MacX™

User's Guide

© Apple Computer, Inc., 1990
20525 Mariani Avenue
Cupertino, CA 95014
(408) 996-1010

Apple, the Apple logo, AppleLink, AppleShare, AppleTalk, A/UX, HyperCard, ImageWriter, LaserWriter, Macintosh, MacTCP, and MultiFinder are registered trademarks of Apple Computer, Inc.

Color QuickDraw, Disk First Aid, Finder, and MacX are trademarks of Apple Computer, Inc.

Helvetica, Linotronic, and Times are registered trademarks of Linotype Company.

ITC Zapf Dingbats is a registered trademark of International Typeface Corporation.

MacDraw and MacWrite are registered trademarks of CLARIS Corporation.

Microsoft is a registered trademark of Microsoft Corporation.

PostScript is a registered trademark, and Illustrator is a trademark, of Adobe Systems Incorporated.

ULTRIX and VMS are trademarks of Digital Equipment Corporation.

UNIX is a registered trademark of AT&T Information Systems.

Varityper is a registered trademark, and VT600 is a trademark, of AM International, Inc.

X Window System is a trademark of MIT.

Simultaneously published in the United States and Canada.

Contents

Figures and tables

Welcome to MacX

THE *MACX USER'S GUIDE* DESCRIBES HOW TO INSTALL AND USE MACX™, an X Window System display server, developed by Apple, for the Macintosh® family of computers. To help you use this guide effectively, this Preface summarizes the contents of each chapter and appendix and suggests alternative approaches to reading this book depending on your knowledge of computers and network communications. A list of related publications and resources is also included should you need additional information about the X Window System.

Who should read this guide

Anyone who plans to install and use MacX should consult this guide. MacX conforms to the latest version of X (version 11, release 3); however, it contains many enhancements made possible by the Macintosh method for interacting with users (formally called the *Macintosh User Interface*). Consequently, anyone using MacX should read or browse through this guide to learn about the product's special features.

Because X users have little in common, with respect to education, knowledge of X, and profession, this user's guide tries to accommodate its readers' divergent needs. The next section, "How to Use this Guide," suggests two approaches to reading this manual, based on your technical knowledge and experience with X.

Although this guide teaches you how to install and use MacX, it will not be sufficient for people who have never used X programs before. Since X applications, called **clients,** run on other mini- or mainframe computers under operating systems like UNIX® or VMS™, you must learn how to issue commands to run clients and perform other necessary work on those systems. Consult the documentation written for the systems on which your clients reside for more information. In addition, "Other Publications and Resources" at the end of this Preface lists some books devoted to using X on UNIX, A/UX® (UNIX for the Macintosh), and other operating systems.

In addition, this guide is aimed at readers who know basic Macintosh techniques and operations. Thus, if you don't know how to drag and click with the mouse, open folders, or move and resize windows, read the tutorial chapter in your Macintosh system software guide or its successor, *Getting Started With Your Macintosh.* Being familiar with at least one other Macintosh application, such as MacWrite®, also helps.

How to use this guide

Your experience with X, computers, and networks will determine the way you should use the *MacX User's Guide.* Choose one of the two suggested approaches that best suits you, depending on your level of experience in these areas. Figure P-1 summarizes the suggested approaches described in the next two paragraphs.

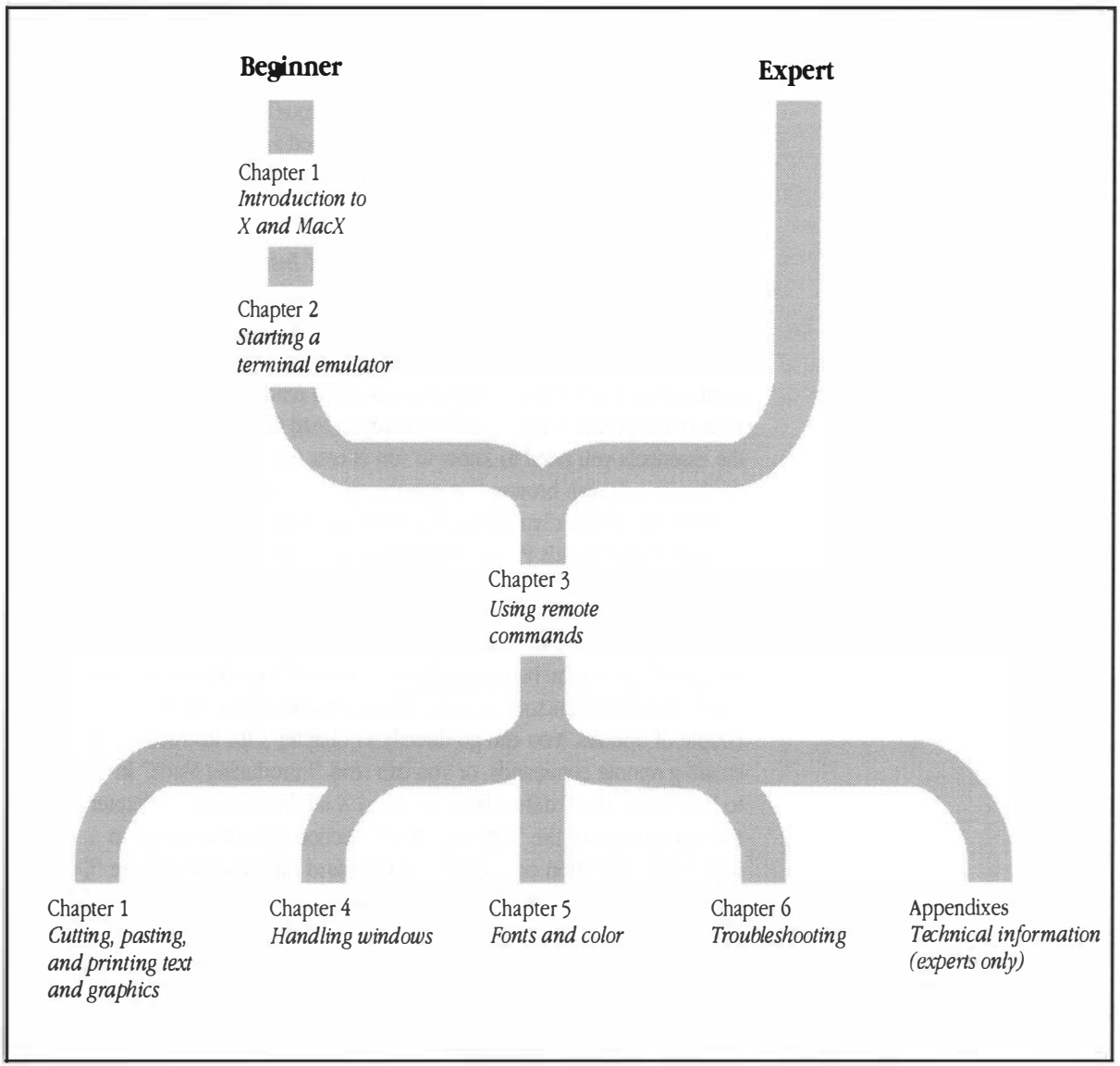
For the beginner

If you haven't installed MacX, go to the *MacX Installation Guide* for step-by-step instructions. Next, read Chapter 1 if you're not too familiar with the X Window System. Chapter 1 briefly covers what X is, how it works, and how MacX differs from traditional implementations of X. It also contains a summary of MacX features that you'll want to read or scan to get a global view of the product. Then you'll be ready to read Chapter 2, which explains the essentials you need to know to run clients on a Macintosh. After learning the basics, you can browse through the other chapters at your convenience to learn about other handy, but nonessential, features. See the "Summary of Contents" later in this Preface for synopses of the remaining chapters and appendixes.

For the expert

People with technical backgrounds or extensive X experience can bypass much of the introductory material. Thus, after installing MacX, you have a couple of options. You can go directly to Chapter 3 for instructions on creating remote commands, or you can read "Introducing MacX" in Chapter 1 to learn how MacX differs from standard X implementations. Chapter 2 is tutorial. Except for the "Opening MacX" section (which contains an important subsection on mouse and keyboard differences) and the "Quitting MacX" section, you can get the same information, expanded upon, from Chapter 3. Essentially, you'll want to glean what you need from the first two chapters and then concentrate on the remaining chapters and appendixes, where specific features are documented in detail.

■ **Figure P-1** Road map to the *MacX User's Guide*



Summary of contents

This section summarizes the contents of the chapters and appendixes in this guide.

- Chapter 1, "MacX Overview," briefly describes the X Window System and then introduces MacX. It defines the major components of MacX and explains how MacX works with a visible or invisible root window and with multiple screens. Finally, it presents an overview of MacX features, including illustrations of menus and the dialog boxes that appear. Included in this overview are instructions for transferring text and graphics between applications and for printing selections from the Clipboard. See "Cut, Copy, Paste and Other Editing Commands" and "Printing Commands" for this information.
- Chapter 2, "Getting Started," covers the basic things you need to know to function in an X environment using MacX. It describes how to open MacX, create a remote command to start a terminal emulator, close clients, and exit from MacX. The section "Mouse and Keyboard Differences" explains what keys to use for extra mouse buttons and contains a table of keyboard substitutes for mouse clicks and selections.
- Chapter 3, "Using Remote Commands," describes how to create, execute, and edit remote commands using the Remote menu and the Remote Command window. It also covers security features and various ways to use the *–display* option.
- Chapter 4, "Handling Windows," explains how to move, resize, and iconify windows. It explains how to select different window styles and how to select a default window style. Also included is a section on using and setting preferences for visible root windows.

- Chapter 5, "Managing Fonts and Color," describes procedures for adding and removing fonts and colors, creating aliases for font names, scrolling, and many other features supplied by the Font Director and the Color Namer.
- Chapter 6, "Troubleshooting," explains how to recover from errors that occur. The section "Memory-Related Problems" gives suggestions for operating with limited memory.
- Appendix A, "MacX Versus Other X Servers," states the differences between MacX and conventional X servers. It also explains the purpose of two miscellaneous preferences concerning smoother but slower animation and mouse **warping.**
- Appendix B, "The MacX Window Manager," covers some technical aspects of the MacX Window Manager.
- Appendix C, "Bitmap Distribution Format 2.1," contains a copy of the X Consortium document that describes the format of BDF files.
- Appendix D, "X Logical Font Description Conventions," contains a copy of the X Consortium document that specifies the format of X font names.
- Appendix E, "Character Conversion Charts," contains a matrix of the ISO Latin 1 and Macintosh extended ASCII character sets that shows the hexadecimal equivalent for each character. The appendix also contains two character-mapping charts.

Special text elements

This guide includes special elements in the text to highlight important or complementary information.

- *By the way:* A note like this presents additional information or interesting sidelights.

△ **Important**

Text labeled *important* cautions you against doing things that might be inconvenient or troublesome, but not damaging to your hardware or software. △

The first reference to a client (an X application) or UNIX command in text is *italicized* for the benefit of readers who are not familiar with X or UNIX naming conventions.

Terms in **boldface** are defined in the glossary.

A special typeface indicates characters you should type.

```
It looks like this.
```

Other publications and resources

Information about the X Window System is widely available from a variety of sources.

Tutorial

For a tutorial on starting a server and running standard clients under the UNIX operating system, read *X Window System User's Guide, Volume Three,* by O'Reilly & Associates, Inc., Newton, Massachusetts, 1988. This document is based on release 2 of X11, but still contains much useful information, including command reference pages for clients and an appendix illustrating standard fonts.

Technical

For X programmers, three useful books available from O'Reilly & Associates, Inc. are the *Xlib Programming Manual, Volume 1;* the *Xlib Reference Manual, Volume 2;* and the *X Toolkit Programmer's Guide.* These books are available in most technical bookstores.

A new book by Robert W. Scheifler, James Gettys, and Ron Newman, the principle designers of X, is *X Window System: C Library and Protocol Reference,* published by Digital Press. This book is available from Digital Equipment Corporation, MIT, and can also be found in technical bookstores.

MIT has published the following documents, which you can order directly from them:

- *Using the X Toolkit*
- *X Toolkit Widgets*
- *X Toolkit Intrinsics*
- *Inter-Client Communications Conventions Manual*

To obtain these books, contact MIT at

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

X11 for A/UX

Three documents exist for those interested in the A/UX version of X. (A/UX is a System V–based version of the UNIX operating system for the Macintosh.) The X user's guide, available from Apple, is called the *X11 User's Guide for A/UX*. The other two documents are the *X11 Reference for A/UX* and *Getting Started with X Window System for A/UX,* which includes installation instructions.

For a complete description of A/UX manuals, as well as part numbers and ordering information, see the *Road Map to A/UX* published by Apple Computer.

MacX for A/UX

If you are interested in running MacX on A/UX, you will need the *MacX for A/UX Supplement* to this user's guide that explains how to launch MacX under A/UX and describes other different or additional features. Ask your Apple representative or an authorized Apple dealer for details about obtaining MacX for A/UX.

MacX for VMS

MacX can also communicate with VAX™ computers running VMS™ and display VMS-based clients. The related documents, written by Apple Computer and published by Digital Equipment Corporation, are *DEC LanWORKS for Macintosh: MacX User's Guide* and *DEC LanWORKS for Macintosh: Network Services User's Guide.*

Usenet

Subscribers to Usenet can join or monitor the news group, comp.windows.x, an on-line forum for news and discussion of the X Window System. Ask your network administrator for information about Usenet.

CHAPTER 1

MacX Overview

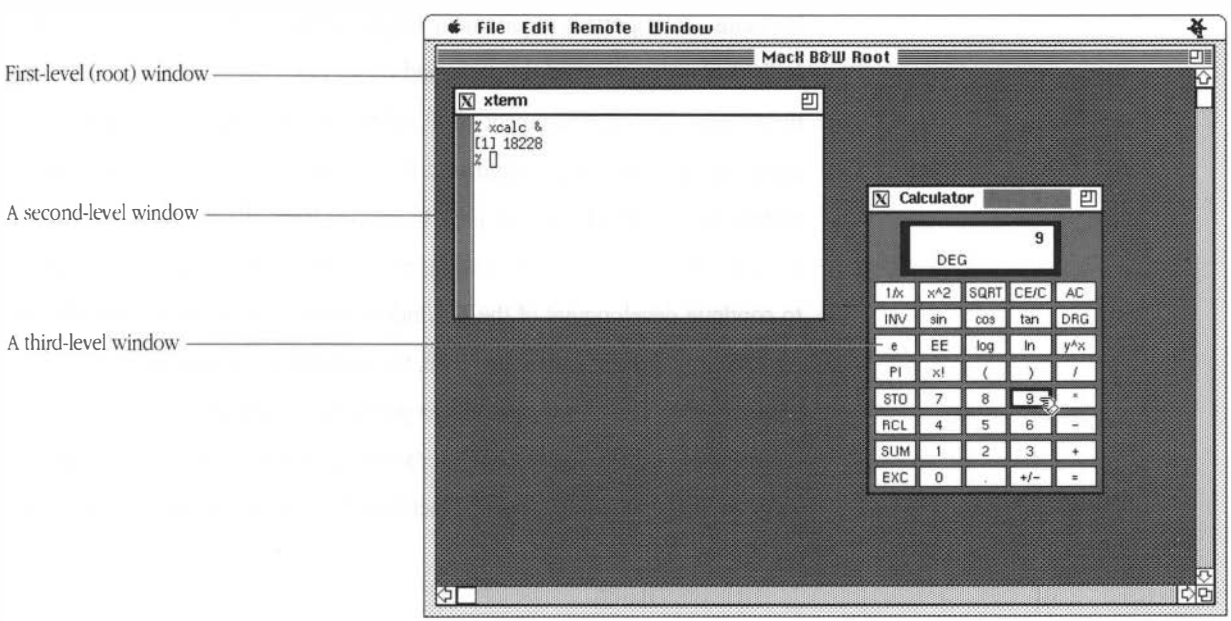
In May 1983, Project Athena and the Laboratory for Computer Science at MIT embarked on a project that would enable people to work with computers in an entirely different way. Students needed to run microcomputer applications, like spreadsheet and word processing programs, while simultaneously running programs on different hosts in remote locations. An important requirement was the ability to display graphics. The individuals who assumed this task devised an innovative solution in just three years. Their development effort, which included input from major computer manufacturers, culminated in the introduction of the X Window System, version 10 release 4, in 1986. In January 1988, MIT formed a consortium with most of the leading computer and workstation companies to continue development of the X Window System and to have it adopted as an ANSI standard. In November 1988, the consortium distributed version 11 release 3 of X. This chapter introduces you to X11R3 and gives you an overview of MacX™, Apple Computer's version of the X server, the specialized software that enables an Apple® Macintosh® computer to run X applications.

What is the X Window System?

The X Window System, or X for short, is a network-based graphics system designed to create an enhanced computing environment using the capabilities of desktop computers, like the Macintosh. X draws graphics, creates **windows** to display and run several applications simultaneously, and enables you to access X applications over a network from different makes of computers without being affected by incompatible hardware or operating systems.

As Figure 1-1 illustrates, X is also called a *window system,* or *windowing environment,* because it displays applications in windows on your **screen.** Essentially, this process converts your screen into a lot of miniature screens in which you can perform many activities rather than just one at a time. You can move, resize, and stack these windows, as well as reduce them to icons—a handy way to set aside windows without closing the applications in them. What's more, an application can also create its own windows, but the application controls how they are manipulated and they must remain within the boundaries of the application's window.

■ **Figure 1-1** Some typical X windows



In X parlance, an application is called a *client*.

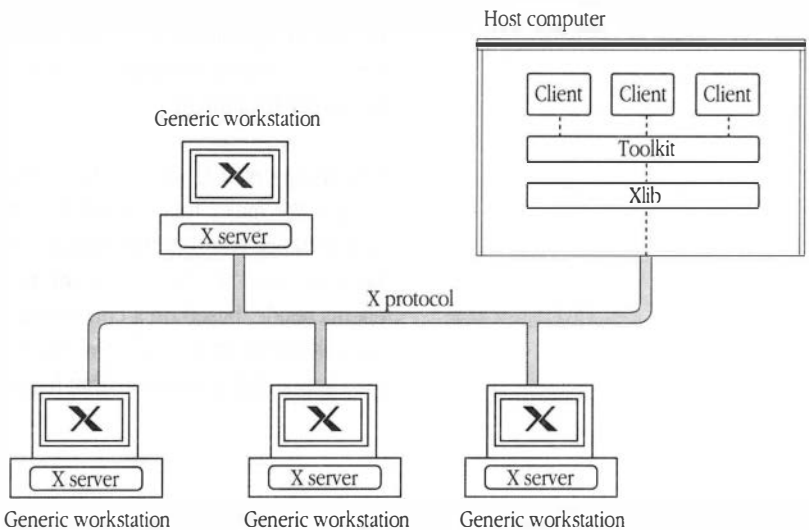
Although you can run X applications on your own microcomputer under an operating system that supports them (such as A/UX®), the primary reason for using X is that it lets you access programs remotely, over a network, and ensures that they function smoothly and reliably, as if the network weren't there. X also isolates different hardware architectures and operating systems that would prevent your machine from interacting with another machine in order to run an X application. By making the hardware appear compatible, the same program can run on any make of workstation or microcomputer in the network. Thus, you can run an application on one machine and display the results on another. Hardware compatibility also facilitates balancing the workload among network hosts by enabling them to share files. Should one host get overloaded, people can shift to other hosts to run their applications. Finally, programmers see X as a great advance because it frees them from having to rewrite programs in order to transfer them from one type of computer to another.

The designers of X realized these enhanced capabilities through a design that assigns the major functions of X to individual components. X functions well in a network environment because these components are distributed between your Macintosh and the remote host computers on which the clients reside, based on a concept called the **client/server model.** Knowing something about the different parts of X and how they work together will give you a clear picture of what happens when you run an X application.

X Window System structure

X is composed of two separate but interrelated groups of software that communicate through a protocol. One group of programs supports clients (X applications), and the other controls the desktop computer on which the clients are displayed. These programs can work together on the same system if clients are installed on the machine that displays them, or they can be separated over a network if the clients are located on different hosts. Figure 1-2 illustrates the relationship of these different components, formally referred to as the client/server model.

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



A *display* is a workstation or desktop computer with a keyboard, a pointing device, such as a mouse, and one or more bit-mapped screens. X will support a mouse with up to five buttons.

As depicted in the illustration, the X server is always located in a **display,** whereas the two application libraries—Xlib and the X toolkit—reside with the clients: either on different hosts or in your own computer. At first glance, this arrangement seems backwards to people who view work stations as the *clients* of a *server,* such as the file server that contains all the programs they use. The file server enables multiple workstations to share the same application simultaneously. In X, however, the workstation contains the server so that a person can run multiple clients simultaneously by displaying each one in a window on the screen. In effect, the clients are *sharing* the workstation.

The following paragraphs briefly describe the main functions of the basic components of X.

X protocol

X Window System protocol is a low-level graphics description language used by clients and the X server in your desktop computer to exchange information. This protocol establishes rules for composing and transmitting packets of instructions used to draw graphic images and windows, transmit text, specify colors and fonts, notify when events (such as a pressed key or the mouse click) occur, and related matters.

X server

The X server is a multilayer program that runs in your desktop computer and controls the flow of information to and from the keyboard, mouse, and screen (or screens). It acts as a switchboard, relaying information from different clients to the appropriate windows on the screen and from your machine to the appropriate client. The language used in this information exchange is the X protocol. Beyond routing messages, the server performs other communications-related services, such as maintaining communications links with each client, interpreting network messages, and assembling and disassembling X protocol request packets. The server also draws graphic images and windows; installs fonts; tracks the cursor and maintains its color, size, and shape; maintains a map of color values; and requests services from the operating system.

Xlib

Xlib is a library of C routines—program segments written in the C programming language—each of which performs a certain task when called by a client program. For example, these routines enable a client to communicate with a server, request that a window be created, draw graphics, respond to output from the keyboard or mouse, and so forth. Xlib calls are translated to X protocol requests before being transmitted to the server.

Window manager

Like any client, the window manager is just another program written with the X library, except that it has special authority to control the layout of windows on the screen. This specialized client enables you to move, resize, and stack windows; create new windows; and shrink windows to icons. Some window managers enforce a **layout policy** that specifies the sizes and positions allowed for windows and icons. Programmers must ensure their clients comply with these policies.

X toolkit

An X toolkit is a library of subroutines that employs several Xlib drawing routines and other functions to produce larger graphic objects, such as menus and scroll bars. Programmers use these objects, called **widgets,** to create the images you see on your desktop. Other toolkit subroutines manage and destroy widgets. A toolkit also furnishes tools called **intrinsics,** used to create new widgets.

A brief look at how X works

Because the way you use X depends on the type of machine and operating system you have, giving step-by-step instructions for starting up and using X is beyond the scope of this guide. However, a general overview will give you an idea of what is common to all implementations of X. It is helpful for readers unfamiliar with X.

In order to display clients on your screen, you must first start up the X server which, in a standard X configuration, creates a background, or **root window,** that covers the entire screen. Typically, your *.login* or *.profile* file will contain a command to start up the window manager, or else you must type the command yourself. At this point, you can issue commands to run clients on your own machine, or you can log in to other hosts and issue remote commands (or their equivalent) to run clients. The client responds by sending X protocol packets requesting the X server to draw a window of a certain size in a certain location (although you can specify a size and location yourself). The X server interprets these requests and relays them to the window manager. If the window manager approves, a window appears on your screen. Using this procedure, you can display as many clients as your machine will accommodate.

