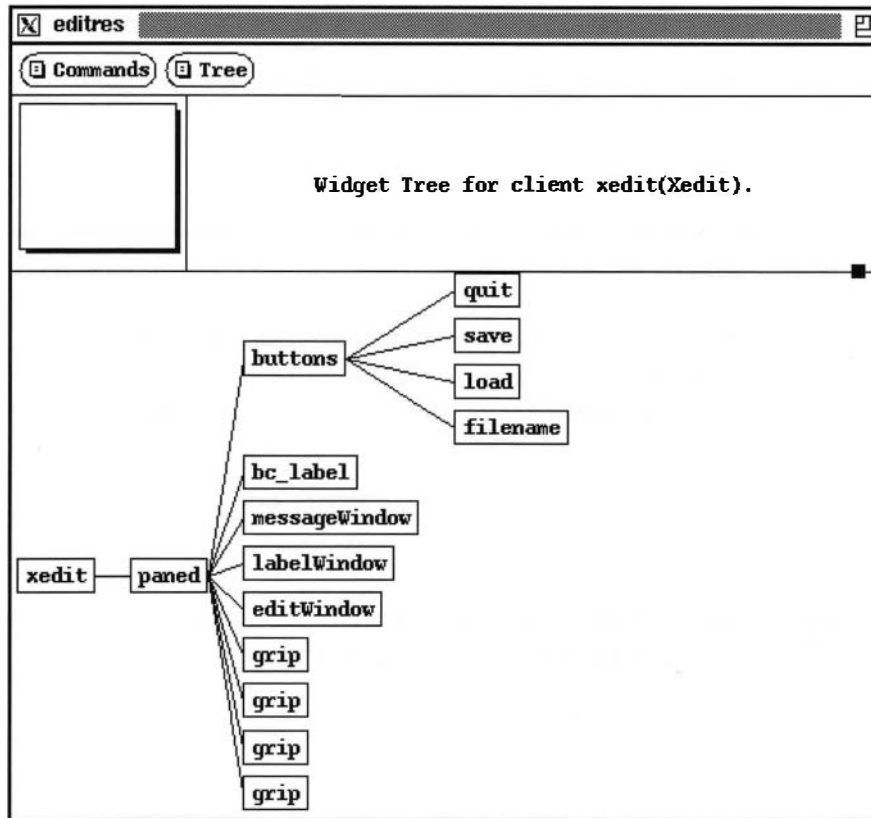
To select an application whose resources you want to view, click the Commands button and choose Get Widget Tree from the pop-up menu. The cursor changes, and you can click on the window of a client application. If the application you click on uses the editres protocol, editres displays the client's widget hierarchy as a tree in the window.

For example, Figure 8-1 shows the widget tree for the xedit client application. To see the entire widget tree, click the box, called the **panner,** located in the upper-left corner of the window just below the Commands button. While holding down the mouse button, drag the panner until the rest of the widget tree appears in the window.

For more information about this client application, see editres(1X) in *X11 Command Reference for A/UX*.

imake

The imake command provides a C preprocessor interface to the make utility. You can use imake in combination with a template, a set of macro functions, and an Imakefile.

For example, to run imake, enter

```
imake -DUseInstalled -I/usr/lib/X11/config
```

in an xterm window. This example uses configuration files specifically for X11 for A/UX.

For more information about this command, see imake(1X) in *X11 Command Reference for A/UX*.

listres

The listres command allows you to list a widget's resource database.

For example, to display information about all known widgets on your computer, enter

```
listres -all
```

in an xterm window. A multicolumn list appears in the window, displaying the names of the widgets and their class hierarchies.

For more information about this command, see listres(1X) in *X11 Command Reference for A/UX*.

makedepend

The `makedepend` command works in conjunction with the `make` command. You run the `makedepend` command on a source file to create a makefile that incorporates dependency information, naming all the files included in a source file. When you run the `make` command, it reads the makefile in order to know which object files must be recompiled when a source file changes.

To run `makedepend` on a source file, enter

`makedepend` *sourcefile.c*

in an `xterm` window, typing the name of a source file in place of *sourcefile.c*. This command creates a makefile that reflects the dependencies of the source file you chose on other included files.

For more information about this command, see `makedepend`(1X) in *X11 Command Reference for A/UX*.

mkdirhier

The `mkdirhier` command allows you to create a directory hierarchy. If any of the parent directories of the specified directory do not exist, `mkdirhier` creates the parent directories as well.

For example, to create a directory named `/apple/workfiles`, enter

`mkdirhier /apple/workfiles`

in an `xterm` window. This command creates a directory named `workfiles` and, if it doesn't already exist, a parent directory named `apple`.