Windows and their hierarchy

X supports displays with multiple screens. These screens can be color or monochrome.

As Figure 1-1 shows, windows in X are organized in a hierarchy. The root window comprises the first level. In traditional X systems, the root window is always visible and covers the entire screen. (If you have more than one screen connected, a root window appears in each one.) All other windows are subordinate to the root window. Occupying the second level in the hierarchy are the top-level client windows: the windows that appear when you start up a client. For example, in Figure 1-1, the terminal emulator (*xterm*) and the calculator (*xcalc*) clients are displayed in second-level windows. These are the windows you can resize, move around, and manipulate in other ways, according to the dictates of the window manager. Finally, clients can create their own third- or lower-level windows; however, the clients control the manipulation of these windows. In Figure 1-1, the keys on the calculator are third-level windows.

- *Window disguises:* Windows don't always have to look like windows. Since, in X, a window is a rectangular object that can be made sensitive to a mouse pointer, clients often draw tiny subwindows that look like menus, buttons, scroll bars, and other shapes. When a pointer enters or the mouse is clicked in one of these subwindows, the client receives the feedback.

Another important point about the window hierarchy is that higher-level windows impose restrictions on lower-level ones. Since a window is subordinate to the window that created it, think of the subordinate as the child and its creator, the parent. In X, every window has a parent except the root window. Accordingly, client windows are children of the root window, but are the parents of any lower-level windows their clients create. A child always remains within the boundaries of its parent window. Even though a child window can be larger than its parent, you will see only the part bounded by the dimensions of the parent window. The rest is invisible, unless the client lets you move the child window around to reveal its hidden parts.

Now that you have a general idea about what X is and how it works, read the rest of this chapter to learn about MacX, Apple's X Window display server for a Macintosh running the Macintosh Operating System.

Introducing MacX

MacX is an X server for the Macintosh. By permitting a Macintosh using the Macintosh Operating System to run X applications, MacX combines the advantages of X with Macintosh advantages and features. The ability to cut and paste text between Macintosh and X applications exemplifies the benefits of this combination. What's more, MacX adds new features to standard X and makes many of the more cumbersome and tedious methods for doing things—such as changing fonts and adding new colors—quicker and easier. Notwithstanding these enhancements, MacX fully conforms to the standards published in the *Inter-Client Communications Conventions Manual* (ICCCM) and to version 11, release 3 of X with some release 4 enhancements.

What you get with MacX

In addition to the X server, MacX software includes:

- **MacX Window Manager**

  The MacX Window Manager allows you to use X windows as you would Macintosh windows. It also supplies a selection of five window styles and lets you convert windows into icons. This window manager handles windows in **rootless style,** a style in which the root window is invisible. MacX provides the option to display or hide the root window.

- **MacX Font Director**

  The Font Director displays and sorts X and Macintosh fonts and compiles Adobe Bitmap Distribution Format **(BDF)** files into X fonts. It also permits you to assign **aliases** to long, unwieldy font names.

- **MacX Fonts folder**

  This folder contains files for a variety of type fonts. Each file contains a single font-family variation. For example, one file could contain Courier 12 point bold and another file, Courier 14 point bold.

- **MacX Colors file**

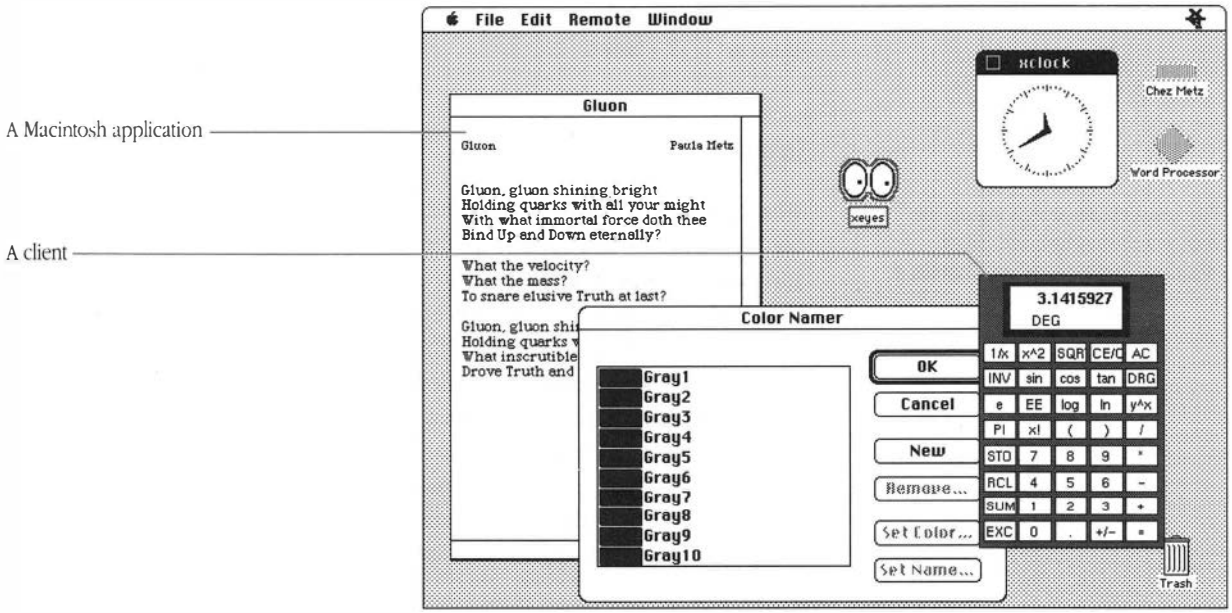
  This file contains an array of colors that you can add to, modify, or delete from using the MacX Color Namer.

X with a Macintosh flair

MacX is designed to blend in with your Macintosh working environment, adding another dimension to your desktop without impeding or limiting Macintosh facilities. While MacX is running, you can use **MultiFinder®** and your desk accessories. You can run Macintosh applications and various clients simultaneously, as shown in Figure 1-3. Functions such as cut, paste, save, and quit operate the same way, and you can cut and paste text and graphics between Macintosh and X applications. MacX uses the Macintosh method for handling windows, but gives you the option to use a different window manager, too. Like standard X products, MacX supplies a variety of colors and fonts, but makes them easier to use. For instance, the Color Namer shows an example of each color available so that you don't have to imagine what they look like. You can create new and remove existing colors, change color names, and copy color names into X commands in client windows. With the MacX Font Director, you can use Macintosh and X fonts interchangeably and assign aliases to long, cumbersome font names. The Font Director also provides a variety of methods for sorting fonts.

MacX application icon

■ **Figure 1-3** Clients and windows on a typical desktop



MacX document icon

Each time you start up MacX, it creates a **settings document** in which you can store commands and preferences. If you name and save this file when you quit MacX, a MacX document icon labeled with that name appears on the screen. You now have your own personalized X environment to which you can return any time by clicking the file icon. Through these individual settings documents, MacX enables you to tailor X environments to different hosts, clients, types of tasks, or working styles.

What you see on the screen

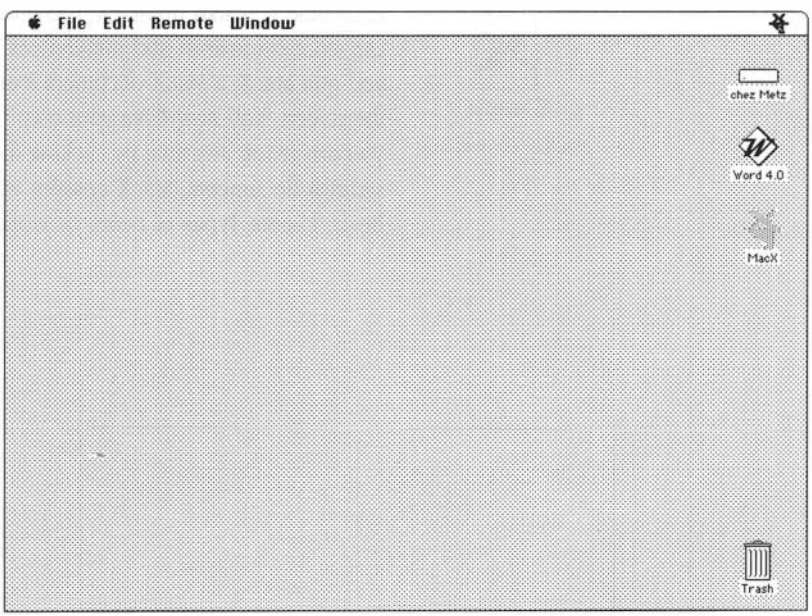
MacX offers an alternative to the standard X root window display. Although the option for displaying the root window exists, MacX is preset to hide the root window and display the Macintosh desktop instead. Figures 1-4 and 1-5 illustrate both cases.

Invisible root window

For instructions on changing the root window setup, see "Root Window Preferences" in Chapter 4.

When you launch the MacX application, no root window appears. Instead, the "About MacX" box appears on the screen until MacX finishes launching and then the MacX menus appear in the menu bar at the top of the screen. If MultiFinder is running, a MacX icon appears in the upper-right corner. As Figure 1-3 shows, the desktop is visible. Since a root window does not appear, this style of operation is called rootless. MacX, by default, hides the root window unless a client requires a visible one or you select a black and white or color root window from the Window menu. MacX defaults to rootless style so that you can manipulate X windows like real Macintosh windows. Since your desktop is readily accessible, rootless style also makes using other Macintosh applications and desk accessories easier.

■ **Figure 1-4** MacX on the desktop with the root window invisible

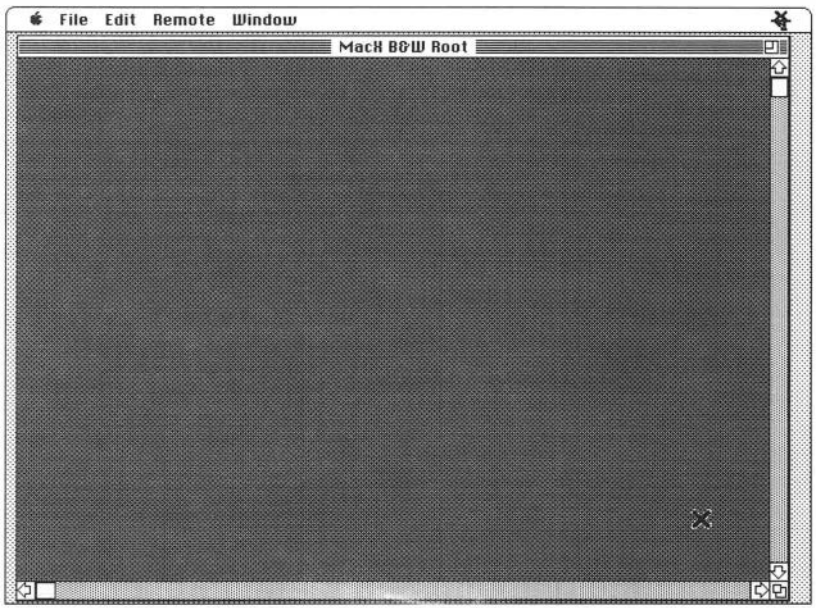


Visible root window

For people who prefer the traditional X Window System environment, MacX provides a different style of operation, called **rooted,** which displays a root window. In this style you must run a different window manager, such as *twm,* because the MacX window manager runs only in rootless style.

Figure 1-5 shows what a desktop looks like with the root window visible, assuming the Macintosh has just one screen attached. (Read the next section, "Using Multiple Screens," if you have more than one screen.) Usually, the window covers the entire screen. Although labeled *root,* this window is really a Macintosh window through which you can see the actual root window stored in memory. This arrangement lets you move and resize the Macintosh window without disrupting any clients running on the actual root window. For more information on using a visible root window, see "About Root Windows" in Chapter 4.

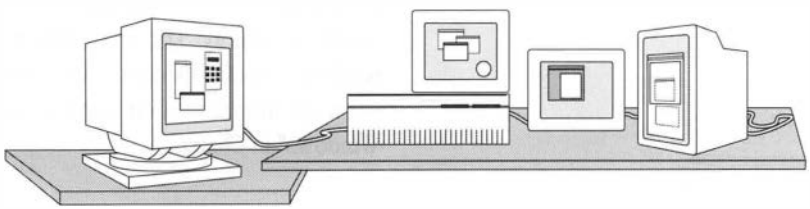
■ **Figure 1-5** MacX on the desktop with the root window visible



Using multiple screens

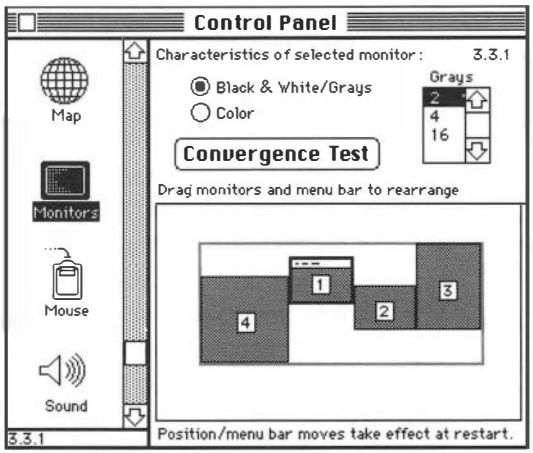
In accordance with X specifications, MacX supports single or multiple screens. The screens can be color, gray scale, or monochrome in any combination. Figure 1-6 illustrates a multiple screen setup. MacX distinguishes the types of screens attached to your Macintosh by checking the System file.

■ **Figure 1-6** A Macintosh with multiple screens attached



If you are familiar with using multiple screens on a Macintosh, you know that when you move a window halfway between two screens nothing in the middle of the window gets lost, despite the physical separation between the screens. The screens act as if they are continuous because the Macintosh Operating System joins multiple screens edge-to-edge into one big screen. (This feature is unique to Macintosh computers.) To see what this arrangement looks like, choose the Control Panel from the Apple (🍎) menu and click the Monitors icon. You will see a diagram similar to the one in Figure 1-7. The screens in the Control Panel should be arranged like the monitors on your desk, except with no spaces in between them.

■ **Figure 1-7** A Control Panel diagram of multiple screens



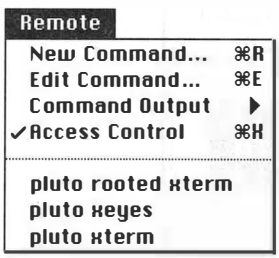
Now imagine a rectangle drawn around all of the screens in your Control Panel, like the red one in Figure 1-7. You will notice some blank areas if the screens are different sizes. MacX treats the entire contents of this rectangle, *including the blank areas,* as one continuous screen on which it can display windows. That's why part of a window can disappear when you move it from a taller screen to a shorter one. For example, if you moved a window from the top of screen 3 in Figure 1-7 straight across to the blank area above screen 2, which is shorter, part of the window would not appear until you pulled it down into screen 2. The window was always there, but MacX just couldn't re-create it in thin air.

In MacX, the rooted and rootless styles coexist. When issuing a command to display a client, you can specify whether you want it to appear in rooted or rootless style on either a color or monochrome screen by adding a *screen number* to your command. (See "Specifying Screen Numbers" in Chapter 3 for a complete explanation of screen numbers.) Chapters 2 and 3 describe how to issue remote commands.

Overview of MacX features

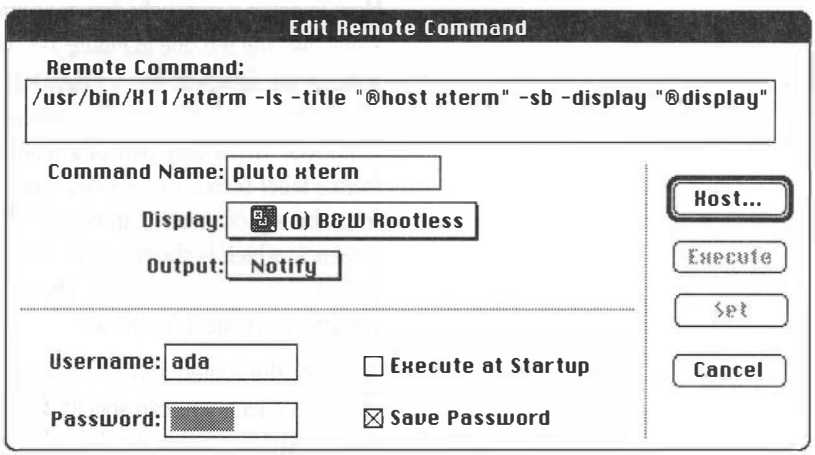
This section introduces you to the functions available in the MacX menus. Besides supplying all the tools found in standard X products, MacX enhances many of them by extending their capabilities or making them simpler to use.

Remote commands



When MacX operates under the Macintosh Operating System, instead of the A/UX operating system, X applications cannot run directly on the Macintosh. To display clients on your screen, you must issue a remote command to a host where the clients you want to use reside. Figure 1-8 shows the Remote Command window in which you type your command, user name, password, and other options. Choosing New Command from the Remote menu, shown in the margin, brings up the Remote Command window.

■ **Figure 1-8** The Remote Command window



Set retains a command without executing or saving it. You can save it when you close your settings document.

After entering a remote command, you have the option to execute, set, or cancel it. Clicking Execute or Set enters the command in the Edit Command list (not shown) and at the bottom of the Remote menu. Commands are listed alphabetically.

The next menu item, Edit Command, enables you to change an existing command. Edit Command is dimmed if no commands exist. The Command Output submenu, if not dimmed, displays a list of commands that are currently running or have command output. Choosing a command in the output list displays an output window that shows any information generated by or about the command. The checked menu command, Access Control, is a security precaution that prevents surreptitious access to your MacX server. For basic information about remote commands and step-by-step instructions for displaying a terminal emulator, like xterm, on your screen, read Chapter 2, "Getting Started."

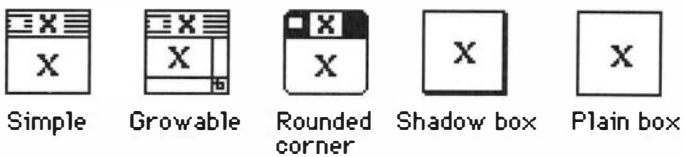
For advanced information about remote commands, read Chapter 3, "Using Remote Commands."

- *MacX for A/UX users:* If you are running MacX on A/UX, you can issue remote commands, too. Read the *MacX for A/UX Supplement* for instructions.

Window management

Use the Window menu, shown in the margin, to shrink client windows to icons, choose a different style for a client window from the five shown in Figure 1-9, and move or resize a window that doesn't have a title bar or **size box.** To learn more about these features, read "The Basics of Using Windows" in Chapter 4.

■ **Figure 1-9** MacX window styles



A window icon

Choosing Iconify from the Window menu lets you shrink a client window into an icon in order to move it out of the way. Iconifying does not kill a client or suspend what it's doing. Double-clicking the icon returns it to normal size and position.

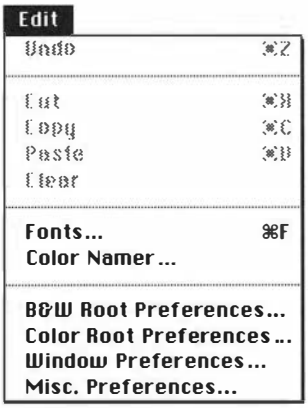
Temporarily Adorn allows you to move and resize a window that does not have a title bar or a size box by temporarily converting the window to the style that has these elements (see the second window from the left in Figure 1-9). You have one opportunity to either move or resize the window, after which it reverts to its former style.

The command, Close Window, changes depending on the type of window that is in the foreground (what we call "active" in the Macintosh world). Close Window closes a MacX window, such as the Remote Command window or the Color Namer window if it frontmost. If a client window is frontmost, the Window menu displays the command Kill Client. Since this command causes an abrupt disconnection, read "Closing Clients" in Chapter 2 before killing a client.

When windows open, their names appear in alphabetical order at the bottom of the Window menu. A bullet identifies the window that is frontmost (called *active* in Macintosh lexicon). For example, the Window menu on the previous page shows that the Color Namer window is both active and the only window open. When more than one name appears in this list, you can make another window active by selecting its name with the mouse.

The last two commands in the Window menu enable you to show or hide a black and white or color root window. When a root widow is visible, the word *Show* changes to *Hide*. Hiding a root window also hides any client windows running in it.

Cut, Copy, Paste, and other editing commands

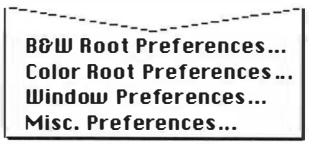


MacX enables you to cut, copy, and paste text between a Macintosh application or desk accessory and a MacX window that contains text, such as a Remote Command window or an xterm window. Similarly, MacX allows you to cut, copy, and paste graphics in **PICT** format as well. As in any other Macintosh application, these functions are located in the Edit menu, pictured at left, and perform the same way. Because clients have their own commands for cutting, copying, and pasting, you cannot use Macintosh commands (Command-X, Command-C, Command-V) in a client window. However, MacX does track when a client cuts or copies text and records the selection in the Clipboard. Once something is in the Clipboard, you can paste it into any Macintosh file or document, such as a MacWrite® document, for example. To paste text or graphics from a Macintosh file to a client window, use the client's command for pasting or inserting (not Command-V). MacX transfers the contents of the Clipboard to a cut/paste buffer, the equivalent of a Clipboard in the X world, and then to the point selected in the client document.

The Edit menu also contains Undo and Clear, which both perform their normal Macintosh function, as well. Undo undoes the last change, but functions only with desk accessories. MacX does not support it. Clear erases a selected item without saving it to the Clipboard.

See the next section for an explanation of the preference items at the end of this menu.

Preferences



A list of preferences, located at the bottom of the Edit menu, allows you to select default settings for the items shown in the left margin. Choosing either of the first two items displays a Rooted Screen Setup window, which allows you to set the size of the root window.

Normally, most people need not alter the default root window preferences. Unless you are an experienced user of X or plan to run memory-intensive color clients on machines with 2 megabytes of **RAM** or less, you would be better off not changing the default root window size. However, those who need more information should read "Root Window Preferences" in Chapter 4.

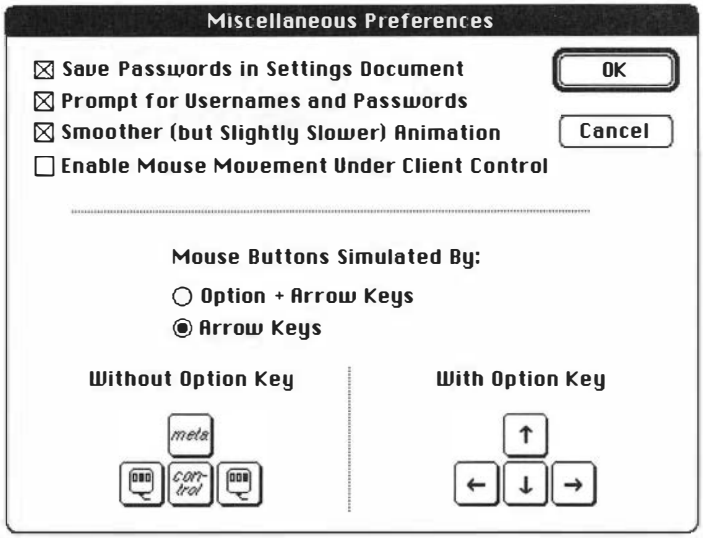
Client Specified icon

Window Preferences, the next item, permits you to choose a default window style for client windows from the ones shown in Figure 1-9. The next time you start a client, its window should appear in that style. One remaining choice, called *Client Specified,* is not actually a window style, but enables the client to indirectly influence the window style selected. If you choose Client Specified, MacX will convert a client's border width request into one of the five window styles. See "Window Styles" in Chapter 4 for a further explanation of this case. The window styles in Figure 1-9 apply only to client windows in rootless style. In rooted style, different window managers control the style, or **adornment,** of client windows.

Figure 1-10 depicts the last entry in the Edit menu, Miscellaneous Preferences, which contains five options. Clicking the first check box saves passwords entered in Remote Command windows so that you don't have to retype your password every time you execute a remote command. The passwords are saved in the settings document and are encrypted to provide an extra measure of security adequate for most business needs. In cases where passwords aren't saved in remote commands, clicking the second check box will cause MacX to prompt you for these passwords when you execute such commands.

The next option for smoother animation should usually stay checked. See Appendix A for information on this option and the one for enabling mouse movement. The last option lets you choose whether to use the Option key with the Arrow keys to do scrolling or to simulate extra mouse buttons and modifier keys. See the section called "Mouse Differences" in Chapter 2 for an explanation.

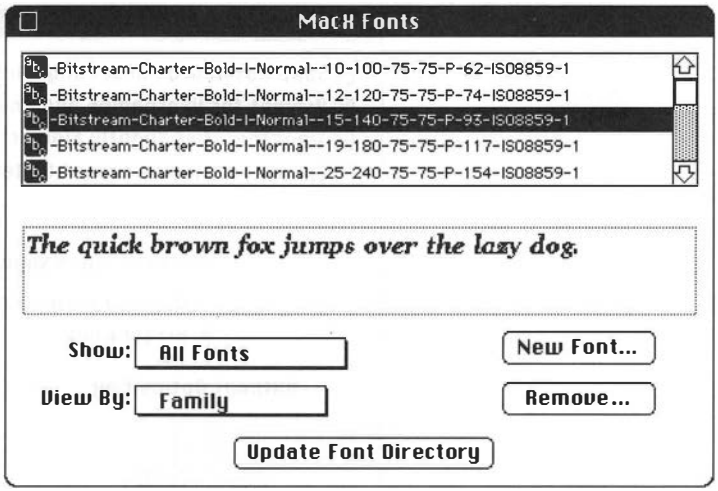
■ **Figure 1-10** The Miscellaneous Preferences window

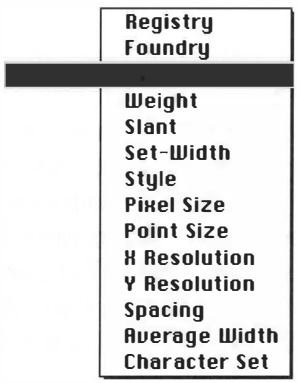


Fonts

Choosing Fonts from the Edit menu brings up the Font Director window pictured in Figure 1-11. This window displays a list of all the fonts available for use with clients. These can be either X Fonts or Macintosh fonts, which are distinguished by a little icon at the beginning of the font name. The Font Director also allows you to create aliases as substitutes for font names. Since X font names tend to be long and unwieldy, you can create aliases that are shorter and easier to remember. Aliases also have a distinguishing icon. Chapter 5 provides more details on fonts and aliases.

■ **Figure 1-11** The Font Director window



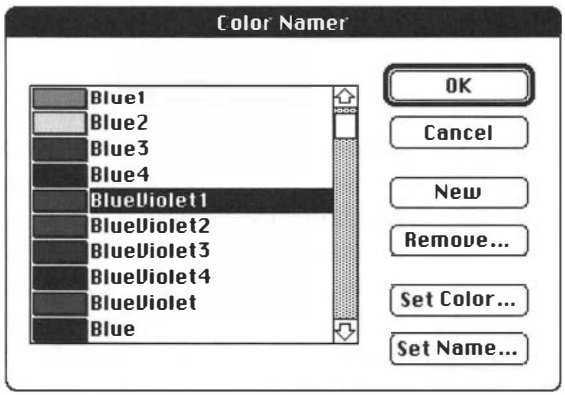


The Font Director also allows you to sort fonts in a variety of ways. In Figure 1-11, the field labeled View By is actually a pop-up menu containing a list of sorting methods. Simply select an item from the list and the Font Director will resort the fonts accordingly. The Font Director can also compile fonts from Adobe Bitmap Distribution Files (BDF). For more information about the Font Director and compiling BDF files, read Chapter 5, "Managing Fonts and Color."

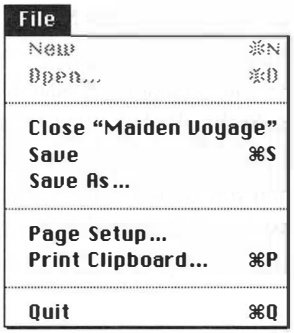
Color Namer

Choosing Color Namer from the Edit menu displays the Color Namer window depicted in Figure 1-12. This window displays a sample of every color in the MacX Colors file, providing you with a complete visual inventory of the colors you can use with color clients. Each color has a name that you can modify or copy to a command in a client window. You can also create new colors and remove existing colors. When you do, the Color Namer automatically resorts the colors by name, so they remain in alphabetical order. See Chapter 5 for more information about using the Color Namer.

■ **Figure 1-12** The Color Namer window



Save, Close, Quit, and other File menu commands



Like the Edit menu, the File menu contains a number of commands that function in the normal Macintosh way. The New command creates a new settings document in which you can create and save new commands and preference settings. Open allows you to open an existing document, provided that one is not currently open. The Close command closes the document currently open. If the file is new, the name "Untitled" appears. Click Save to save your settings or Save As to save your settings and name the file. However, should you forget and select Close "Untitled," a standard Macintosh dialog box inquiring "Do you want to save your changes?" appears anyway. Read Chapter 2 for more information about the settings document. Chapter 2 also explains the procedure for quitting from MacX, the last command in the File menu.

Printing commands

The Page Setup and Print Clipboard commands in the File menu are used for printing. Print Clipboard is dimmed unless you have copied something to the Clipboard. Use Command-C (⌘-C) or Command-X (⌘-X) to copy or cut text or graphics from a Macintosh application or a MacX window (such as the Color Namer). In a client window, use the client's method for copying and cutting. MacX intercepts these selections and pastes them into the Clipboard as well. Before printing, choose Page Setup and turn off Text and Graphics Smoothing. These options weren't designed for bit maps and as a result degrade the appearance of the printed graphic image. Leave Font Substitution on, however, as it preserves spacing better and uses a more readable font (Courier versus Monaco 9). Choose Print Clipboard to print. MacX prints text or PICT-style graphics. When printing text, MacX includes spaces, carriage returns, and line feeds, but omits any formatting.

CHAPTER 2

Getting Started

BEFORE READING THIS CHAPTER, BE SURE TO INSTALL YOUR COMMUNICATIONS software and MacX by following the instructions in the *MacX Installation Guide.* Once MacX is up and running, you'll be ready to read this chapter and learn the basics about MacX.

This chapter covers the minimum you need to know to function in an X environment using a Macintosh. You'll learn how to open MacX, log in to a host and start a terminal emulator on your screen, log off the host, and, finally, close MacX. Once you feel comfortable with these fundamentals, you can consult the remaining chapters to customize your X environment and learn about more advanced features. However, please note that some areas, such as handling fonts, require technical expertise. Consequently, to avoid problems, consult your resident X expert if you are not technically oriented and are not sure what to do.

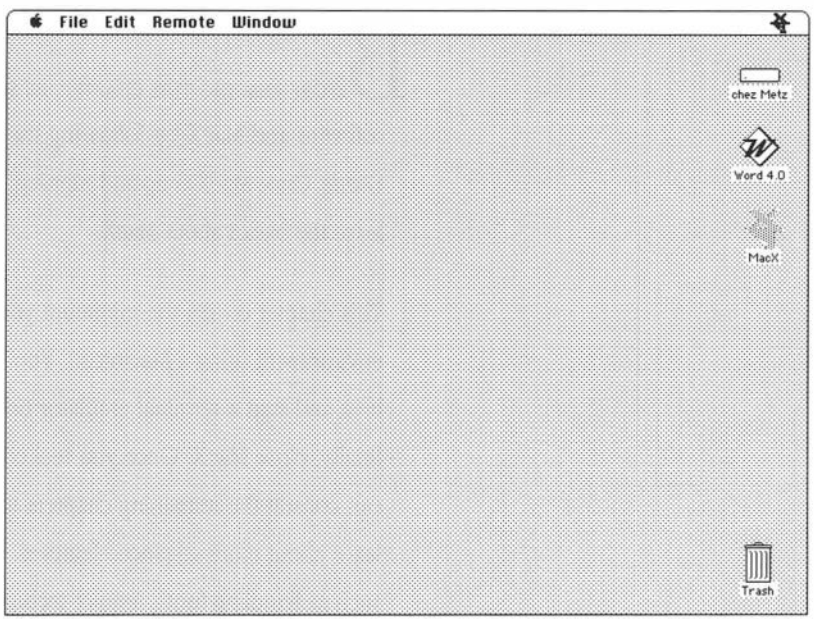
MacX for A/UX users: If you are running MacX on A/UX, read the *MacX for A/UX Supplement* before reading this chapter. Procedures for opening MacX and using remote commands are different.

Opening MacX

Application icon

To open MacX, double-click the MacX icon, shown at left, or click the icon once and choose Open from the File menu. Your desktop should look similar to the one shown in Figure 2-1. You should see the four MacX menu headings in the menu bar at the top of your screen and, if Multifinder is active, the MacX icon at the far right. If you have not already done so, read Chapter 1, "MacX Overview," for an explanation of the menus and an overview of MacX.

■ **Figure 2-1** The MacX desktop



A document icon

Although you don't see a window for it, MacX always opens a new settings document when it starts up. This document stores your whole environment and everything you do in it, that is, every remote command you create, every preference, and so forth. When you quit, MacX will prompt you to name and save your settings document. If you don't, MacX will discard everything you've set up. If you do, MacX will create a document icon labeled with the name you have entered. Later, you can double-click the icon and return to that environment exactly as you left it. This feature enables you to customize environments for different hosts, clients, types of tasks, or working styles. For example, you could create an environment for doing budgets—in which a spreadsheet client automatically opens in the middle of your screen and a calculator appears in the upper-right corner—and save it in a settings document called "budget."

Making a fast exit

If you want to exit from MacX and you haven't made any changes that you want to save, such as creating a remote command, simply choose Quit from the File menu or press ⌘-Q and MacX will close. If you have made changes, the dialog box "Save changes before closing?" appears. See the section called "Quitting MacX" at the end of this chapter for further instructions, if you need them.

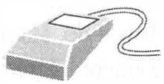
Mouse and keyboard differences

Before proceeding further, you should know about the differences between the Macintosh mouse and keyboard and the standard mouse and keyboard specified for X11.

Mouse differences

Because the Macintosh uses a one-button rather than a three-button mouse, you cannot issue commands with the mouse like you normally would in other X environments. As an alternative for the missing two buttons, use the arrow keys instead, as shown in Figure 2-2.

■ **Figure 2-2** Substitute mouse buttons

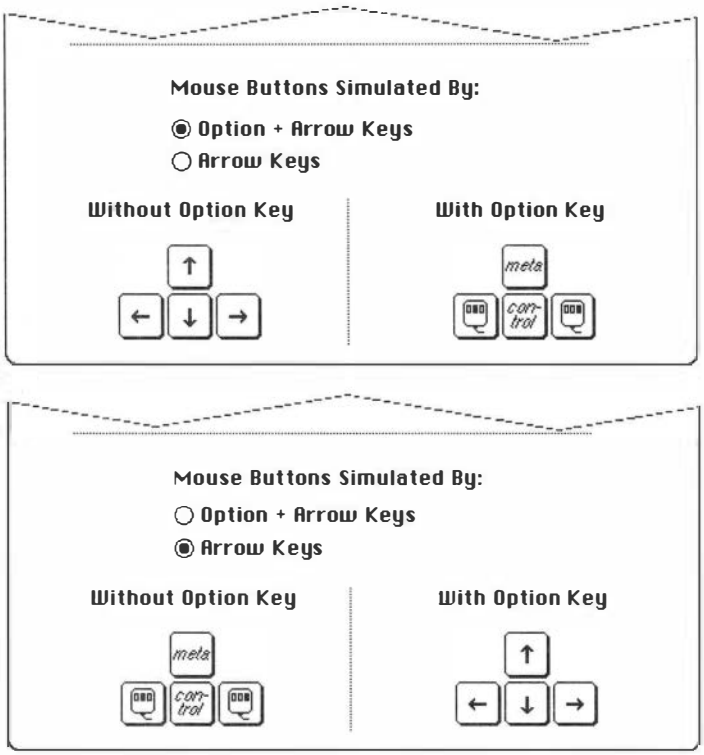
Macintosh key	MacX function
	Left mouse button
	Middle mouse button
	Right mouse button
	Meta modifier (mod1)
	Control key

To use the arrow keys for their normal purpose, hold down the Option key before pressing one of them.

❖ *Which mouse button?* References to the mouse button in this document (for example, "Press the mouse button...") apply to the mouse button on the Macintosh mouse, which serves as the left mouse button in MacX. The other mouse buttons are specifically identified as the middle mouse button or the right mouse button.

If you would rather reverse this procedure—that is, press the Option key together with an arrow key to get extra mouse buttons—you can select the Option + Arrow Keys button in the Miscellaneous Preferences window, as shown at the top of Figure 2-3.

■ **Figure 2-3** The arrow keys preference option



You'll find Miscellaneous Preferences at the end of the Edit menu.

Keyboard differences

The Macintosh Plus keyboard has no Control key, so you must use the Command key (⌘) instead. As a result, you can no longer use the Command key to select menu items from the keyboard, such as ⌘-X to cut and ⌘-V to paste, while MacX is running.

Substituting keys for mouse clicks

MacX has a variety of dialog windows, dialog boxes, and alert boxes that it will display when you select a certain option or need to respond to an event. See Figure 2-4 for an example. These boxes contain a collection of fields, buttons, or both, which you must select or click. Sometimes pressing keys on the keyboard is more convenient than using the mouse to click a button, choose a field, or select the contents of a field. For such instances, Table 2-1 lists some substitute keys and the action each performs.

■ **Table 2-1** Keyboard substitutes for mouse clicks and selections

Key(s)	Action	When to use
Y, O	Yes or OK	Use to specify *yes* or *OK* in a dialog or alert box that has no text or scrollable fields. If the box has text fields, the number or letter is interpreted as text rather than as a command and is entered into the field that has the cursor in it. If the box has scrollable fields, the letters select a position in the field.
CR, Enter	Click button	The carriage return or Enter key. Use to select an outlined or default button in any dialog window, dialog box, or alert box, including those with text fields.
N, C	No or Cancel	Use to specify *no* or *cancel* in a dialog or alert box that has no text fields. If the box has text fields, the number or letter is interpreted as text rather than as a command and is entered into the field that has the cursor in it.
⌘ .	No or Cancel	Command-Period. Use to specify *no* or *cancel* in any dialog window, dialog box, or alert box, including those with text fields.

■ **Table 2-1** Keyboard substitutes for mouse clicks and selections *(continued)*

Key(s)	Action	When to use
Esc	Cancel	Escape key. Use to select the cancel button in any dialog window, dialog box, or alert box, including those with text fields.
Fwd. Delete	Deleting	The Forward Delete key. Deletes text to the right of the cursor. (This key appears only on the extended keyboard.)
Tab	Tabbing	Tab key. Use to tab to the next text field.
Arrows	Scrolling	The Up, Down, Left, and Right arrow keys. Use to move the cursor and to scroll vertically and horizontally in text fields. Fields can contain a maximum of 255 characters. The arrow keys will also move the cursor to a different insertion point.
Home, Page Up	Scrolling	The Home and Page Up keys. Use to scroll to the beginning of a text field or to the top of a list. (These keys appear only on the extended keyboard.)
End, Page Down	Scrolling	The End and Page Down keys. Use to scroll to the end of a text field or to the bottom of a list. (These keys appear only on the extended keyboard.)

Forward Delete key

❖ *Using letters and numbers to scroll:* The Color Namer and the Font Director, two features described in Chapter 5, won't interpret letters or numbers as yes, no, OK, or cancel actions. Instead, they interpret any letters or numbers typed as scrolling instructions. See Chapter 5 for more information.

Starting a terminal emulator

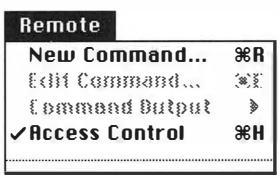
Application icon

Once you've opened MacX, you'll notice that your desktop doesn't look any different except for the different menu headings in the menu bar at the top of your screen. Since the presumption, in this section, is that you'll be creating a brand new X environment, be sure to click the MacX application icon, pictured at left, and not a MacX document icon. If you click a MacX document icon, someone will have already entered commands and set preferences that may have slightly or totally altered the initial look of the desktop described here. For example, the root window may be visible or some clients, like *xclock,* may suddenly appear. This section will not describe these nuances. Instead, it will show you a basic way to establish a connection with a remote host computer so that you can run clients and operate as you normally would.

After opening MacX, the next step is to create a remote command that logs on to a host, starts up a terminal emulator, and displays a window on your screen. You'll need to run a terminal emulator so that your Macintosh resembles a type of terminal recognized by your host. This section will cover an example of this procedure for a host running A/UX or ULTRIX™ (two versions of the UNIX® operating system).

Starting xterm

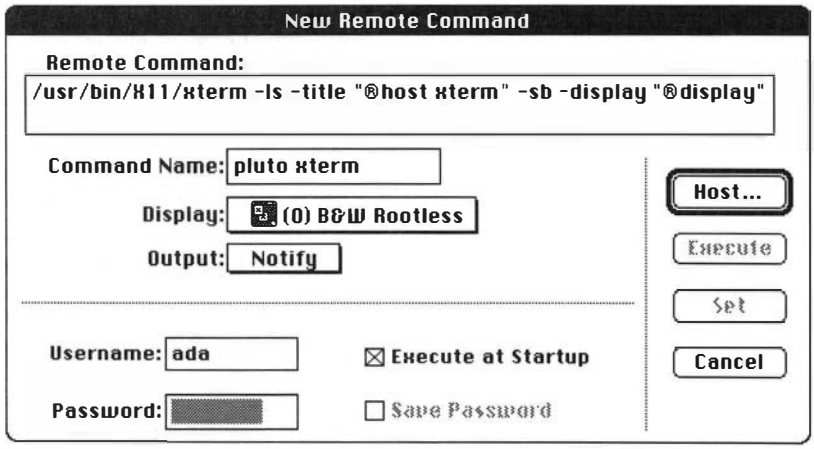
The *xterm* client is a terminal emulator that operates under UNIX. It enables your Macintosh to communicate with another Macintosh running A/UX, Apple Computer's implementation of UNIX System V developed by AT&T. A similar client, *dxterm* enables your Macintosh to communicate with a VAX running ULTRIX, Digital Equipment Corporation's implementation of 4.2 BSD UNIX. To start up xterm, follow these steps:



1. Choose New Command from the Remote menu or press ⌘-R.

The Remote Command window, illustrated in Figure 2-4, appears.

■ **Figure 2-4** The Remote Command window



2. Enter the remote command to start xterm.

Including the pathname, /usr/bin/X11/, is very important because it identifies the directory where xterm, and other clients, are located. Remember to include this pathname when issuing a remote command to start up *any* client.

Type the command exactly as shown in the Remote Command field in Figure 2-4. Begin the command with a slash (/) and type `/usr/bin/X11/xterm` as one word, with no spaces in between. Use straight (" ") quotation marks, as shown, rather than curly(“”) quotation marks so that multiple-word names are treated as a unit. When you have finished typing the command, press the Tab key. Do not type carriage returns; characters automatically wrap to the next line.

Your UNIX or host documentation should explain xterm and all its options; however, two of these options (®host and ®display) are MacX enhancements, which require explaining.

To type the registered mark, hold down the Option key and press *r*.

```
-title "®host"
```

This option includes the host name in the title bar of the xterm window that will appear on your screen. The option, ®host, is substituted for the name of the host you want to contact. You can enter the actual host name instead of ®host if you want to.

```
-display "®display"
```

The ®display option is substituted for the display station name or network address used by the host to identify your Macintosh. Alternatively, you can enter the actual name or address.

Putting these macros in remote commands is useful when you want to reuse a command for another host, because you don't have to change the host or display name in the command line.

- *Don't forget the man pages:* Most UNIX machines provide an on-line manual that describes UNIX commands and X11 clients. This information is located in the *man* directory. To display information on a particular client, type `man` and the client name. For instance, typing `man xterm` will display information about xterm.

3. Complete this xterm command by following the instructions for each field and check box listed below.

Command Name: Enter an abbreviated name for your command here. This name appears in the Edit, Execute, or Command Output submenus so that you can select it later when you want to edit, execute, or view command output.

Display pop-up menu

Display: This pop-up menu contains a list of four types of screens on which you can display your remote command. An illustration of this list appears at left. For this xterm command, select B&W Rootless if you have a monochrome monitor or Color Rootless if you have a color monitor.

The screen number (0 through 3) preceding each screen type represents a style (rooted or rootless) and and a type of screen (color or black and white). In Figure 2-4, the number zero (0) specifies a black and white screen in which the root window does not appear (rootless). MacX will display the xterm window on the largest screen that fits the screen type selected.

Output pop-up menu

Output: Select Save or Notify from this pop-up menu to retain any system or error messages generated in case the command doesn't work or gets interrupted by a problem with the host or the network. The Notify option triggers a beep and a small, flashing MacX icon to appear at the left end of the menu bar when MacX receives output from the host. See Chapter 3 for an explanation of the other output options.

Username: Enter, in this field, the user name assigned to you on the host receiving this command.

Password: Enter your password for the host receiving this command. Each character typed appears as a dimmed, gray box for security purposes.

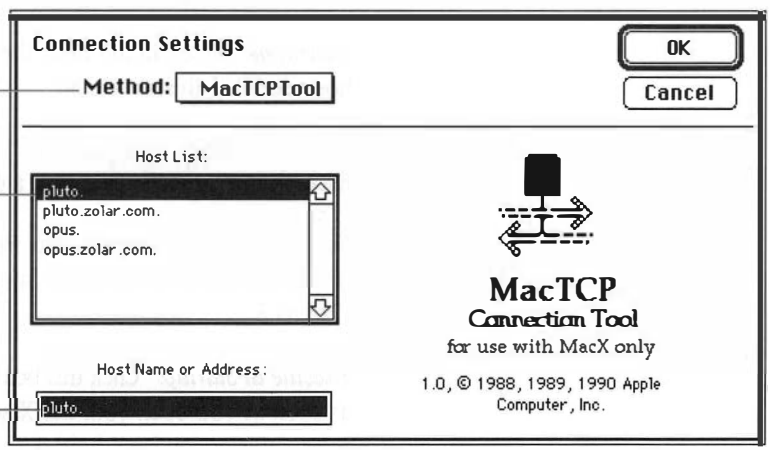
Save Password: This check box should be dimmed. If not, leave it unchecked for now. For more information, see "Saving Passwords" in Chapter 3.