For more information about this command, see `mkdirhier`(1X) in *X11 Command Reference for A/UX*.

showrgb

The showrgb command lists the contents of the X11 color database.

To run showrgb, enter

showrgb

in an xterm window. A list of the colors available on your server appears in the window.

For more information about this command, see showrgb(1X) in *X11 Command Reference for A/UX*.

viewres

The viewres client application provides a class browser for the X Toolkit (Xt). You can use viewres to examine the class structure and resources for the Athena Widget Set.

To open viewres, enter

viewres

in an xterm window. The window shown in Figure 8-2 appears on your screen.

- **Figure 8-2** The `viewres` client application
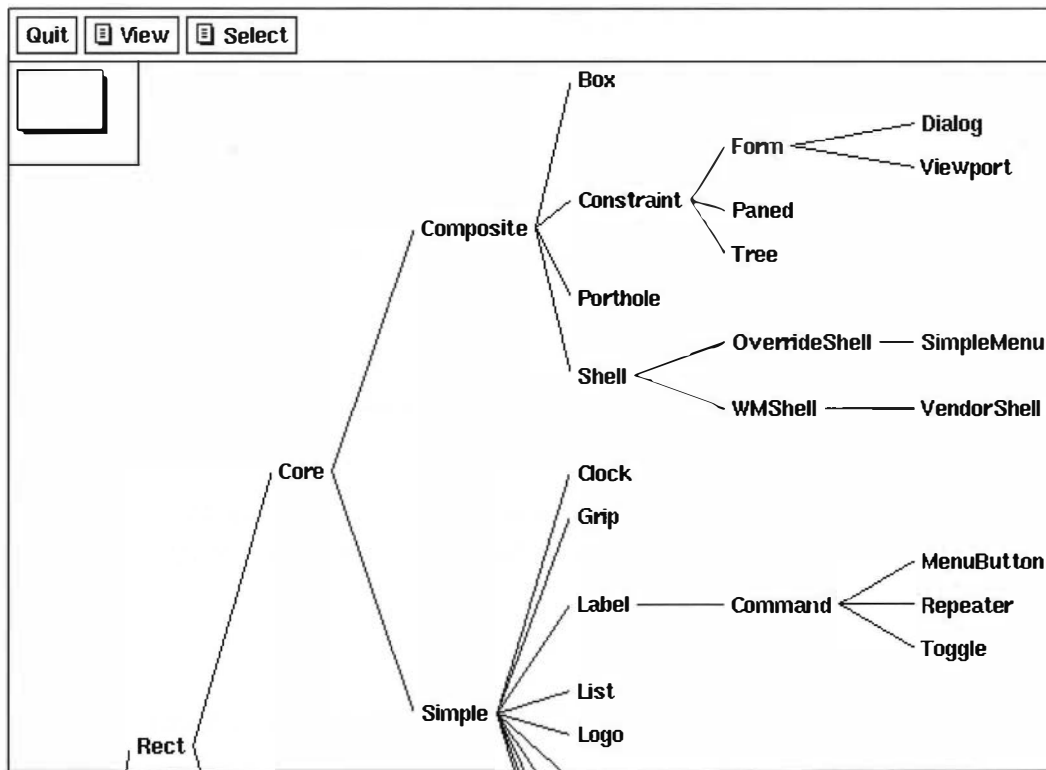
You can click the View button to choose an application whose resources you want to view. Figure 8-2 shows the widget tree for the `viewres` client application. To see the entire widget tree, click the box, called the panner, located in the upper-left corner of the window just below the Quit button. While holding down the mouse button, drag the panner until the rest of the widget tree appears in the window.

You can click the Select button to make `viewres` the PRIMARY selection. You can then paste the string for the `viewres` widget tree into another application.

For more information about this client application, see `viewres(1X)` in *X11 Command Reference for A/UX*.

xev

The xev command creates a window and then asks the X server to send it events when anything happens in the window. You can use xev to see what causes particular events to occur and then display the event information.

To run xev, enter

xev &

in an xterm window. The xev command creates a new window and displays information like the following in the xterm window:

```
NotifyReparentNotify event, serial 10, synthetic NO, window 0xa00001,
   event 0xa00001, window 0xa00001, parent 0x30010f,
   (0,19), override NO
```

For more information about this command, see xev(1X) in *X11 Command Reference for A/UX.*

xlsatoms

The xlsatoms command lists the interned atoms defined on a server. Atoms provide a shorthand method of communication between client applications and a window manager running on a particular server.

To run xlsatoms, enter

xlsatoms

in an xterm window. A list of the atoms interned on the server appears in the xterm window.