Execute at Startup: Click this box so that the xterm window will appear as soon as you open your MacX settings document. If you want to bypass this option later, hold down the Option key while MacX is starting up. After the MacX startup window disappears, you may release the key and no xterm window will appear.

Host: Click this button to display the connection settings box illustrated in Figure 2-5. At the top left side of the box, a pop-up menu labeled Method contains a list of communications protocols, called **connection tools,** that enable you to communicate with hosts that contain clients (X-based application programs). Select the one appropriate for the host that will receive this xterm command. If you don't know which connection method to select, contact the system administrator who maintains the host you want to log in to. Ignore any dimmed connection tools in the Method pop-up menu. MacX does not support them.

In Figure 2-5, the connection method chosen is MacTCP®, Apple's implementation of TCP/IP communications protocol for the Macintosh. You'll notice when you select a different connection tool the lower part of the settings box changes. That's because each connection tool has a different method for selecting hosts and zones or domains. The instructions in this section explain the method used with MacTCP. For instructions on using the connection settings box presented by other connection tools, refer to the user's guide for that tool.

■ **Figure 2-5** The Connection Settings box



Choose connection tool from this pop-up menu.

Click host name to select it.

If Host List is empty, enter host name or IP address here.

The lower part of the settings box contains a scrolling field from which you can select a host. The host and domain names that appear in this box come from the Hosts file in your System Folder. Simply click a host name to select it. (It will be highlighted.) Then press the carriage return or click OK to return to the Remote Command window.

If the Host List field is empty, then you either don't have a Hosts file or the file is empty. In this case, type the name or IP address of the host you want in the Host Name or Address field and press the carriage return or click OK to return to the Remote Command window.

- *If you select MacTCPTool:* Before issuing a remote command that specifies MacTCP as the connection method, contact your system administrator or consult your host documentation to verify whether an entry identifying your Macintosh needs to be recorded on the host you want to log in to. In some cases, without this entry, the host will refuse to establish a connection with your Macintosh and will not be able to send you an error message.

4. Upon returning to the Remote Command window, click the Execute button or press the carriage return to execute your command.

If you're not concerned about security breaches, you can turn off the Access Control alert by choosing Access Control in the Remote menu and, thus, removing the check mark next to it.

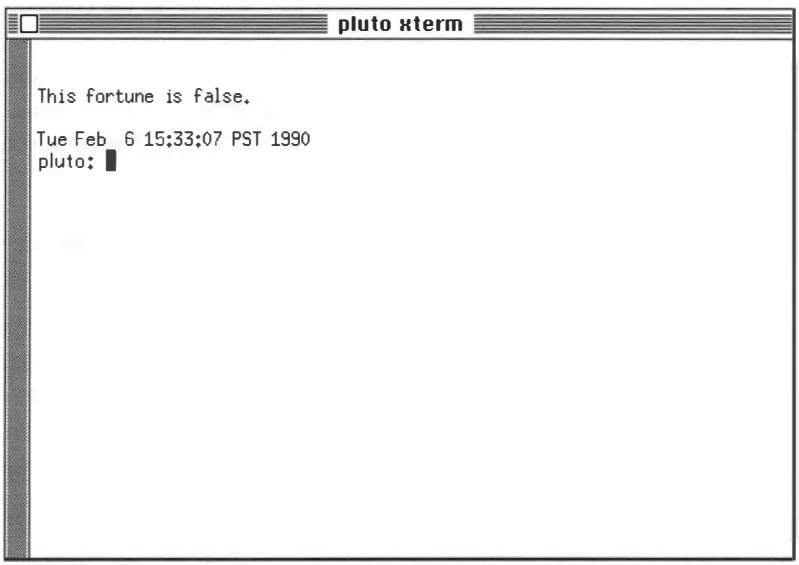
After a brief pause, MacX displays the alert box illustrated in Figure 2-6. This alert appears in response to any attempt to connect with your server, as a precautionary measure. It prevents someone from surreptitiously accessing MacX without your knowledge or approval.

Since this warning has appeared in response to your xterm command, click OK.

■ **Figure 2-6** The Access Control alert box

Within a few moments, an xterm window similar to the one in Figure 2-7 should appear on your screen.

■ **Figure 2-7** Sample xterm window



If nothing appears after a minute or so, you may have made an error typing the command, your username, or your password. For advice, see "Handling Errors" later in this chapter.

For other types of problems, such as network malfunctions, alert boxes will appear on your screen stating the type of error. See Chapter 6, "Troubleshooting," if you need more help.

Where to go from here

You have now successfully transmitted your first remote command to a host computer using MacX. The xterm window displays a system prompt at which you can enter commands just like you would on any terminal logged in to this host. At this point, you have a number of options. You can

- **Start up other clients**

  You can enter commands in your xterm window to start up other clients stored on the host. You'll find starting up clients this way uses less system memory than issuing remote commands. For information on starting up other clients, consult your UNIX host's on-line manual or other publications about the X Window System. See "Other Publications and Resources" in the Preface for suggested publications.

  - *Starting color clients:* Remember to use the –display option and to specify a color screen number (2 or 3) when starting a color client from an xterm window that's on a monochrome monitor. For an example and instructions, see "Specifying Screen Numbers" in Chapter 3.

- **Learn more about windows**

  Notice that the xterm window has a title bar like a Macintosh window, which allows you to move it around. If you want to learn more about manipulating windows in MacX, read "The Basics of Using Windows" in Chapter 4.

- **Learn more about remote commands**

  Read Chapter 3 to learn more about creating and editing remote commands.

- **Quit from MacX**

  When you are ready to exit from MacX, read "Quitting MacX" at the end of this chapter.

Tips for enhancing xterm

Among the numerous options for the xterm command, a few (such as the option that turns on the scroll bar) are especially handy to use. Although you could learn about these options by reading the on-line manual pages for xterm, the tips in this section provide an expedient way for you to set up some of these basic, handy features.

Table 2-2 defines some useful command options. These options appear in the xterm command illustrated in Figure 2-4 and are reproduced below:

```
/usr/bin/X11/xterm -ls -title "®host xterm" -sb -display
"®display"
```

■ **Table 2-2** Handy xterm command options

Option	Description
–ls	Tells your host to execute your shell as a login **shell** while starting xterm. This process activates the commands and other features you expect to have when you use that host.
–title *name*	Specifies a name for the xterm title bar.
–sb	Creates a scroll bar on the left side of the xterm window and enables you to scroll back through the most recent 64 lines of text displayed on your screen.
–sl *number*	Allows you to specify the number of lines you want to scroll back and see. For example, specifying `-sl 100` lets you scroll back and see the last 100 lines displayed on your screen.

You can also use xterm menus to select features, such as the scroll bar. To display these menus hold down the Control key and press the mouse button or the left Arrow key (middle mouse button).

Another thing to check after your xterm window opens is whether or not the Delete key actually deletes characters. If it doesn't, type this simple command at a system prompt:

```
stty erase \^\?
```

Hold down the Shift key and press 6 to get the caret (^).

These tips don't begin to cover all of xterm's features, but, hopefully, they're enough to get you started. Should you want to learn more, one excellent source of information on xterm for nontechnical people is the *X11 User's Guide for A/UX* published by Apple Computer.

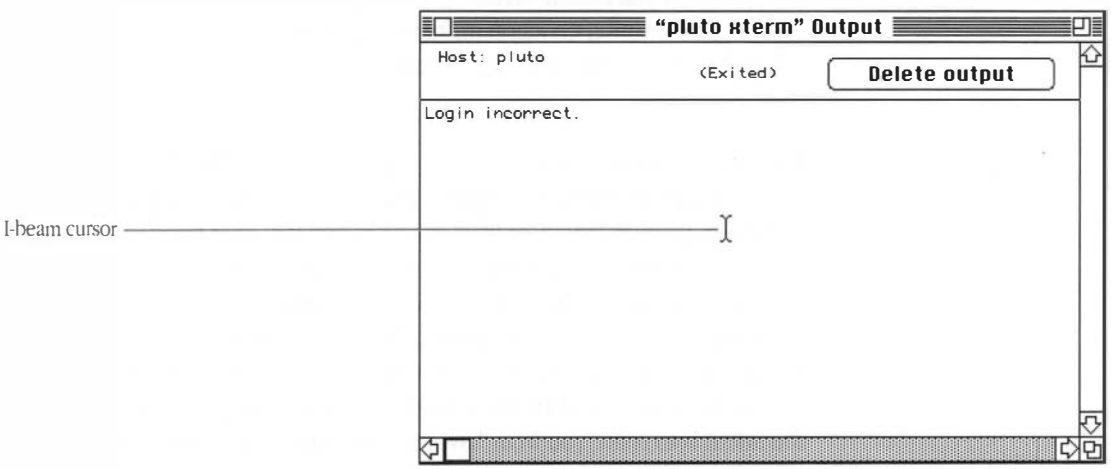
Handling errors

Under certain circumstances an error can occur without MacX notifying you of it. This type of error can happen if you make a mistake typing a remote command, your username, or your password and you have selected the Save or Notify command output option. The Save option saves any information from the host about your command in the command's Output window, but does not notify you of this occurance. The Notify option is more noticeable because it makes your Macintosh beep and display a small, blinking MacX icon in the menu bar when MacX receives command output. In both cases, the intent is to spare you from having to deal immediately with every bit of information generated by the host, regardless of its pertinence to what you are doing. When something goes wrong, however, you have to bring up the Output window yourself to discover what system or error messages the host transmitted. If, instead, you prefer immediate notification, select the Pop Up option from the Output menu. "Creating New Commands" in Chapter 3 explains this option.

No xterm window opens

After issuing the xterm command, if you do not see the xterm window within a couple of minutes, an error most likely occurred. To verify this condition, choose Command Output from the Remote menu and slide the mouse to the right until you see your remote command name. Select this name to display the Output window. You'll see a window, like the one in Figure 2-8.

■ **Figure 2-8** Sample Output window



This window contains an error message indicating that the log-in information in the remote command is incorrect, perhaps because the user name or password is typed incorrectly. Once you determine what your error is, choose the Edit Command from the Remote menu and bring up the Remote Command window for your command. Correct the problem and execute the command again to see if it works.

- *Using the I-beam cursor:* Although you can use the I-beam cursor to type in the Output window, don't assume that you can correct command errors by retyping the command here. Instead, make your corrections where you first created or executed the command, such as in the xterm or the Edit Remote Command window. Since the purpose of the Output window is just to convey information, use the I-beam to type notes and to select text that you want to cut or copy.

Command not found A typical error message displayed in the Output window when the xterm window does not open is "Command not found." Usually this message was sent because you mistyped your command or forgot to precede it with `/usr/bin/X11/`. This string of characters is the pathname to the directory, X11, the traditional location on UNIX machines for all clients used in X Window. Consequently, when issuing commands to start up *any* client, be sure to precede the client name with `/usr/bin/X11`. For example, to start up *xclock,* type `/usr/bin/X11/xclock`. Begin the pathname with a slash (/) and type the entire string as one word, with no spaces in between.

Can't open display This X toolkit error occurs when you have forgotten to enter the –display option in your remote command. Check the command to make sure you typed `-display "®display"` as shown in Figure 2-4.

Client windows suddenly disappear

When square-cornered dialog or alert boxes appear on your screen, MacX stops responding to requests from active clients and waits for your reply. (Figure 2-9 illustrates an alert box.) MacX also waits in this manner while you are working in a large Macintosh application, such as MacDraw®, that requires all the system resources to do something like update the screen. After about 10 minutes, clients that haven't received any responses to their requests will quit running, a condition called *timing out*. However, the client windows will remain on the screen until you release MacX by dismissing the dialog box or until the operating system finishes updating your other application. MacX will then detect which clients have timed out and close their windows without explanation. To prevent this sudden disappearance of clients from your screen, respond to dialog and alert boxes as soon as possible and try not to monopolize system resources on a single task for more than a few minutes.

Other problems

For other types of problems, such as a misspelled host name or a network malfunction, alert boxes describing the error will appear on your screen. Most of these problems should be easy to fix. However, if you need more help, see Chapter 6, "Troubleshooting."

Closing clients

In addition to the three ways MacX gives you to exit from clients, window managers and some clients, themselves, have methods for exiting from or *killing* clients when you're finished with them. Take care to choose a method that's appropriate for the client you want to close, though, so that you don't lose any work you've been doing. This precaution applies especially to terminal emulators, like xterm, and editors, like *xedit*.

Logging off versus killing xterm

You may be tempted to close your terminal emulator window by clicking the close box or by choosing Kill Client from the Window menu. However, Kill Client is not the preferred way to close an xterm or other terminal emulator window because it effects an abrupt disconnection, like disconnecting the telephone. Any unsaved data you had entered in a file would probably be lost unless the host had some way of retaining it. Instead of killing xterm, save any work in progress and log off the host on which xterm is running.

Using Kill Client

Kill Client is designed to disconnect clients, like xclock, that provide you with no way to issue commands to them directly. It is also useful to programmers who want to stop a runaway program. MacX gives you three ways to kill a client. You can

- Click the client window's close box.
- Make the client window active by clicking its title bar and either
  - Press ⌘-W

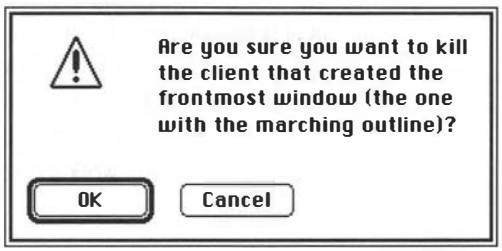
    or
  - Choose Kill Client from the Window menu.

Consult your UNIX documentation or the on-line manual pages for an explanation of the kill command.

If you started a client from your xterm window, you can also terminate it by killing its process ID with the *kill* command. Whatever method you choose, though, make sure to save your work on clients, like xedit, before killing them.

Fortunately, MacX does not immediately execute a Kill Client request. An alert box appears first inquiring whether you really want to kill the client (see Figure 2-9). In addition, the client window displays a moving, dashed outline to make absolutely apparent which client will be killed.

■ **Figure 2-9** The Kill Client warning



Quitting MacX

When you want to exit from MacX, follow these steps:

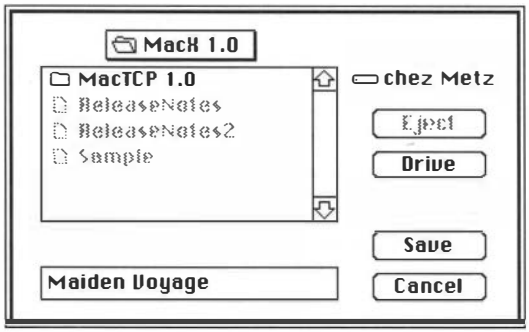
1. **Save your work.**

   Be sure to exit gracefully from any clients that contain information you want to save. Specifically, make sure to log off the host session running in your terminal emulator (xterm or equivalent) so that you don't lose any work you've done.

2. **Choose Save As from the File menu.**

   The Save As dialog box, depicted in Figure 2-10, appears.

■ **Figure 2-10** The Save As dialog box



As discussed at the beginning of this chapter, MacX has created a settings document to preserve your remote command and everything else you have set up, that is, your entire X environment. Unless you save the settings document, MacX will discard everything.

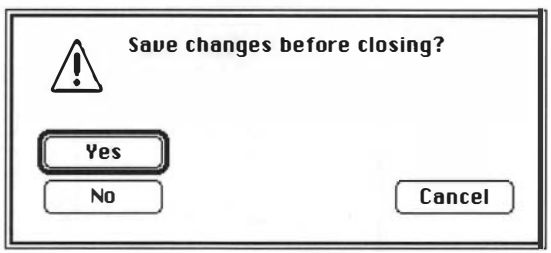
3. **Name your settings document.**

   The document is currently labeled Untitled. Enter another name in the highlighted field and either press the Return key or click the Save button. (Figure 2-10 shows the name "Maiden Voyage" entered in this field.) MacX will create a document icon labeled with the name you have entered. Later, you can click the icon and return to the same environment you just created.

4. **Choose Quit from the File menu or press Command-Q to exit from MacX.**

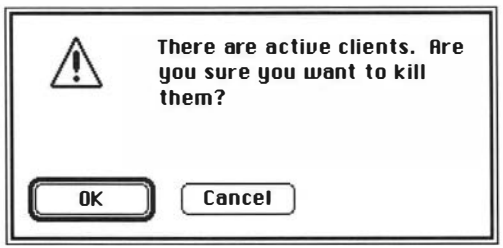
   Should you forget to save your document before quitting, MacX will display the alert box depicted in Figure 2-11. Simply click the Yes button to display the Save As dialog box and continue as before.

■ **Figure 2-11** The Save Changes alert box



If clients are still running when you quit from MacX, the alert box in Figure 2-12 appears. Since clicking OK abruptly disconnects all clients and jeopardizes any unsaved work, don't quit this way if a terminal emulator, such as xterm, is running. Instead, click Cancel and log off as previously described in "Logging Off Versus Killing xterm."

■ **Figure 2-12** The Active Clients alert box



Closing a settings document without quitting MacX

To leave the environment in which you are currently working without quitting MacX, save your document and choose Close "Untitled" from the File menu. The Save As dialog box appears as described above. Follow the same procedure to save and rename your untitled settings document. MacX will create a document icon labeled with the name you have entered, just as it does when you specify Quit. The difference is the MacX application will still be open. The MacX menus will still be there; however, you won't be able to use them unless you open an existing MacX settings document or create a new document by choosing New from the File menu.

CHAPTER 3

Using Remote Commands

CHAPTER 2 INTRODUCED REMOTE COMMANDS BY DESCRIBING HOW TO ACTIVATE a terminal emulator, such as the xterm terminal emulator used with the UNIX operating system. This chapter covers remote commands in depth. It describes all of the command options available as well as how to create new commands, edit and execute commands, and view command output.

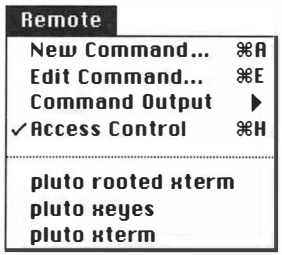
Since MacX creates a settings document each time you invoke it, your entire environment, anything you do—from creating commands to setting window preferences—is stored in that document until you close it. At that point, be sure to save the settings document unless you want to lose everything you've been working on. MacX will prompt you for a name, if the document is new, or else ask you if you want to save your changes.

MacX for A/UX users: If you are running MacX on A/UX, read the *MacX for A/UX Supplement* before reading this chapter. Procedures for opening MacX and using remote commands are different.

The Remote menu

MacX enables you to issue commands to a remote host using the Remote menu illustrated in Figure 3-1. Notice that the menu is divided in two parts by a dotted line. The area above the line contains MacX commands; the area below displays the remote commands that you create.

■ **Figure 3-1** The Remote menu



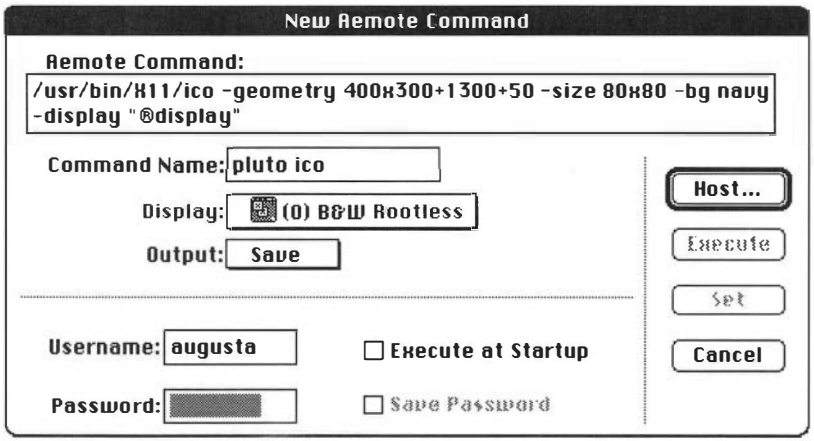
Choosing New Command from the menu causes a dialog box to appear in which you can type a new command and execute or save it. Choosing Edit Command displays a list of existing commands that you can edit, execute, or remove. Names of existing commands also appear at the bottom of the Remote menu so that you can quickly execute a command by choosing its name or quickly edit a command by holding down the Option key while choosing its name. Choosing the Command Output submenu displays a list of commands that are currently executing. To see the output and status of a particular command, choose its name from the list. For more information on using the Edit and Command Output submenus, read "Viewing, Editing, and Executing commands" at the end of this chapter. Next in the menu is the Access Control command. It prevents individuals or programs from connecting to MacX without your knowledge or permission. "Security Features," in this chapter, describes more about this command.

- *Use remote commands sparingly:* Remote commands are a time-consuming and memory-intensive way of starting up clients. For quicker results, start up a client like a terminal emulator or a window manager that allows you to issue commands through it to start other clients.

Creating new commands

To create a new command choose New Command from the Remote menu. The Remote Command window, illustrated in Figure 3-2, appears. This window contains a number of text fields and pop-up menus. If you have not previously edited a command, the fields will be blank. Otherwise, they will contain information from the last command edited.

■ **Figure 3-2** The Remote Command window



Text fields in any MacX dialog box or window hold up to 255 characters. In addition, all text fields scroll up and down and from left to right. To scroll through them, drag with the mouse or use the arrow keys. For a complete list of keys you can type as alternatives to clicking and dragging the mouse, see Table 2-1 in Chapter 2.

The following paragraphs explain the use of each field and menu item.

- **Remote Command**

  This field scrolls so that you can enter a lengthy command or more than one command. Type the command the way you normally would, using the syntax required by the host that will receive the command, or substitute the following macros for the relevant command options, when applicable. By enabling you to avoid specifying a particular type of host or display, these macros make the commands that use them more generic. Be sure to include the registered mark, ®, when using these macros so that the MacX server can distinguish the macro name from regular text.

To type the registered mark, hold down the Option key and press *r*.

  `"®display"`

  Replaced by the display variable (a combination of the display name or network address and the screen identifier) used by the host to identify your Macintosh. For instance, if your network uses TCP/IP protocol, ®display would replace the IP address assigned to your Macintosh plus the screen type selected in the Display pop-up menu: 90.1.3.13:0.0, for example.

  `"®host"`

  Replaced with the host name specified in the Host Name or Address field of the Connection Settings dialog box (see Figure 3-4). Among other things, this macro is useful for displaying the host name in the title bar of a client window so that you remember what host that client is running on.

- **Command Name**

  Assign a command name for the command you've entered so that you can select it later from the Edit Command or Command Output submenus.

Display pop-up menu

Display

This pop-up menu contains a list of four types of screens on which you can display your remote command. An illustration of this list appears at left. The screen number (0 through 3) preceding each screen type represents a style (rooted or rootless) and and a type of screen (color or black and white). In Figure 3-2, the number zero (0) specifies a black and white screen in which the root window does not appear (rootless). MacX will display the client window on the largest screen that fits the screen type selected.

If you specify an IP address in your command instead of the macro ®display, you must append a display station and screen number to the address. *The display station number for a Macintosh is always zero.* See the next section, "Specifying Screen Numbers," for a complete explanation and a table of screen numbers.

Output pop-up menu

MacX icon

Output

This pop-up menu, depicted in the margin, allows you to select four ways to be notified of output from your command once it executes. Unless you select the Discard option, MacX will put this output in the command's Output window as it arrives.

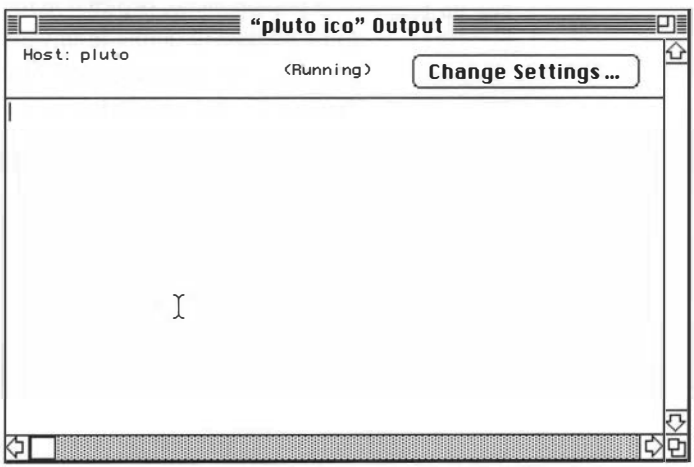
Notify This selection causes a beep and displays a small, blinking MacX icon at the left end of the menu bar. To cancel the notification and see the command output, choose the command name (or names) marked with a diamond from the Command Output menu. A Macintosh window displaying the output appears, as shown in Figure 3-3. Simply click the close box to close the window.

Pop Up Selecting Pop Up causes the Output window to automatically appear on the screen when command output arrives. If you are using MultiFinder and MacX is in the background, the Output window also appears in the background, so if you don't see it pop up, it might be concealed behind a foreground window.

Save If you want to save command output, but not be notified, select this option. When you want to see the output, choose the command name from the Remote menu.

Discard Select Discard if you don't want to be notified of or save command output. This option is useful when the output is a process ID number or some similar system information that does not interest you.

■ **Figure 3-3** The Output window



- **Username**

  Enter the username assigned to you on the host receiving your command.

- **Password and Save Password**

  See "Saving Passwords" in this chapter for instructions on setting this preference.

  Enter the password assigned to you on the host that will receive your command. Each character typed appears as a dimmed, gray box for security purposes. If you have chosen "Save Passwords..." in the Miscellaneous Preferences window, the Save Password check box in the Remote Command window will be activated. Click this box if you want to save your password for this command in the settings document. When you execute the command, MacX will automatically send the password to the host. MacX encrypts all passwords in the settings document, thus ensuring a measure of security suitable for most business purposes. However, be careful not to give away copies of settings documents that contain saved passwords, because the recipients will gain unauthorized access to the corresponding host accounts.

  If you don't click this box, the Password field will always be empty and you will have to enter your password every time you execute the command. In the latter case, MacX will display a password dialog box so that you can enter your password *as long as the Prompt for Usernames and Passwords option is checked in the Miscellaneous Preferences window*. If this option is not checked, MacX sends an empty password to the host, causing an error if the host requires a password.

- **Execute at Startup**

  Check this box if you want your command to begin executing when you open the settings document it is saved in, either by double-clicking the document icon or by choosing Open from the File menu. This option will take effect the next time you open the document. To temporarily override this option, hold down the Option key while opening the settings document and do not release this key until the MacX startup window has disappeared from your screen.

- **Host**

  Click this button to display the Connection Settings box illustrated in Figure 3-4. At the top-left side of the box, a pop-up menu labeled Method contains a list of connection tools that enable you to communicate with hosts that contain clients. Choose a tool that can establish a connection with the host receiving your remote command. If you don't know which connection method to select, contact the system administrator who maintains the host you want to log in to.

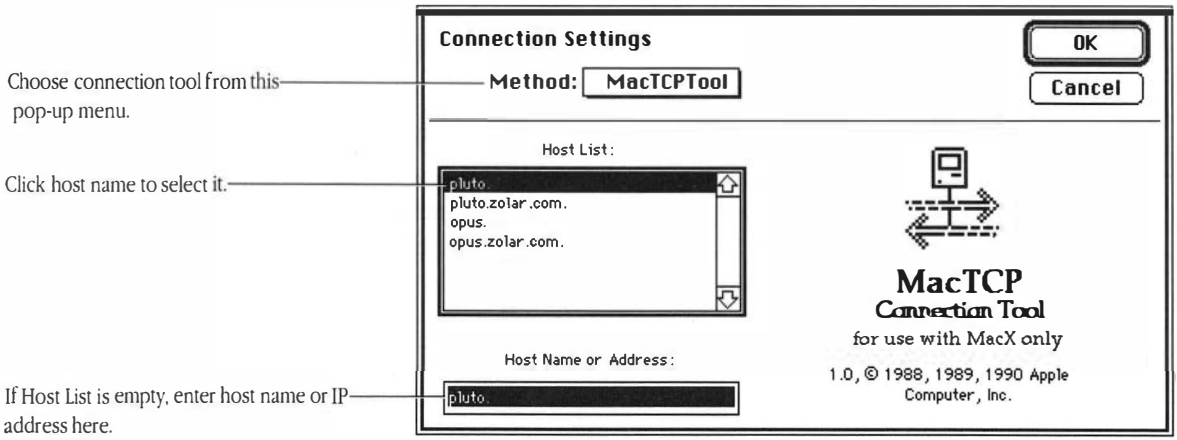
  MacX dims any connection tools in the Method pop-up menu that cannot establish duplex, reliable byte-stream connections. This act does not imply that all the undimmed connection tools will work in every case, though. The host you select in the Host List field must support the protocol for the connection tool you choose.

  ❖ *If you select MacTCPTool:* Before issuing a remote command that specifies MacTCP as the connection method, contact your system administrator or consult your host documentation to verify whether an entry identifying your Macintosh needs to be recorded on the host you want to log in to. (For UNIX hosts, this entry is often recorded in the /etc/hosts file.) In some cases, without this entry, the host will refuse to establish a connection with your Macintosh, yet will not be able to send you an error message.

In Figure 3-4, the connection method chosen is MacTCP, Apple's implementation of TCP/IP communications protocol for the Macintosh. You'll notice when you select a different connection tool the lower part of the configuration box changes. That's because each connection tool has a different method for selecting hosts and zones or domains. The instructions in this section explain the method used with MacTCP. For instructions on using the connection settings box presented by other connection tools, refer to the user's guide for that tool.

The lower part of the settings box contains a scrolling field from which you can select a host. The host and domain names that appear in this field come from the Hosts file in your System Folder. Simply click a host name to select it. (It will be highlighted.) Then press the carriage return or click OK to return to the Remote Command window.

■ **Figure 3-4** The Connection Settings box



If the Host List field is empty, then you either don't have a Hosts file or the file is empty. In this case, type the name or IP address of the host you want in the Host Name or Address field and press the carriage return or click OK to return to the Remote Command window.

- *Why use a Hosts file?* Consider creating a Hosts file if you don't already have one. This file is a convenient place to store the names of the hosts you frequently use. MacX can display these names in the Host List field so that you can select a name rather than enter an IP address. For instructions on creating a Hosts file see Appendix A in the *MacTCP Administrator's Guide*.

- **Action buttons**

  Clicking any of these buttons performs the desired operation and closes the Remote Command window.

Execute	Executes the command and automatically retains it.
Set	Retains the command without executing or saving it.
Cancel	Erases current changes not previously saved.

To save a command, choose Save from the File menu.

When executing commands, MacX may sometimes display an alert box inquiring whether you want to permit a connection by a new X11 client. This alert is controlled by the Access Control command. Read "Security Features" in this chapter for further explanation.

Specifying screen numbers

In MacX, rooted and rootless styles coexist. When issuing a command to display a client, you must specify a screen style (rooted or rootless) and a screen type (color or monochrome) as part of the -display option. MacX provides two ways to specify these selections. As explained earlier, you can use the ®display macro and select a screen type from the Display pop-up menu in the Remote Command window. Alternatively, you can specify a display name in place of the ®display macro and add a display station number and screen number to the server name. This screen number overrides the one displayed in the Display pop-up menu. The server name can take the form of the name stored on the destination host (usually in the /etc/hosts file) or the network address. The complete format of the display option is:

`-display` *server*:*displaystn*.*screen*

where –display is the remote command option, *server* represents the name of your server, *displaystn* represents your display station number, and *screen* represents the screen number. A space separates --display from the server name. A colon (:) separates the server name from the display station number and a period (.) separates the display station number from the screen number.

The display station number for a Macintosh is always zero because MacX can support only one display station. The screen number represents the screen style (rooted or rootless) and type (color or monochrome).

An example of the display option using an IP address is:

```
-display 90.1.0.5:0.0
```

where 90.1.0.5 is the server name (*server*). In the :0.0 extension, the first 0 (zero) is the display station number (*displaystn*) and the second 0 is the screen number (*screen*). As Table 3-1 shows, a screen number of 0 specifies a monochrome screen in which the root window does not appear (rootless).

■ **Table 3-1** Screen numbers

Screen number	Description
0	Monochrome screen, rootless style
1	Monochrome screen, rooted style
2	Color screen, rootless style
3	Color screen, rooted style

Note: Machines that don't have Color QuickDraw™, namely the Macintosh Plus and Macintosh SE, do not support screen numbers 2 and 3.

If your Macintosh has more than one screen of the same type attached to it, MacX displays the client on the largest of those screens. On the other hand, if no screen matches the type specified by the screen number—for example, if you specify "2," but do not have a color screen—then MacX displays the client on the screen displaying the menu bar.

Special uses of the –display option

Some circumstances require using the –display option, specifying particular arguments with the –display option, or both in order to to issue certain commands. In some cases you can't use the ®display macro. In other cases, you simply have to specify a different screen number. This section describes these special cases.

Starting color clients from a monochrome xterm

You must use a display name and a color screen number to start a color client from an xterm window on a monochrome monitor. A typical command to start a rootless, color xclock is:

```
xclock -bg red -fg yellow -hd navy -display karnak:0.2 &
```

Suppose you have issued this command to a host called *pluto*. On pluto, the name for your server is karnak. Since pluto is a UNIX machine, the name "karnak" is listed next to the IP address for your Macintosh in the /etc/hosts file. However, if no name was assigned to your server in the /etc/hosts file, you will have to use your machine's IP address instead. For example, you would substitute a number such as 90.30.0.12 for karnak. To find out what your IP address is, type `printenv` at a system prompt in your xterm window. This command lists your display's environmental variables. Look for a variable called "DISPLAY"; it should look something like this:

```
DISPLAY=90.30.0.12:0.0
```

Notice that the screen number (the last one) is 0, indicating that your xterm client is running in a rootless window on a monchrome screen. Unless you specify the -display option with a color screen number (2 or 3), every client you start from this xterm will appear in a monochrome, rootless window.

Starting clients on another server

To start up a client on another server, you must use the -display option with the display name or network address of that server. Don't use the ®display macro because it supplies the name of *your* server. If you don't know the other server's display name or network address, log in to it, use the *printenv* command to obtain the IP address, and check the /etc/hosts file to find the corresponding display name, if any. This procedure applies to UNIX machines. If the host on which you want to display your client does not run the UNIX operating system or one of its variants, check that host's documentation for instructions.

Clients that expect root windows

Some clients, such as *xfish,* change the appearance of the root window. If you attempt to run such a client in rootless style, MacX ignores root window operations and, often, nothing appears on your screen as a consequence. Should this situation happen to you, try starting up your client in rooted style, using screen numbers 1 or 3.

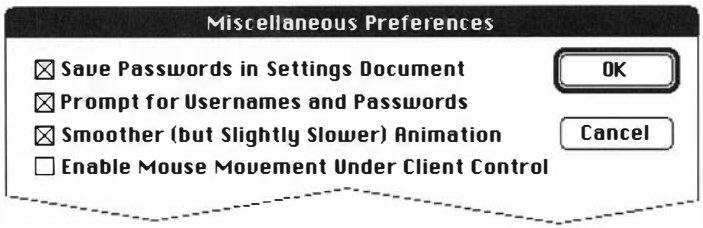
Security features

Through passwords and the Access Control command, MacX protects itself and your accounts on remote hosts from unauthorized or surreptitious intrusion. Learning about these features and taking some simple precautions will, in most cases, afford you more than adequate protection.

Saving passwords

If you would rather not type your passwords every time you issue a remote command, you can have MacX save them in the settings document. To save passwords, choose Miscellaneous Preferences from the Edit menu. When the Miscellaneous Preferences window appears, click the Save Passwords check box and then OK. As a result, the corresponding Save Password check box in every Remote Command window will be highlighted so that you can check it. Should any Remote Command windows be open, MacX will also check their Save Password check boxes.

■ **Figure 3-5** The Miscellaneous Preferences window



MacX encrypts all passwords in the settings document, providing a level of security sufficient for the needs of most businesses. Taking additional precautions to protect secret or sensitive information, however, is a wise practice.

△ **Important** Be careful not to give away copies of settings documents that contain saved passwords because the recipients will gain unauthorized access to the host accounts protected by those passwords. △

Unless you do not need passwords for one or more of the hosts you log in to, you should also click the second check box, Prompt for Usernames and Passwords. This option ensures that MacX will prompt you for a password if you execute a remote command that has a blank password field. Selecting this option is especially important if you do not save passwords in your remote commands because it will circumvent log-in failures on hosts that require you to have a password.

Access Control

Access Control prevents a person or program from connecting to MacX without your knowledge or approval. This security feature is enabled when a check mark appears next to its name in the Remote menu. Simply choose the Access Control command from the menu to remove the check mark and turn this feature off. Pressing Command-H also turns Access Control on or off.

When activated, Access Control displays an alert box any time someone or something attempts to connect to your server, including when you execute remote commands. Figure 3-6 illustrates this alert.

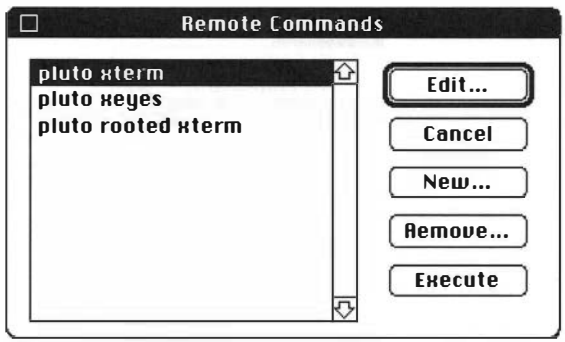
■ **Figure 3-6** The Access Control alert box

If this warning appears just after you have issued a command to start a client, click OK. Otherwise, take whatever precautions you think necessary.

Viewing, editing, and executing commands

The Remote menu provides you three ways to edit and execute existing commands. You can also view the output of commands currently being executed. The quickest way to edit a command is to hold down the Option key while choosing a command name from the bottom of the Remote menu. Another way is to choose Edit Command from the Remote menu and use the window shown in Figure 3-7. Each item listed in the window's scrolling field is the name of a command. To edit a command simply double-click its name in the list. Alternatively, select a command name and click one of the buttons on the right to edit, remove, or execute that command. When you click the Edit button, the dialog window for the selected command appears and you can edit or execute it. Clicking the New button displays either a blank Remote Command window, in which you can enter a command, or a Remote Command window with a copy of the last command edited, which you can modify.

■ **Figure 3-7** The Edit Command window



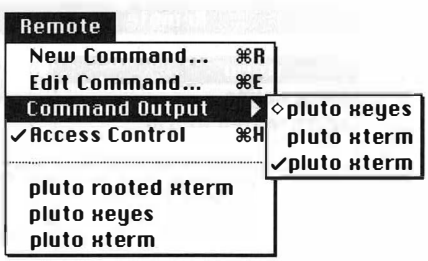
After you have created and executed a command, choosing its name from the Command Output submenu displays the command's Output window, like the one in Figure 3-3. For more information about the Output window, read the following section.

Listed at the end of the Remote menu, below a dashed line, are the names of any existing remote commands. For example, the Remote menu in Figure 3-8 contains three remote commands. Choosing a name from this list automatically executes the command without displaying its Remote Command window.

The Output window

As explained earlier in this chapter, the Output window displays information generated by a command (or about a command) while the command is executing. To get an Output window, select a command name from the Command Output submenu, as shown in Figure 3-8. The submenu lists the command names in the order of command execution, so that the name for the most recently executed command appears at the top of the list.

■ **Figure 3-8** An example of the Command Output submenu



If you have chosen to discard command output (by selecting the Discard option in the Remote Command window) the Output window will appear, but it will be empty. To make the Output window pop up as soon as the

command or the host executing the command generates output, select the Pop Up option. As long as MacX is in the foreground, the Output window pops up in front of any other windows on your screen. MacX is always in the foreground unless you are running MultiFinder and working in a different application.

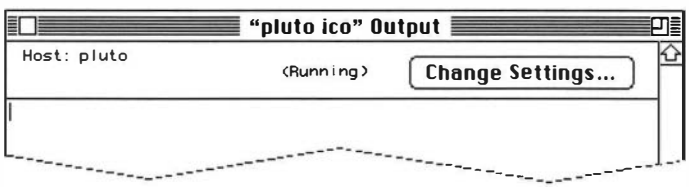
Notice that a symbol precedes some of the command names in the Command Output submenu. These symbols indicate whether an Output window contains new information or information you have already read. The absence of a symbol means the window contains no command output.

Symbol	Means Output window contains
◇	Unread command output
✓	Output you have already read
blank	No command output

Because you can issue the same remote command more than once—for instance, you could issue *pluto xterm* three times and have three xterm windows on your screen—these symbols help you keep track of which command output applies to which client window.

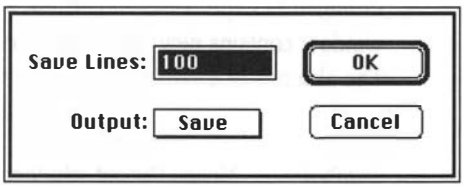
Figure 3-9 illustrates the top portion of the Output window. The name of the host on which the command is running appears on the left, in this case "pluto." A word indicating the status of the command appears in the middle, in parentheses: either Opening, Running, or Exited.

■ **Figure 3-9** Top portion of the Output window



Clicking the Change Settings button on the right enables you to change the output status of your command. A dialog box appears, as shown in Figure 3-10.

■ **Figure 3-10** The Change Settings box



The Output field in this box pops up like the one in the Remote Command window so that you can easily make a selection. In the Save Lines field, you can enter the number of lines of command output you want to save, up to a maximum of 5,000 lines. Remember that the Output window will contain no more than the number of lines you specify here. MacX will discard any additional output by erasing from the beginning so that the most recent command output is retained.

Clicking either of the two buttons on the right will close the Change Settings box. Click OK to save your changes and close. Click Cancel to close the box without saving any changes.

CHAPTER 4

Handling Windows

CHAPTER 1 COVERED THE ESSENTIALS ABOUT WINDOWS, INCLUDING HOW THEY function in rooted and rootless style. This chapter describes how to handle windows in more detail. A section at the end of this chapter focuses on the root window. Also covered are instructions for setting window style and root window preferences.

The basics of using windows

One advantage to running X applications on a Macintosh is that they can run in windows that look and act like Macintosh windows. So, if you're familiar with the Macintosh, you already know a lot about handling windows in MacX. In addition to these familiar features, though, MacX offers other handy features as well, including a selection of five different window styles and the ability to set aside windows by shrinking them down to the size of an icon (called *iconifying*). Consequently, if you already know the basics, read selectively in this section to fill in the gaps; otherwise, read the entire section to learn the essentials required to manipulate windows effectively.

The Window menu

Figure 4-1 illustrates the Window menu. With the exception of the Show B&W and Show Color Root Window commands, all other commands on this menu are dimmed until you have an active client window on your screen. The menu in Figure 4-1 shows one active client, xclock. For instructions on starting up a client, consult Chapter 2, "Getting Started," or Chapter 3, "Using Remote Commands."

■ **Figure 4-1** The Window menu

For your convenience, MacX lists the titles of open windows in alphabetical order at the bottom of the Window menu. The windows listed are client windows and MacX windows, such as the Remote Command window and the Color Namer. The titles appearing in the Window menu are the same ones displayed in the title bar of each open window. In the case of client windows, MacX extracts their titles from certain X protocol requests sent by the clients, not from the Command Name field in the Remote Command window. When visible, the title of a root window will also appear in this list, but none of its client window titles will appear, as they are under the control of a different window manager and are not Macintosh windows.

Since MacX lets you run several versions of the same client simultaneously, it differentiates among these iterations by appending numbers to their window titles. The first time you start up a client, MacX appends no number to its title; however, as soon as a duplicate of that client appears, MacX adds a "(1)" to the original client window's title and a "(2)" to the duplicate window's title. Once issued, these numbers remain even when the original client or one of its clones is terminated. In Figure 4-1, the title "xclock (2)" indicates that a second xclock is running, while the absence of an "xclock (1)" title indicates that the first xclock has ceased operating. The bullet preceding "xclock (2)" means that this client's window is active, or frontmost.

Making windows active

MacX provides a couple of ways to make a window active in rootless style: either by clicking a window's title bar or by selecting its title from the Window menu. In the latter case, a bullet appears in front of the window's title, indicating that the window is now frontmost (or active). An active window in rootless style also has **keyboard input focus** and **colormap focus.** These X terms simply mean that the client in an active window receives characters typed from the keyboard and controls the colors displayed on your screen. If you're not sure whether a window is active, check the Window menu to see if a bullet precedes the window's title.

Since rooted style requires different window managers, the method for making a window active or assigning input focus depends on the window manager you started up. If no window manager is operating, you can usually make a window active and receptive to keyboard input by moving the mouse pointer into it. This technique works for an xterm window.

- *Don't click in windows:* Whether in rooted or rootless style, avoid clicking in a client window to make it active. Taking this precaution prevents the client from receiving the mouse click and performing some action that you don't expect or don't want. Click the title bar or use the Window menu instead.

Positioning windows

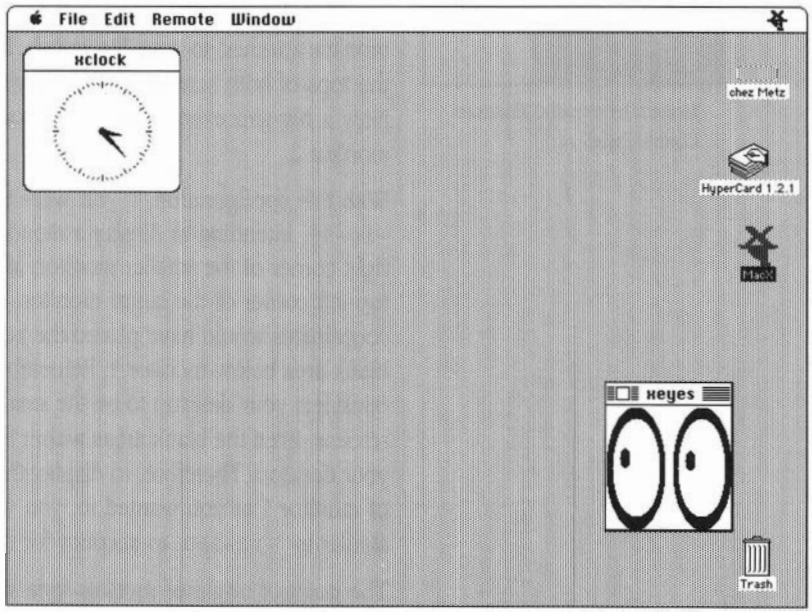
For many clients, the –geometry option determines the position of their top-level windows on the screen. The arguments to this option specify the width and height of the window in pixels followed by x,y coordinates that specify how far in pixels to position the window from the corners of the screen. For example, a –geometry option that specifies

```
-geometry 150x150+10+40
```

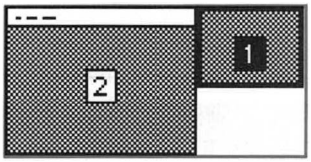
The –geometry option in the xterm command specifies width times height in characters rather than in pixels.

requests a window 150 pixels wide and 150 pixels tall, with the upper-left corner located 10 pixels from the left edge of the screen and 40 pixels from the top. The xclock in Figure 4-2 illustrates this geometry position. If the coordinates were `-10-50`, the lower-right corner of the window would appear 10 pixels from the right side of the screen and 50 pixels from the bottom. The xeyes in Figure 4-2 approximately illustrates this geometry position.