For more information about this command, see xlsatoms(1X) in *X11 Command Reference for A/UX.*

xlsclients

The `xlsclients` command lists the client applications running on an X server.

To run `xlsclients`, enter

```
xlsclients
```

in an `xterm` window. A list of active client applications appears in the window.

For more information about this command, see `xlsclients`(1X) in *X11 Command Reference for A/UX.*

xmkmf

The `xmkmf` command runs the `imake` command to generate an X11 makefile from an Imakefile.

For example, to run `xmkmf`, enter

```
xmkmf
```

in an `xterm` window. The following information appears in the window:

```
imake -DUseInstalled -I/usr/lib/X11/config
```

This command shown in this example uses the `imake` command line presented earlier in this chapter. The `xmkmf` command creates an X11 makefile that you can use with the `make` command.

For more information about this command, see `xmkmf`(1X) in *X11 Command Reference for A/UX.*

Xrepair

The `Xrepair` command resets the A/UX console window after the `XmacII` server exits. `Xrepair` is normally run by the shell script X.

For more information about this command, see xrepair(1X) in *X11 Command Reference for A/UX.*

xmodmap and xprkbd

The xmodmap and xprkbd commands let you modify how keys are mapped on the keyboard. You use xprkbd to display the current keyboard map, and then you use xmodmap to change the values.

The keyboard map is a table of key codes and keysym codes. Key codes are server specific and represent the actual physical keys on the keyboard; keysym codes are server independent and represent the name of a key, which usually corresponds to its function.

To run xprkbd, enter

xprkbd

in an xterm window. A listing like this appears:

```
0x8    0x61   a     0x41   A
0x9    0x73   s     0x53   S
0xa    0x64   d     0x44   D
0xb    0x66   f     0x46   F
0xc    0x68   h     0x48   H
0xe    0x67   g     0x47   G
0xf    0x7a   z     0x5a   Z
0x10   0x78   x     0x58   X
0x11   0x63   c     0x43   C
0x12   0x76   v     0x56   V
0x13
0x14   0x62   b     0x42   B
0x15   0x71   q     0x51   Q
0x9    0x77   w     0x57   W
0x8    0x65   e     0x45   E
0x9    0x72   r     0x52   R
0x8    0x79   y     0x59   Y
0x9    0x74   t     0x54   T
0x14   0x31   1     0x21   exclam
```

```
0x15   0x32   2      0x40   at
0x9    0x33   3      0x23   numbersign
0x8    0x34   4      0x24   dollar
0x9    0x36   6      0x5e   asciicircum
0x8    0x35   5      0x25   percent
0x9    0x7d   equal  0x2b   plus
```

To run xmodmap, enter

```
xmodmap
```

in an xterm window. A listing like this appears:

```
xmodmap: up to 2 keys per modifier, (keycodes in parentheses):
```

```
shift      Shift_L (0x40)
lock       Caps_Lock (0x401)
control    Control_L (0x3e)
mod1       Meta_L (0x3f), Meta_R (0x46)
mod2
mod3
mod4
mod5
```

This list displays the current settings of the modifier keys. To change the mapping of a key, look up the setting in the xprkbd listing and enter that value as an argument to xmodmap. For example, to make the CLEAR key function as the CONTROL key, enter

```
xmodmap -e "keysym Clear = Control_L"
xmodmap -e "add Control = Control_L"
```

These commands specify that the CLEAR key operates as a second CONTROL key. For example, if you now wanted to press CONTROL-G, you could press CLEAR and the letter G instead.

◆ *Note:* You can run xmodmap from your .x11start startup file so its settings are in effect each time you start X11.

For more information about these client applications, see xmodmap(1X) in *X11 Command Reference for A/UX*.

xstdcmap

The `xstdcmap` command lets you define colormap properties.

You can define the following colormap properties:

```
RGB_BEST_MAP
RGB_BLUE_MAP
RGB_DEFAULT_MAP
RGB_GRAY_MAP
RGB_GREEN_MAP
RGB_RED_MAP
```

To define all six properties, enter this command in an `xterm` window:

```
xstdcmap -all
```

Note that not all screens support all six colormap properties. The `xstdcmap` command determines the best allocations and visuals for the colormap properties.

Each property can also be selected individually. For example, to select `RGB_BLUE_MAP`, use the command

```
xstdcmap -blue
```

To remove a property, use this command:

```
xstdcmap -delete property
```

where *property* is one of the six properties. For example, to remove gray, enter

```
xstdcmap -delete RGB_GRAY_MAP
```

◆ *Note:* You should `run xstdcmap` from your `.x11start` startup file so all client applications accessing the server can share finite colormap resources.

For more information about this command, see `xstdcmap`(1X) in *X11 Command Reference for A/UX*.

Index