■ **Figure 4-2** Rootless window geometry



The MacX Window Manager will automatically position a window if its coordinates would display it off of the screen, if the command to create it specifies no geometry option, or if the x,y coordinates specified in the geometry option are 0,0. MacX positions such windows in a cascading sequence starting near the screen's point of origin, which X11 specifies as the upper-left corner of the screen. In addition, when no x,y coordinates are specified, MacX automatically positions a window on the largest screen best-suited to display it; for example it displays a color window on the largest color screen available.



Monitor icons in the Monitors Control Panel

When specifying geometry coordinates, be sure to account for the principle that MacX displays a client on the screen that best fits the screen number specified. For example, suppose you have a 19-inch and a 13-inch monochrome monitor attached to your Macintosh. In the Control Panel, you have aligned the tops of both screens so that they're even, as shown in the illustration. Now a big gap exists from the bottom of monitor 1 to the bottom of monitor 2.

With this configuration, if you were to specify geometry coordinates `-10-50`, intending to display a monochrome xeyes client at the bottom right corner of the smaller monitor, MacX would instead display it in the top-left corner of the larger monitor. It does so because your geometry coordinates would have placed the xeyes window off the screen in the blank area below monitor 1. Remember that the MacX Window Manager considers your desktop to be the smallest rectangle that can enclose all your screens. Even the blank areas within this rectangle are considered part of your desktop. Therefore, to display the xeyes client in the lower-right corner of monitor 1 as you wanted to, you would have to specify coordinates similar to `-10-300` to account for the gap between the two monitors.

The point of origin of rootless-style windows is the left corner on the *inside* edge of the border. Therefore, when specifying the y coordinate for a rootless window, account for the height of the title bar. For example, in Figure 4-2, if the y coordinate for the xclock had been +20 instead of +40, the xclock window would have appeared on the screen with its title bar hidden under the Macintosh menu bar.

Styling and manipulating windows

While MacX enables you to treat client windows like regular Macintosh windows, it has no control over windows that open within a client window. According to X protocol convention, these lower-layer windows are controlled by the client, itself, so MacX simply relays the information to your screen. On the other hand, the client and the host it runs on have no control over the type of window MacX creates for the client. They can make requests, called *hints*, but, ultimately, the MacX Window Manager, guided by your preferences, determines what is in the realm of the possible when mapping out windows on the screen.

Window styles

When clients open up on your screen, they appear in Macintosh-style windows. This style has a title bar across the top and a size box in the lower-right corner. If you prefer a different style, follow these simple steps:

1. **Bring the window you want to restyle to the front.**

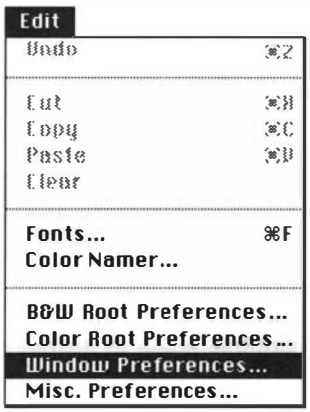
2. **Select Set Window Style from the Window menu to display the five window styles.**

   Figure 4-3 shows these styles. The one currently in effect will have a check mark next to it. Simply highlight another style to change the selection.

■ **Figure 4-3** MacX window styles

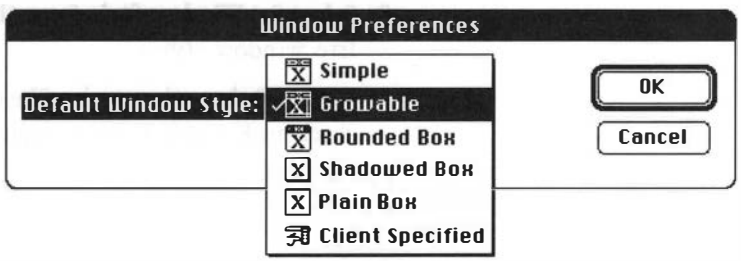
Remember that a window must have a title bar to be moved and a size box to be resized. However, if you select a style without a title bar or a size box you can still move or resize the window by selecting Temporarily Adorn from the Window menu. See "Moving and Sizing Windows" later in this chapter for details.

Setting the default window style



The Edit menu contains a Window Preferences command that sets a default window style for client windows. Figure 4-4 shows the pop-up menu from which you can choose a default style. The default takes effect the next time clients start up. These window styles (magnified in Figure 4-3) apply only to client windows in rootless style, where the MacX Window Manager is in control.

■ **Figure 4-4** Window style preferences



The last item in the menu, Client Specified, allows MacX to convert client border-width hints into one of the five MacX window styles. For example, if you issue a command to start up an xclock with a border width of 4, MacX displays the xclock in a shadow-box style window. If you do not specify a border width, clients usually have a default that MacX converts to a window style. Should this default argument be out of range (less than 1 or larger than 5), MacX substitutes a *2* in its place. See the next section, "Specifying a Window Style in a Command," for information about using the border-width option.

Specifying a window style in a command Usually top-level client windows displayed in a visible root window have a border, the width of which the client specifies when invoked. In rootless style, this border is not displayed; however, you can use the border width option to specify one of the window styles in Figure 4-3, provided the client supports this option and you have selected Client Specified from the Window Preferences pop-up menu. When entering a command to start up a client, specify a window style by including the border-width option in the command.

The procedure for specifying a border width varies depending on the client. For MIT and MIT-compatible clients, the format of the border width option is:

```
-bw n
```

where *n* represents a number from 1 to 5. The style corresponding to each argument is:

Argument	Window style
1	Simple
2	Growable
3	Rounded corner
4	Shadow box
5	Plain box

For example, the following command, typed in an xterm window, starts up the xclock client in a window with rounded corners and a solid title bar.

```
xclock -bw 3
```

In rooted style, where the root window is visible, the border width option functions as originally intended: it specifies the thickness of the window border.

Moving and sizing windows

Before moving or sizing a client window, make sure it is active (in the foreground). Then, depending on the window style selected, you can move and size the window in one of two ways. If the window has a title bar, move it with the mouse pointer as you would any Macintosh window. If the window has a size box, use the mouse to drag the window's dimensions to whatever size you prefer.

If the window has no title bar or size box, Temporarily Adorn is one way to move or size it. First, make sure the window is active. Next choose Temporarily Adorn from the Window menu. Your window will acquire a temporary title bar and size box, giving you one opportunity to move or size it like any Macintosh window. Once you release the mouse button, the window reverts to its former style. If you aren't satisfied with its new location or size, select Temporarily Adorn again.

Cursor for moving windows

The other way to move a window with no title bar employs the Option key. First, move your mouse pointer into the window. Now hold down the Option key, press the button on your mouse, and drag the window anywhere you want. Pressing the Option key changes the mouse pointer to a different shape, shown at left.

- *Stacking windows:* MacX permits you to stack client windows and maintains the contents of windows concealed in the stack so that they reappear as you left them. In other words, you don't have to wait for a client to refresh, or restore, the contents of its window when you uncover it.

MacX also allows you to move and resize a root window displayed on your screen just as you would a client window, because this so-called root window is actually a Macintosh window. For further explanation, see "About Root Windows" near the end of this chapter.

Converting windows to icons

An alternative to moving client windows when they get in the way is to *iconify* them, that is, convert them into icons on your screen. The client can determine what the icon looks like, but if it doesn't specify anything, MacX displays a miniature Macintosh window, pictured in the margin. Iconified windows can be moved around with the mouse pointer, just like any icon on your desktop. You can return them to their original state by double-clicking them or by clicking them and choosing Uniconify from the Window menu.

A window icon

MacX provides two ways to convert a window to an icon:

- Click the window to make it active and choose Iconify from the Window menu.
- Press Command-I.

While the window is iconified, any commands, calculations, or other processes currently operating will continue to execute. When you double-click the icon to restore the window to its previous size and appearance, the client redraws the contents of the window.

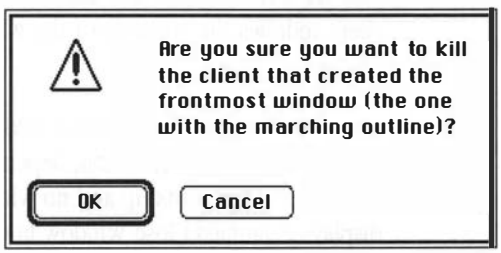
Killing clients and closing windows

Close Window and Kill Client are two commands that appear in the same location in the Window menu, depending on what type of window is active. When MacX first starts up and no windows have opened, the Window menu displays a dimmed Close Window in that spot. When enabled by an active MacX window, such as the Color Namer, choosing Close Window closes the dialog window. It's an alternative to clicking the window's close box or Cancel button.

Kill Client appears only when a rootless client window is active. It is designed to disconnect clients, like xclock, that provide no way to issue commands to them directly. It is also useful to programmers who want to stop a runaway program. However, Kill Client is not the preferred way to close an xterm or other terminal emulator window because it effects an abrupt disconnection, like disconnecting the telephone. Any unsaved data you had entered in a file would probably be lost unless the host had some way of retaining it.

Fortunately, MacX does not immediately execute a Kill Client request. An alert box appears first inquiring whether you really want to kill the client (see Figure 4-5). In addition, the client window displays a moving, dashed outline to make absolutely apparent which client will be killed.

■ **Figure 4-5** The Kill Client warning



❖ *Another way to kill clients:* Clicking the close box in the left corner of a client window's title bar is equivalent to choosing Kill Client.

About root windows

MacX will create a visible root window if one of your clients requests one. In addition, you can show or hide the black and white or color root window at any time by choosing a command from the Window menu. Figure 4-6 shows what your desktop looks like with the root window visible, assuming that your Macintosh has just one screen attached. Usually the root window covers the entire screen when it first appears, although you can set a preference to change its size.

You'll notice that the window in Figure 4-6 has a size box in the lower-right corner. Though labeled *root*, this window is actually a Macintosh window that reveals all or part of an off-screen **bitmap**—a virtual root window stored in memory. This arrangement enables you to manipulate the root-viewing window without disturbing clients running in the underlying virtual root window.

■ **Figure 4-6** MacX on the desktop with the root window visible

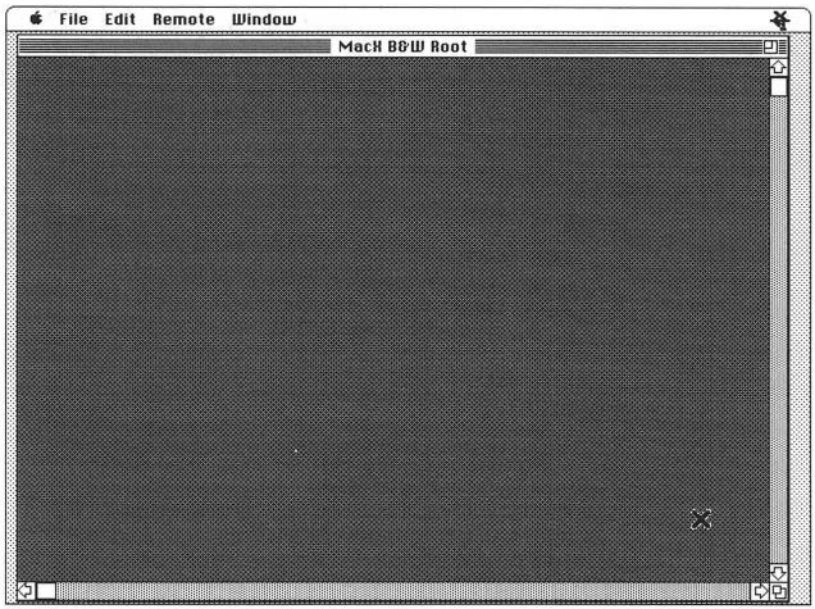
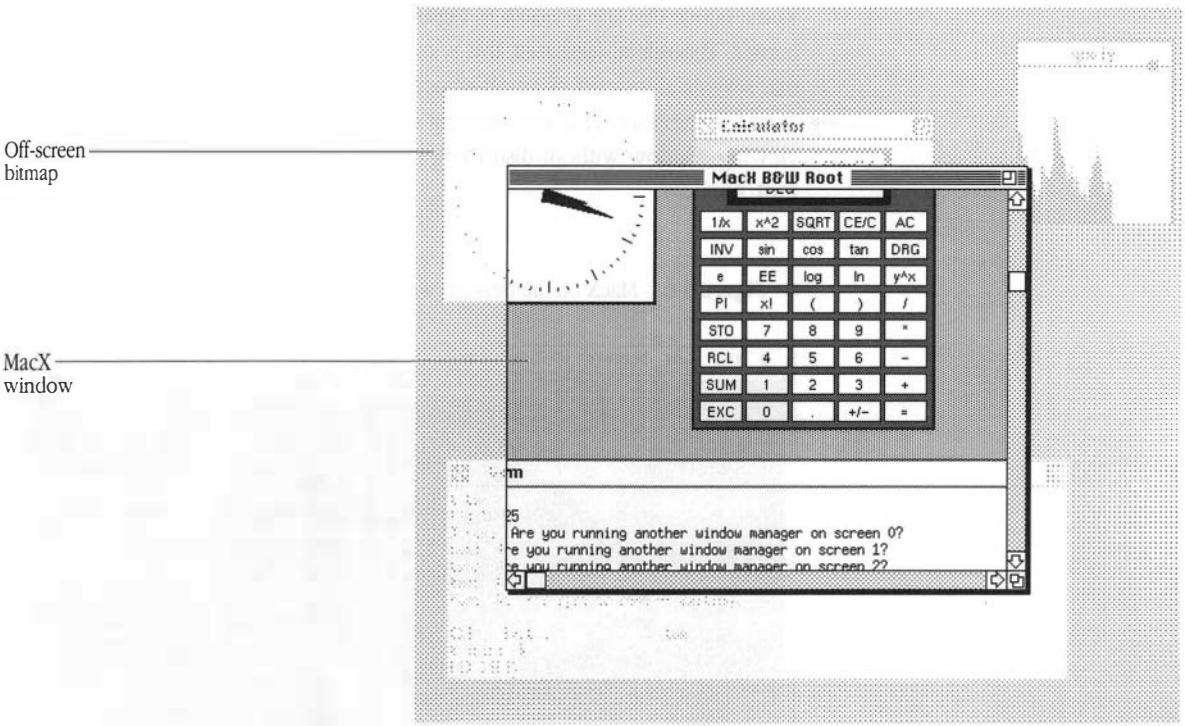


Figure 4-7 illustrates this concept. Notice the root-viewing window is displaying part of a larger off-screen bit map on which an xclock, a calculator, and an xterm client are running. Looking through the B&W root window is analogous to looking through binoculars at a movie screen. In this case, you see parts of the xclock, calculator, and xterm clients. Scrolling reveals other parts.

■ **Figure 4-7** Viewing the off-screen bitmap



Naturally, the root-viewing window can be smaller, but never larger, than the dimensions of the virtual root window. When it is smaller, the scroll bars activate so that you can scroll through the entire root window. Alternatively, you can use the hand cursor for faster scrolling (see note below).

Hand cursor

- *Scrolling quickly with the hand:* To scroll quickly in root windows, change the mouse pointer to a hand by holding down the Option and Command keys. Now press the mouse button and move the mouse to change your view of the root window.

In traditional X implementations, the root window never moves or changes size, so clients are not designed to accommodate such conditions. Accordingly, MacX will not let you resize the virtual root window while any clients—whether rooted or rootless—are running. For more information on resizing the virtual root window, read "Root Window Preferences" at the end of this chapter.

Starting rooted clients

Before starting rooted clients, remember that you'll need to start up a window manager first so that you can move and resize your client windows when they appear on the screen. You can't use the MacX window manager in a root window because it controls rootless windows. Issuing a remote command is one way to start up a window manager. For example, you could start up twm in a black and white root window by typing:

```
setenv DISPLAY "®display";/usr/bin/X11/twm -s &
```

Be sure to choose B&W Rooted (screen number 1) or Color Rooted (screen number 3) from the Display pop-up menu in your Remote Command window before executing the command.

Another way to start up a window manager is to start a terminal emulator, like xterm, and then type the command to start up a window manager in the terminal emulator's window. For example, to start up twm in an xterm window, simply type

```
/usr/bin/X11/twm &
```

at a system prompt. In a moment, the twm banner will appear at the top of the root window and you'll be ready to go.

Since the root window covers up all or part of your desktop, you might find it occasionally gets in the way of something else you want to do. When this happens, you can always hide the root window, even with clients running in it, using a command described in the next section.

- *Trouble starting color clients?* Some color clients, such as xfish and *muncher,* require a lot of memory to run. If MacX tells you that it does not have enough memory to execute a command or warns you that memory is running low, you should allocate MacX a larger block of memory. See "Memory-Related Problems" in Chapter 6 for instructions.

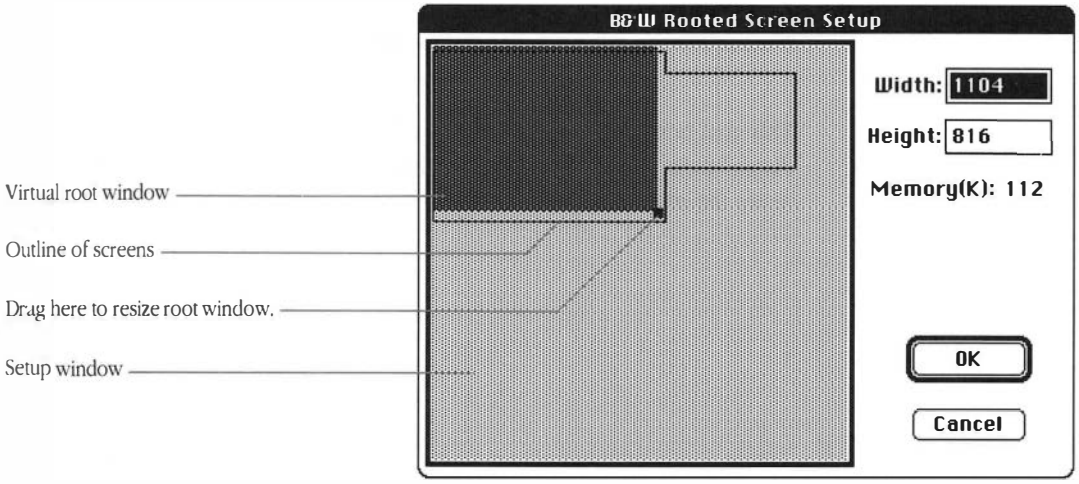
Showing and hiding the root window

MacX does not allow you to iconify a visible root window, but it does give you a way to make it disappear for as long as you like, even when it is displaying client windows. By choosing one of the last two commands in the Window menu, you can hide a black and white or color root window. Once hidden, the command you chose changes from Hide to Show so that you can make the root window reappear. Figure 4-1 illustrates these two commands.

Root window preferences

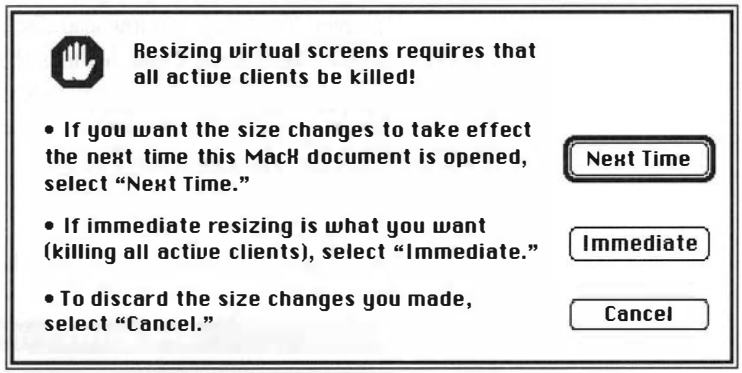
The B&W and Color Root Preferences items, located at the end of the Edit menu, allow you to modify the default settings for a monochrome or color root window, respectively. Selecting either of these items displays a Rooted Screen Setup window. Figure 4-8 shows the monochrome version; the format for the color root window preference is the same. Both preferences let you change the size of the virtual root window stored in memory by dragging the small, black square in the setup window with your mouse pointer. The setup window also displays an outline of the screens attached to your Macintosh. These outlines show you how big a root window you are creating. For example, in Figure 4-8, the setup window shows an outline for one large and one small screen. The virtual root window is slightly smaller than the large screen.

■ **Figure 4-8** The Black and White Rooted Screen Setup window



You shouldn't attempt to resize a root window when any clients—rooted or rootless—are running, but if you do MacX displays an alert box, shown in Figure 4-9.

■ **Figure 4-9** The root window warning message



This alert box gives you the option to defer your changes until later or discard them all together if you would rather not kill your clients in order to resize the root window immediately.

CHAPTER 5

Managing Fonts and Color

CHAPTER 1 INTRODUCED THE FONT DIRECTOR AND THE COLOR NAMER AND briefly described what they do. This chapter covers these features in depth. It also summarizes the X convention for specifying font names. Programmers who want to create fonts should refer to the *X Logical Font Description Conventions, Standard V1.2* in Appendix D for a complete description. For those interested in compiling fonts from Adobe bitmap distribution font files, Appendix C provides the *Bitmap Distribution Format 2.1* standard for your reference.

Although most of the color and font features are easy to use, the procedures for compiling fonts and creating new fonts can get complicated without an understanding of the X font standards and some solid programming experience. If you don't have this background, consult with someone who does before attempting to compile or create fonts.

Using fonts in MacX

The method devised for implementing fonts in MacX enables you to deal with them in a timely, efficient manner. This implementation incorporates three basic components: a MacX Fonts folder, a Font Directory file, and the Font Director. The MacX Fonts folder holds the X font files, the Font Directory file contains a list of these fonts, and the Font Director lets you view and use the fonts.

The MacX Fonts folder

The MacX Fonts folder holds files that contain the standard set of X fonts. These files are organized by font family in subfolders labeled Times, Helvetica, and so forth. Each file contains one variation of a font. For example, one file contains Helvetica 18 point bold, a variation of the Helvetica font family. MacX stores a list of these fonts in the Font Directory file, also located in the MacX Fonts folder. Since the font names in this file must conform to the format specified in the X logical font description standard, they are too long to serve as filenames. Consequently, files in the Font folder have shorter, more obvious names, like Charter 10 bold.

X font file

The primary purpose of the Font Directory file is to supply the Font Director with a current list of X fonts, any Macintosh fonts you have chosen to use in MacX, and any font aliases you have created to act as substitutes for font names or other font aliases. MacX updates this file every time you start up a settings document or whenever you click the Update Font Directory button in the Font Director window. *Turning on the RAM cache in the Control Panel speeds up the updating process significantly.*

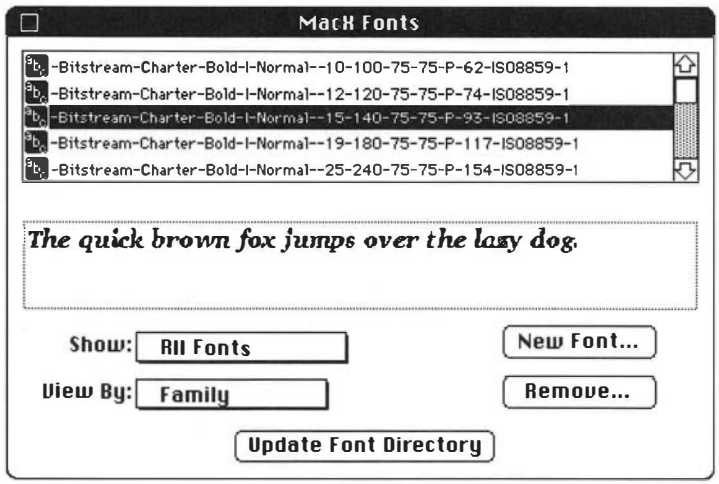
The MacX Fonts folder belongs either in your System Folder or in the same folder as your MacX application. To avoid complications, keep just one MacX Fonts folder on your disk.

The Font Director

The Font Director eliminates many of the tedious and time-consuming aspects of dealing with fonts in a conventional X environment. To see what a font looks like, you do not have to run font clients or type commands with long, cryptic font names in them. Instead, you can display the Font Director window, illustrated in Figure 5-1, click a font name, and see a sample of the typeface. Besides viewing fonts, you can sort fonts in a variety of ways, assign aliases to font names, and compile fonts from bitmap distribution font (BDF) files in a straightforward manner by clicking buttons in the Font Director window.

To use the Font Director, choose Fonts from the Edit menu or press Command-F on the keyboard.

■ **Figure 5-1** The Font Director window



X font alias

Macintosh font

X font icon

The list of fonts in the Font Director window includes all the fonts in the MacX Fonts folder, any Macintosh fonts you have added, and any X font aliases you have created. An icon precedes each item in the list to readily identify whether it is an X font, a Macintosh font, or an X font alias. To expedite the task of finding font names in a long list, the Font Director supports all the standard Macintosh scrolling methods. Besides responding to the scroll bar and the directional arrows, the Font Director scrolls to the font element in the list that matches a letter, word, or number typed at the keyboard. The View By pop-up menu shows the font element that will be searched. For example, if the font element displayed is Family, typing `s` makes the Font Director scroll to the first font family beginning with *s*. Typing `Times` makes the Font Director scroll to the first font in the Times family. Similarly, typing a number produces the first font family starting with that number. However, if none of the Family elements begins with a number, the Font Director scrolls to the top of the list. The Font Director ignores case and blank spaces, but not diacritical marks, when it searches.

If the Font Director cannot open a font file, it displays cross-hatched lines instead. However, if the file contains a reverse-character font—one that prints to the left rather than to the right—the display field is left blank. In this case, the Font Director *can* open the file, but the font tries to display itself off the left edge of the display field where it cannot be seen.

The X format for font names

In Figure 5-1, the long strings of characters in the scrolling field of the Font Director window are font names. Font names comprise a collection of font elements or properties separated from each other by hyphens, a format required by the X logical font description standard. Font names always begin with a hyphen. A pair of hyphens in a font name string indicates that a font element has been omitted, not because it is optional, but because it does not apply to that particular font.

Table 5-1 summarizes the format of the font name registry and suffix specified by the X standard. The table lists the font elements in the order of their appearance in the font name string, starting from the left side. For more information, refer to Appendix D.

■ **Table 5-1** X logical font name format

Element	Definition
Registry	The registration authority that owns the font (its syntax and semantics).
Foundry	The type foundry that digitized and supplied the font shape or metric information.
Family name	The name of a typeface, including all its style variations, such as bold or italic.
Weight name	A name identifying a font's typographic weight, for example bold or medium.
Slant	A one or two-letter code indicating the posture of the font, for example, *R* for Roman, *I* for italic.
Set-width name	One or two words designating the width of a font, for example Normal, Condensed, Double Wide.
Add style name	A term that identifies additional type style information, for example, Serif, Sans Serif. If a typeface doesn't have additional style information, this field is empty.
Pixel size	The number of pixels required to display a particular point size. The pixel size is usually equivalent to or close to the point size.
Point size	A number representing the height of the font in tenths of a point. For example, 8 points would be expressed as 80.
X resolution	A number representing the horizontal resolution of a font, measured in dots per inch (dpi).
Y resolution	A number representing the vertical resolution of a font, measured in dots per inch (dpi).

continued

■ **Table 5-1** X logical font name format *(continued)*

Element	Definition
Spacing	A letter that indicates the escapement class or pitch of a font, for example, M for monospacing, P for proportional spacing.
Average width	A number, measured in tenths of a pixel, that represents the unweighted arithmetic mean width of all characters in a font.
Character set registry	A name or string of alphanumeric characters identifying the organization that owns a typeface character set.
Character set encoding	The character set used to associate each character in a font with its glyph or image.

The format for Macintosh font names

Fortunately, Macintosh font names appearing in the MacX Font Director window have a much briefer format. The format begins with the Macintosh name for the font followed by a hyphen, the point size in decimal, and, optionally, one or more of the following letters:

b	Bold	u	Underline
s	Shadow	e	Extend
i	Italic	c	Compress
o	Outline	m	Monospaced

For example, the Font Director would display the font name for 18 point Geneva bold shadowed italic as Geneva–18bsi.

Adding a Macintosh font to the Font Director list

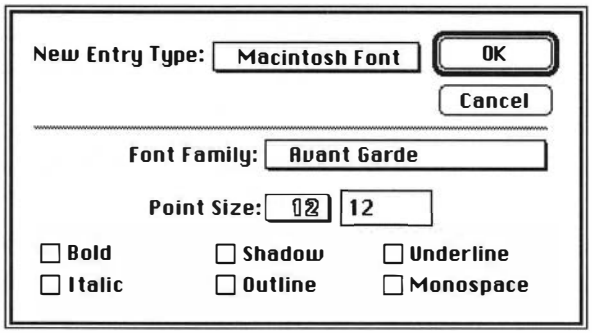
Initially, the Font Director does not list any Macintosh fonts because each font family has so many variations that the font name list would be too long. As an alternative, the Font Director provides a way for you to add the Macintosh fonts you frequently use. To add a Macintosh font to the font name list, follow these steps:

1. **Choose Fonts from the Edit menu to display the Font Director.**

2. **Click the New Font button.**

   When the dialog box illustrated in Figure 5-2 appears, the New Entry Type pop-up menu should display the words "Macintosh Font."

■ **Figure 5-2** Adding a Macintosh font



3. **Choose a font family, point size, and other font properties.**

   Click the Font Family and Point Size pop-up menus to choose a font name and point size. Click the check boxes to select other properties, such as bold or outline.

4. **Click OK to add the font to the font name list.**

   As explained earlier in this chapter, in the font list Macintosh fonts are distinguished from X fonts by a different icon and font-name format. The location of Macintosh fonts in the font list depends on the font element selected from the View By pop-up menu. However, when the font list is set to show all fonts, Macintosh font names usually appear after X font names. For more information about sorting, see "Sorting Fonts" later in this chapter.

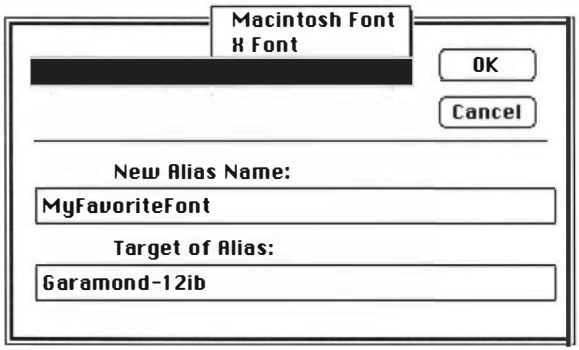
Assigning aliases to font names

If you find the long X font names too cumbersome or puzzling, you can create aliases for them. An alias is simply another name for a font. To create an alias, follow these steps:

1. **Select a font name in the Font Director window and click the New Font button.**

   The dialog box illustrated in Figure 5-3 appears. The font name you selected appears in the Target of Alias field.

■ **Figure 5-3** Entering a font alias



2. **Click the New Entry Type pop-up menu and select Font Alias Name, as shown in Figure 5-3.**

3. **Type any combination of words, letters, or numbers you prefer in the New Alias Name field and click OK.**

   The Font Director inserts the alias, along with any other aliases you create, into of the font name list. Aliases are sorted, along with other font names, according to the font element you have selected from the View By pop-up menu. For more information about this process, see "Sorting Fonts" later in this chapter.

Copying and pasting font names

Using the Copy command in the Edit menu lets you copy a font name to the Clipboard and paste it into remote commands or into client windows. First, select a font name in the Font Director window and then copy it, either by pressing ⌘-C on the keyboard or by choosing Copy from the Edit menu. To paste your selection into a remote command, pull down the Remote menu and choose a command from the Edit Command submenu or choose New Command. When the Remote Command window appears, click the spot in the command string where you want the font name to appear and paste it in, either by pressing ⌘-V or by choosing Paste from the Edit menu. Remember to enclose multiple-word font names in quotation marks before executing your command. Use straight rather than curly quotes.

Once a font name is in the Clipboard you can also paste it into a client command or file by using that client's equivalent paste or insert command. For example, you could paste the font name into a command in the xterm window by pressing the Left Arrow key, the Macintosh equivalent to the middle mouse button on a three-button mouse.

Removing fonts and font aliases

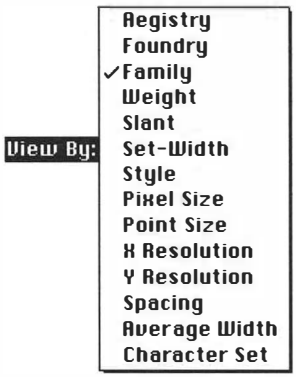
The Remove button removes an X font from the MacX Fonts folder. It also deletes font names and font aliases from the Font Directory file and the font names list in the Font Director window. Simply deleting a font name does not delete the aliases associated with it, however.

To remove a font or alias, select its name in the Font Director window and click the Remove button. A dialog box appears asking whether you really want to remove the font (see Figure 5-4). Click OK to remove the font or Cancel to retain it.

■ **Figure 5-4** Removing a font

❖ *Can't remove cursor and fixed:* Since MacX needs the cursor and fixed fonts to display cursor shapes and text, these fonts can't be removed.

Sorting fonts



View By pop-up menu

The pop-up menu in the margin shows all the ways you can sort font names in the Font Director window. Since the items in the menu are font elements, you can sort font names by any element you desire. The order in which the Font Director sorts font names, however, depends on a number of factors. Here is a list enumerating the sorting order:

1. **First, the Font Director sorts font names by the element chosen.**

   For example, if the sorting element is Weight, all boldface fonts appear first, followed by medium-weight fonts, and so forth.

2. **Next, the Font Director sorts successively, character-by-character, starting with the first element and moving right.**

   Since Registry is optional for fonts that conform to the X logical font description standard, this field is empty and the first element is Foundry. Starting with the Foundry name (such as Adobe) the Font Director sorts font names within the boldface group one character at a time, continuing in this manner through the last element (Character Set).

The way numbers are sorted depends on whether the element chosen to sort by is alphabetic or numeric. If it is alphabetic, numbers appear to sort out of order because each character in multicharacter numbers, like *10* is sorted individually. Since a *1* is smaller than an *8*, the Font Director sorts *10* before *8*. For example, the following font names are sorted by Family, an alphabetic field:

-adobe-times-bold-i-normal--12-120-
-adobe-times-bold-i-normal--14-140-
-adobe-times-bold-i-normal--22-220-
-adobe-times-bold-i-normal--8-80-

Sorting font names numerically makes numbers appear in proper ascending order because a numeric sort treats multicharacter numbers as one character. The same example, shown above, sorted by the numeric field, Pixel Size, appears in the following order:

-adobe-times-bold-i-normal--**8**-80-
-adobe-times-bold-i-normal--**12**-120-
-adobe-times-bold-i-normal--**14**-140-
-adobe-times-bold-i-normal--**22**-220-

The following font name elements are numeric:

- Pixel Size
- Point Size
- X Resolution
- Y Resolution
- Average Width

Compiling fonts from BDF files

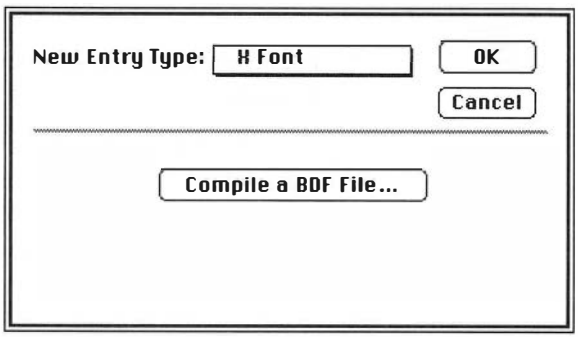
The Font Director enables you to compile BDF files into fonts without exiting from MacX. After compiling a font file, click the the Update Font Directory button to update the Font Directory file and include the new font in the Font Director window. Otherwise, the Font Directory file will automatically be updated the next time you start up MacX.

To compile a font, follow these steps:

1. **Click the New Fonts button on the Font Director window.**

   When the New Entry dialog box appears, select X Font from the New Entry Type pop-up menu. The layout of the box changes, as depicted in Figure 5-5.

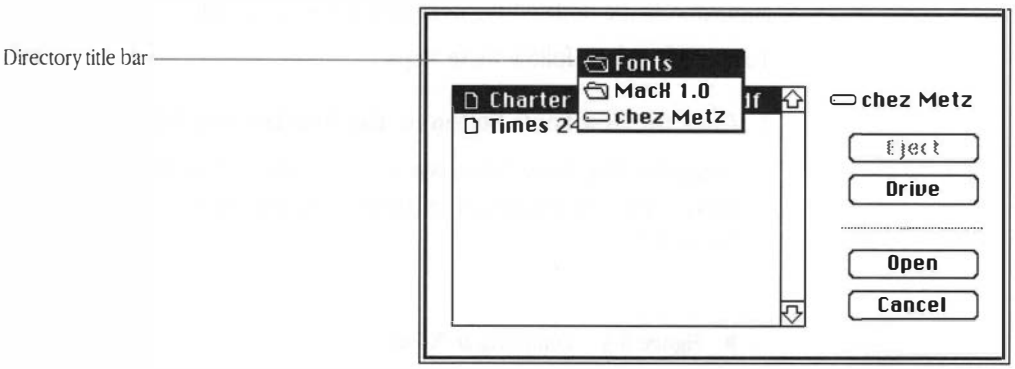
■ **Figure 5-5** Compiling an X font



2. Click the Compile a BDF File button.

A directory dialog box appears.

■ **Figure 5-6** Finding your BDF font file



3. Locate and open the folder containing the BDF files you want to compile.

Check the name of the disk at the top right of the directory box to verify it is the one you want. The Directory title bar above the scrolling window indicates whether the items displayed in the window are at the top level of the disk or in a folder. If the bar shows an open folder icon, clicking the bar reveals the hierarchy of directories (folder names) in descending order from the lowest to highest level. For example, in Figure 5-6, the fonts folder is at the lowest level and Chez Metz is at the top. If, after finding the right directory level, your folder isn't visible in the scrolling window, use the scroll bars or type a letter, word, or number to scroll directly to the folder name.

4. **Select a BDF file name and click Open or double-click the name.**

   A message advising you that the font compiler is compiling your BDF source file appears. When the box disappears, the compiled font file will be in your folder.

■ **Figure 5-7** Compiling your font

5. **Repeat steps 2 through 4 for as many files as you want to compile. When you finish, drag the files into the MacX Fonts folder.**

6. **Click the Update Font Directory button on the Font Director window.**

   MacX updates the Font Directory file. As soon as this file is updated, the newly compiled fonts appear in the Font Directory window.

Using color in MacX

Unlike other X servers, which represent colors in numeric form only, MacX offers a visually oriented approach to specifying colors. Traditionally, the differing amounts of red, green, and blue that constitute colors are expressed as sets of three numeric values. In MacX, these sets of numbers are stored with corresponding color names in the MacX Colors file. MacX converts these values into colors and displays them in a dialog window called the Color Namer. You can also specify a broader range of color values in MacX. Macintosh color values for red, green, and blue range from 0 to 65,535, whereas MIT RGB values range from 0 to 255.

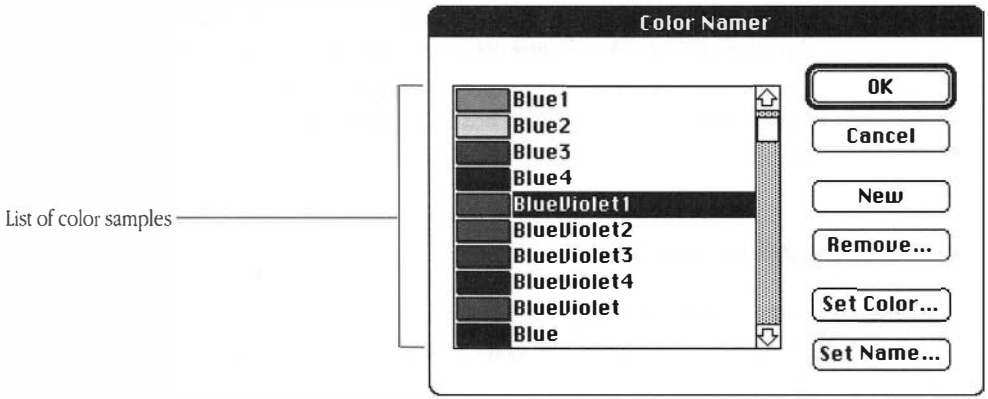
This section explains how to create, modify, and delete individual colors using the Color Namer. For information on displaying clients that use these colors, see "Specifying Screen Numbers" in Chapter 3.

The Color Namer

The Color Namer eliminates the chore of reproducing colors in X. Instead of having to visualize the color you want and experiment with numerical red, green, and blue values until you hit upon the correct combination, you can simply open the Color Namer Window, shown in Figure 5-8, and browse through a long list of color samples until you find the one you want. If the color isn't in the list, you can create a new one or modify an existing color. In addition, you can remove colors, change color names, and copy color names to client commands.

To display the Color Namer window, choose Color Namer from the Edit menu. If the Color Namer command is dimmed, you can't choose it. In this case, your Macintosh does not support Color QuickDraw, a utility the Color Namer requires. Typically, Macintosh Plus and the Macintosh SE models do not have Color QuickDraw so the Color Namer won't work on these machines.

■ **Figure 5-8** The Color Namer window



Sorting, scrolling, saving, and canceling

Sorting color names Any time you make a change that adds, removes, or modifies color names, the Color Namer resorts the color list alphabetically. The sorting method used ignores uppercase, lowercase, and spaces. Any numbers at the end of a color name are sorted numerically, as whole values. For example, the color Gray8 would appear before the color Gray10.

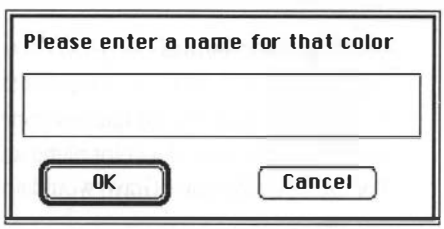
Scrolling To expedite the task of finding colors in a long list, the Color Namer supports all the standard Macintosh scrolling methods. Besides responding to the scroll bar and the directional arrows, the Color Namer scrolls to names in the list that match a letter, word, or number typed at the keyboard. For example, to scroll to colors beginning with *s*, type `s` when the Color Namer window is visible. Type `gray` and the Color Namer scrolls to the first color named *gray* in the list. Similarly, typing `50` produces the first color name starting with that number. The Color Namer ignores case and blank spaces, but not diacritical marks, when it searches.

Saving or canceling changes When you have finished using the Color Namer, click OK to save any changes and close the Color Namer window. Clicking Cancel closes the window without saving any changes. For example, if you've made a color change you don't like, clicking Cancel prevents it from taking effect (along with any other changes as well). Before closing the Color Namer, MacX displays an alert box that asks whether you really want to discard your changes.

Adding new colors

To add a new color, click New in the Color Namer window. First, a dialog box appears requesting that you enter a name for the new color.

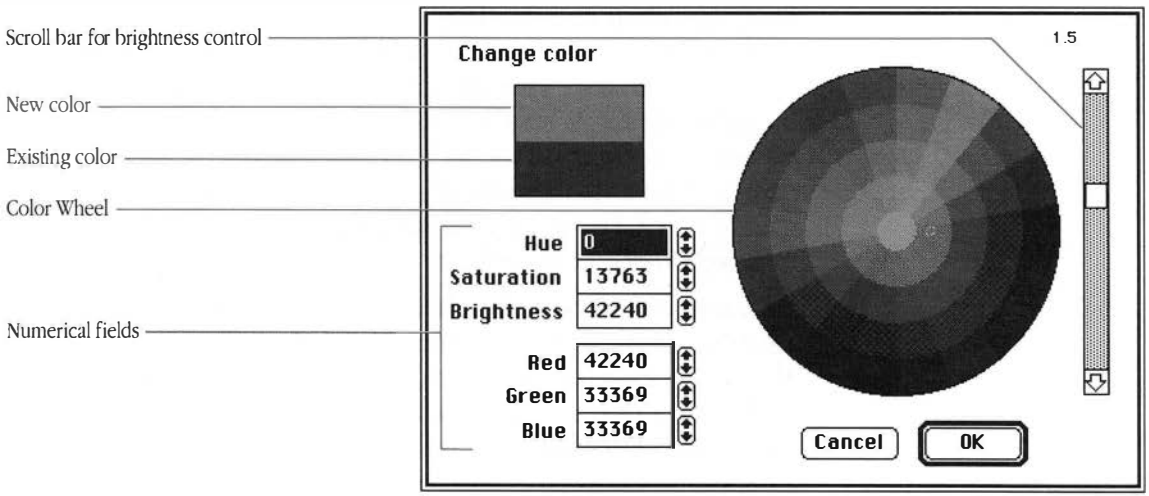
■ **Figure 5-9** Entering a new color name



Next, the Color Wheel dialog box, displayed in Figure 5-10, appears. Clicking anywhere in the color wheel displays the selected color in the top half of the display box directly beneath the legend "Change color." Keep clicking in the color wheel until the box displays the color you want, then click OK. The color will appear in alphabetical order in the Color Namer Window.

If you have RGB values for a particular color, you can enter them in the numerical fields shown in Figure 5-10. Click OK and your color will appear in the Color Namer Window as well.

■ **Figure 5-10** The Color Wheel dialog box



The scroll bar to the right of the color wheel controls color brightness. Dragging the scroll box down makes colors darker; up makes them brighter. For more information on the color wheel read "Using the Color Control Panel," in the *Macintosh System Software User's Guide, Version 6.0.* or see Chapter 8 in the successor to this book, *Macintosh Reference.*

Removing colors

To remove a color, click its color name and then the Remove button (see Figure 5-8). A warning box enquiring "Are you sure you want to remove that color?" appears. Click OK to remove the color or click Cancel to retain it.

Changing colors and color names

Changing a color is similar to adding a new color. Simply double-click the color sample you want to change or click the color sample and then click the Set Color button. The Color Wheel dialog box shown in Figure 5-10 appears. At the top left, the sample color box displays the existing color. As you click the color wheel, the top half of the box reflects the color choices you have made, while the bottom half retains the original color so that you can compare them. When you find the color you want, click OK. The Color Namer window reappears showing the changed color.

If, for some reason, you made a mistake and want the original color back, select Cancel to close the Color Namer and discard any changes you made. Alternatively, if you remember the red, green, and blue or the hue, saturation, and brightness numerical values for the original color, typing either set of these numbers in the appropriate fields in the Color Wheel box restores the color, too.

To change a color name, select the name by double-clicking or using the Set Name button. The Color Namer then displays a dialog box like the one in Figure 5-9 so that you can enter another name.

Copying and pasting color names

Using the Macintosh Copy command enables you to copy a color name to the Clipboard and paste it into remote commands or into client files or documents. First, select a color name in the Color Namer window and then copy it, either by pressing ⌘-C on the keyboard or by choosing Copy from the Edit menu. To paste your selection into a remote command, pull down the Remote menu and select a command from Edit Command submenu or choose New Command. When the Remote Command window appears, click the spot in the command string where you want the color name to appear and paste it in, either by pressing Command-V or by selecting Paste from the Edit menu.

Once a color name is in the Clipboard, you can also paste it into a client command or file by using that client's equivalent paste or insert command. For example, you can paste a color name into a command typed in the xterm window by pressing the Macintosh equivalent to the middle mouse button: the left arrow key. Thus, pasting "seagreen" into the command

```
xclock -bg seagreen
```

starts up an xclock with a sea-green background. Remember to enclose multiple-word color names in quotation marks before executing your command. Use straight rather than curly quotation marks.

CHAPTER 6

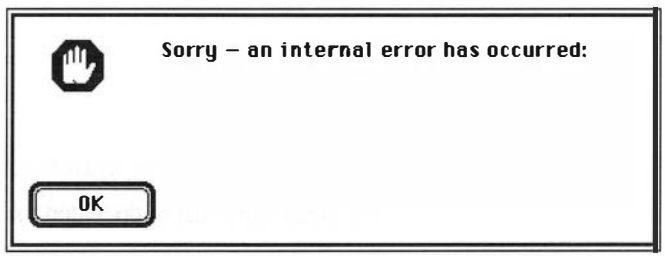
Troubleshooting

Consult this chapter when MacX displays an error message and you aren't sure what to do. Often the problem is easy to fix. In some cases the problem may involve your network or the host you are trying to reach, rather than MacX itself. If the problem does involve MacX and you aren't able to find a solution in this chapter, consult the network or systems administrator at your company, or if you bought MacX yourself, an authorized Apple dealer.

About errors

MacX displays different types of error messages depending on the type and severity of the problem. If the problem involves a malfunction of MacX, the Macintosh hardware, or the operating system, MacX displays an alert box containing the phrase "Sorry—an internal error has occurred." Figure 6-1 illustrates this type of alert box.

■ **Figure 6-1** Sample internal error alert box



Usually, you can do nothing to fix this kind of problem yourself. Any time MacX displays an internal error message, call the network or systems administrator at your company, or if you bought MacX yourself, an authorized Apple dealer.

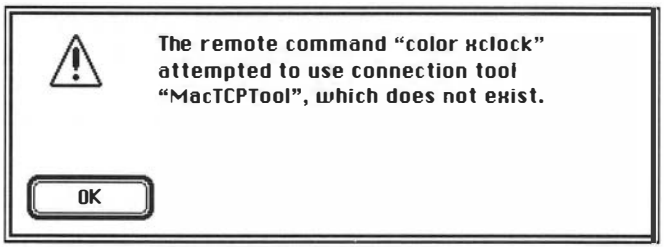
Another set of errors makes MacX quit or prevents MacX from starting up. The phrase identifying these errors is "An unexpected error has occurred." Figure 6-2 illustrates the alert box displayed in this case.

■ **Figure 6-2** Sample alert box for an unexpected error

Frequently, you can recover from these types of errors. When you get an unexpected error, refer to the section in this chapter that covers the type of problem you have encountered.

The last type of error message MacX might display is a warning (see Figure 6-3). Warnings normally advise you of minor problems, such as a field you forgot to fill in, but do not cause MacX to quit.

■ **Figure 6-3** Sample warning



For assistance with warnings, refer to the section in this chapter that covers the type of warning you encounter.

Problems opening MacX

The following messages describe errors that prevent the MacX application or a MacX settings document from starting up. Except where indicated, these errors appear in the unexpected-error alert box depicted in Figure 6-2.

Communications Toolbox not initialized because no connection tools installed

The error "An unexpected error has occurred: the Communications Toolbox could not be initialized because there are no connection tools installed" indicates that the Communications Folder in your System Folder has no connection tools in it. If you forgot to install the MacTCP tool, see the *MacTCP Connection Tool Module* in your *MacX Binder* for instructions. Besides installing the tool, make sure you install and configure the MacTCP driver as well. If you are using a different tool, consult the installation instructions for that tool. If, instead, you inadvertantly dragged your connection tool (or tools) out of the Communications Folder, drag it (or them) back in.

Could not open the default cursor font

A font, called *cursor,* comprises all the cursors used in X. MacX will not open if it cannot open the cursor font. If you see the message "An unexpected error has occurred: could not open the default cursor font 'cursor.' Did you install the MacX Fonts?" follow these steps:

1. **Verify that the *cursor* font file is in the MacX Fonts folder or one of its subfolders.**
   - If it is, drag the file to the trash and replace it with a known good file.
   - If it isn't, drag a copy from the Misc folder in the MacX Fonts folder on your master disk into the MacX Fonts folder.
2. **Verify that you have only one MacX Fonts Folder: either in the System Folder or in the same folder as your MacX application.**
3. **Restart your machine and try opening your MacX application or settings document again.**

Could not open the default font "fixed"

MacX uses a font, called *fixed,* when writing text in client windows, in cases where the clients, themselves, haven't specified a font for this purpose. MacX will not open if it cannot open this default font. If you see the message "An unexpected error has occurred: could not open the default font 'fixed.' Did you install the MacX Fonts?" follow these steps:

1. **Verify that the *fixed* font file is in the MacX Fonts folder or one of its subfolders.**
   - If it is, drag the file to the trash and replace it with a known good file.
   - If it isn't, drag a copy from the Misc folder in the MacX Fonts folder on your master disk into the MacX Fonts folder.
2. **Verify that you have only one MacX Fonts Folder: either in the System Folder or in the same folder as your MacX application.**
3. **Restart your machine and try opening your MacX application or settings document again.**

Error opening or reading a settings document

If your settings document is corrupted, MacX will display the warning, "An error occurred when opening or reading your settings document—a new (Untitled) one will be used instead" in the alert box depicted in Figure 6-3. Chances are the document is unusable. The message advises you that MacX has opened a new settings document since it could not open the one you specified. Be sure to throw this damaged document away.

Initialization of the Communications Toolbox failed

The message "An unexpected error has occurred: Initialization of the Macintosh Communications Toolbox failed (error code *n*)" indicates that, although the Communications Toolbox has been installed, an error occurred when MacX tried to open it. If this error message appears when you start up MacX, verify that the Communications Folder is in your System Folder. If no Communications Folder exists, follow the instructions in Chapter 3 of the *MacX Installation Guide* to reinstall the Communications Toolbox.

MacX requires that the Communications Toolbox be installed

Before MacX can open, the Communications Toolbox must be installed in your System Folder. If the message "An unexpected error has occurred: MacX requires that the Macintosh Communications Toolbox be installed" appears when you start up MacX, verify that the Communications Folder is in your System Folder. If no Communications Folder exists, follow the instructions in Chapter 3 of the *MacX Installation Guide* to install the Communications Toolbox.

No connection tools are working at the moment

The full text of this warning is "None of the connection tools are working at the moment—check that they are properly installed and restart MacX. Until this is done, no incoming X client connections can be accepted." If you see this warning when starting up MacX, follow these steps:

1. **Is the network up?**

   Check with your network administrator to verify that the network is working. If it is, go to step 2.

2. **Does your Communications Toolbox contain connection tools capable of supporting the X11 protocol (such as MacTCPTool)?**

   Some connection tools, such as the serial tool and the modem tool, cannot supply the duplex, reliable byte-stream communications service that MacX needs to set up host connections. If you're not sure which tool or tools to use, check with your network administrator. He or she can help you install the appropriate tool or tools if you don't have them. However, if you already have the right connection tools, go to step 3.

3. Are these connections tools installed correctly?

Consult the installation procedures for your duplex, reliable byte-stream tool(s) to verify they are installed correctly. (Again, your network administrator can help you with this.) If everything is in order, restart your Macintosh and try starting up MacX again. If the same message recurs, call the network or systems administrator at your site, or if you bought MacX yourself, an authorized Apple dealer, for further assistance.

Not enough memory for MacX to run

The error message "There is not enough memory for MacX to run. At LEAST *nn*K more is required" appears when the memory allocated to run MacX is substantially below the suggested memory size in the Get Info dialog box. (The *nn* in the message equals the number of kilobytes of memory required.) See "Memory-Related Problems" at the end of this chapter for instructions on increasing the MacX's memory allocation.

Not enough memory to open "MacX"

The MultiFinder warning "There is not enough memory to open 'MacX'... Do you want to open it using the available memory?" appears when the amount of RAM available to run MacX is less than MacX's application memory size. Usually, you can free more RAM by quitting from another application. Nevertheless, MacX can run in less than its assigned application memory size if this amount is more than adequate to handle the types of activities you perform. If not, MacX may run out of memory while you are performing an operation and display an error message to that effect. For more information, see "Memory-Related Problems" at the end of this chapter.

Problems with remote commands

Messages in this section describe errors that prevent remote commands from executing. Some of these errors involve problems with the host or network specified in the command. Others result from lack of sufficient memory. Unless identified as errors, these messages are warnings that appear in the alert box illustrated in Figure 6-3.

The word *cmdname* appearing in any of the following error messages represents the name of the remote command in question. The word *toolname* represents the name of the connection tool in question.

Auto Execute commands could not be started

The full text of this warning is "One or more 'Auto Execute' commands could not be started." It usually appears after other warnings that better identify the source of the problem. Normally, this error occurs when the host you are trying to reach is down or the connection tool being used is improperly installed. If you're not sure what the problem is, call your system administrator, first, to verify the status of the host. If the host is up, consult the documentation for the connection tool being used to verify that it is properly installed or contact your network administrator for assistance.

Creation of remote command stream failed

When MacX doesn't have enough memory to build a connection to the host, it displays the warning

The remote command *"cmdname"* could not be executed via the tool *"toolname."* Creation of the remote command stream failed.

<*Connection tool error string appears here.*>

As indicated, MacX will also display an error message or warning from the connection tool that should provide more information about the specific problem. For suggested ways to free memory, see "Memory-Related Problems" at the end of this chapter. See also the troubleshooting section of your connection tool documentation for more information about the error or warning the tool generated.

Connection tool does not support ®host or ®display macro in command

The warning "The connection tool *'toolname'* does not support the *'®macro'* used in the command *'cmdname'* " occurs when the connection tool being used to transmit your remote command cannot translate these macros into actual server and host names. (The word ®*macro* represents either ®display or ®host.) Consequently, you should edit your remote command and substitute a server name and screen number for ®display and a host name for ®host. See Chapter 3 for more details.

Not enough memory

MacX displays one of three warnings when it runs out of memory while processing a command. These warnings are:

- **Remote command did not succeed: there was not enough memory to perform the operation.**
- **There is not enough memory to store the remote command in the MacX document.**

  This warning appears when you attempt to save or execute a remote command you have just edited or created and MacX does not have enough memory to store it.

- **There was not enough free memory to create a new remote connection.**

  This warning appears when you execute a remote command and MacX does not have enough memory to set up a connection with the host.

To create more memory, follow the suggestions in "Can't Use a MacX Feature" at the end of this chapter.

Remote command tried to use connection tool which does not exist

If you see the warning "The remote command *'cmdname'* attempted to use connection tool *'toolname,'* which does not exist," the remote command named in the error message specifies a connection tool that you discarded or renamed sometime after you created the remote command. To correct the problem, edit your remote command and select a different connection tool from the Connection Settings dialog box.

Problems with connection tools

The warning in this section also prevents remote commands from executing, except that this messages deals specifically with setting connection tool options in the Connection Settings dialog box. As explained in Chapters 2 and 3, the Connection Settings box appears when you click the Hosts button in a Remote Command window.

Setup options rejected by the connection tool

When MacX warns that "The setup options you specified in the Connection Setup dialog were rejected by the connection tool. The default setup for the tool has been substituted," consult your connection tool documentation to find out which setup options are appropriate. The default setup options will most likely not work.

Problems with fonts

Messages in this section describe errors that prevent MacX from updating the Font Directory file or prevent you from using a font. Figure 6-3 depicts the type of alert box in which these warnings appear. The word *fontname* in any of the following commands represents the name of the font in question. The expression (*n*) represents an error code.

Error getting information about MacX Fonts folder

If the message "An unexpected error has occurred: Error (*n*) getting information about the 'MacX Fonts' folder. Did you forget to install the MacX Fonts?" does not occur for the reason stated it means your fonts folder has been corrupted by a disk error, power surge, or some other rare mishap. To solve the problem, throw away your existing MacX Fonts folder and install a new one from a known good disk. See the *MacX Installation Guide* for instructions.

Error updating Font Directory file

The full text of this warning is "The 'Font Directory' file (in your 'MacX Fonts' folder) needs to be updated, but an error (*n*) occurred when it was being written. MacX can continue, but problems may be encountered when opening new fonts." You will probably not be able to open new fonts until the Font Directory file can be updated. After using the following steps to discover and correct the problem or problems, start up MacX or a settings document to automatically update the Font Directory file.

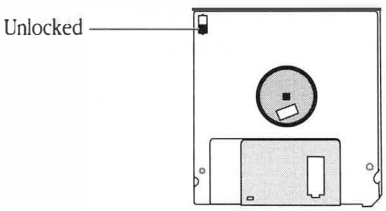
1. Is the Font Directory file locked?

Find the Font Directory file in the MacX Fonts folder. Click the file icon, select Get Info from the File menu, and ensure that the box labeled Locked in the upper-right corner of the Get Info box is not checked.

2. Does MacX have sufficient RAM?

Some operations in MacX, such as displaying color windows, use a lot of memory. On the other hand, doing a lot of things at once can also deplete memory resources allocated to MacX. To ensure that MacX has enough memory to update the Font Directory file, try the suggestions in "Can't Use a MacX Feature" at the end of this chapter. Then try updating the Font Directory file again by starting up MacX or a settings document. If you get the same error message, go to step 3.

3. Is the MacX Fonts Folder on a disk to which you don't have write access?



Back view of disk

Make sure the disk isn't write-protected. If it is a floppy disk, verify that you *cannot* see through the square hole on the top-left side of the disk. Otherwise, slide the black locking tab down so that the hole is blocked. If the MacX Fonts Folder is on a file server, click the file server icon, choose Get Privileges from the File menu, and verify that you have the privilege to make changes. Repeat this procedure to check the MacX Fonts Folder, itself, and any folder in which it is stored.

4. **Is your disk full?**

   Check the upper-right corner of your disk directory folder for the amount of available free disk space.

5. **Is the disk you are using defective?**

   Run the Macintosh utility called Disk First Aid™ or an equivalent to ensure your disk is OK. The *Macintosh Utilities User's Guide* or its successor *Macintosh Reference* contains instructions for using Disk First Aid.

Font Directory file has been damaged. A new file will be created from scratch

This warning states "The 'Font Directory' file has been somehow damaged. A new 'Font Directory' file will be created from scratch to be used instead." Probably a disk error, power surge, or similar accident has corrupted the file. Since the new file will not have the aliases or Macintosh fonts you added, consider restoring an old version of the Font Directory file if you have one.

Font property could not be loaded

If a font file is corrupted, MacX displays the warning "For some reason, the FONT property of the font file *'fontname'* couldn't be loaded. This font will be left out of the Font Directory list and thus will be unavailable for use." When this error occurs, drag the corrupted font file to the trash and get a new copy of the file from your MacX distribution disk or compile a new font file. After dragging the file into the appropriate folder in the MacX Fonts folder, click the Update Font Directory button in the Font Director to insert the font in the Font Directory list. See Chapter 4 for instructions about compiling fonts.

Font file is malformed

The warning "The font file *'fontname'* is malformed and could not be opened" means that the file for the font you selected is missing or corrupted. First, verify that the file exists in the MacX Fonts folder. If it doesn't, drag a copy of the file from your MacX Fonts master disk into your MacX Fonts folder. If the file does exist, drag it to another folder and replace it with a copy from your master disk. Next, click the Update Font Directory button in the Font Director to insert the font in the Font Directory list and try reselecting the font. Once you can select it, be sure to discard any original font file you moved out of the MacX Fonts folder. The file is probably corrupted.

Not enough memory to add fontname to the list of known fonts

Receiving the warning "Unable to allocate enough memory to add the font name *'fontname'* to the list of known fonts," means that MacX has run out of memory to perform this operation. Close any unnecessary windows, clients, or applications to release the memory dedicated to them and try your command again. Also, see "Can't Use a MacX Feature" at the end of this chapter for more suggestions about releasing memory.

One or more fonts could not be matched against the pattern *fontname*

When the warning "Due to a memory allocation failure, one or more font names could not be matched against the pattern *'fontname'* during the servicing of a client request" occurs, MacX has run out of memory to display the font you requested. Close any unnecessary windows, clients, or applications to release the memory dedicated to them and try your command again. Also, see "Can't Use a MacX Feature" at the end of this chapter for more suggestions about releasing memory.

Unable to delete the selected font

MacX displays one of two warnings when it cannot delete a font. These warnings are:

- **Unable to delete the selected font because you do not have sufficient access privileges to delete the font file**

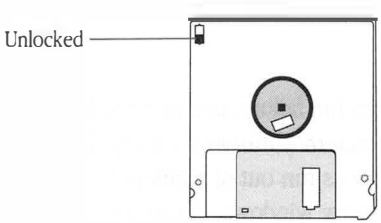
  This warning occurs when the font file or files in question are locked, the MacX Fonts folder is on a disk or in a folder to which you don't have write-access, or the Font Directory file is locked. Follow these steps, in order, until you discover the source of your problem. Then try removing the fonts again.

  1. Is the font file or the Font Directory file locked?

     Find the font file in the MacX Fonts folder. Click the file icon, select Get Info from the File menu, and ensure that the box labeled Locked in the upper-right corner of the Get Info box is not checked. If the font file is not locked, check the Font Directory file using the same procedure.

  2. Is the MacX Fonts folder on a disk or in a folder for which you don't have write-access?

     Make sure the disk isn't write-protected. If it is a floppy disk, verify that you *cannot* see through the square hole on the top-left side of the disk. Otherwise, slide the black locking tab down so that the hole is blocked. If the MacX Fonts folder is on a file server, click the file server icon, choose Get Privileges from the File menu, and verify that you have the privilege to make changes. Repeat this procedure to check the MacX Fonts folder, itself, and any folder in which it is stored.



Back view of disk

- **Unable to delete the selected font, error code *n*.**

  (The *n* represents an error code number.) This warning occurs very rarely. If it appears, record the error code and call Technical Support at Apple Computer. Ask one of the technicians to interpret the error code for you and explain how to solve the problem.

Problems with printing

Printing problems are rare. If you do get a warning stating that an error occurred when attempting to print, you most likely forgot to select a printer name. Instructions in the next section explain how to identify and correct such problems. If the error concerns opening the printer driver, read the subsequent printer driver section for troubleshooting instructions.

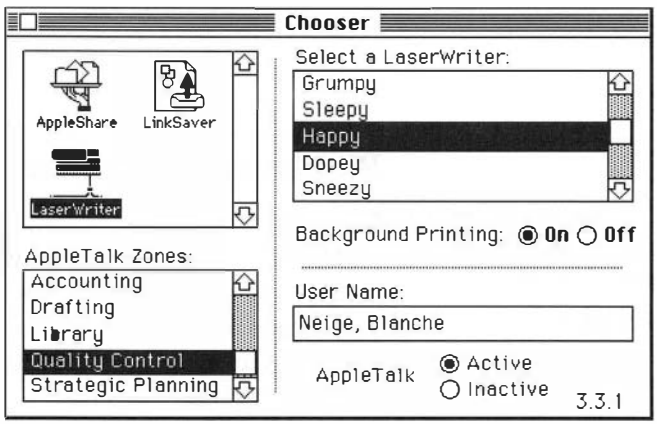
Error occurred when attempting to begin printing or print a page

The two warnings, "An error occurred when attempting to begin printing" and "An error occurred when attempting to print a page," are triggered by the same events. Follow these steps, in order, until you discover the source of your problem. Then try printing again.

1. **Did you select a printer?**

   To check, open the Chooser, click a printer icon, and verify that a printer name is highlighted, as illustrated in Figure 6-4.

■ **Figure 6-4** Checking the printer driver in the Chooser



2. Is the printer driver installed correctly?

Verify that an icon for a printer, such as a LaserWriter® or ImageWriter®, appears in the upper-left scrolling window of the Chooser window. If no printer icon appears, you must install one. See "Using the Chooser with Devices on AppleTalk" in the chapter on desk accessories in the *Macintosh System Software User's Guide, Version 6.0* for instructions.

Error occurred when attempting to open your printer driver

The warning, "An error occurred when attempting to open your printer driver," means that the driver software for your printer is either malfunctioning or not installed. To fix this problem, select the Chooser from your menu.

When the Chooser window appears, verify that an icon for a LaserWriter, ImageWriter, or other printer appears in the upper-left scrolling window, as shown in Figure 6-4. If no printer icon appears, you must install one. See "Using the Chooser with Devices on AppleTalk" in the chapter on desk accessories in the *Macintosh System Software User's Guide, Version 6.0* or see Chapter 8 in the successor to this guide, *Macintosh Reference,* for instructions. If you do see a printer icon, click it and select a printer name from the list that appears on the right. (At least one printer name should appear for you to select.) Now close the Chooser and try choosing Print Clipboard again.

Problems saving a settings document

Error occurred while saving document

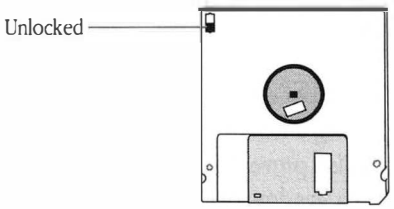
Several types of errors can prevent MacX from saving a settings document. Each of these errors generates the warning "An error occurred while saving your settings document; the document could not be saved." Perform the instructions for each of the following steps until you solve the problem that applies in your case:

1. Does MacX have sufficient RAM?

Some operations in MacX, such as displaying color windows, use a lot of memory. On the other hand, doing a lot of things at once can also deplete memory resources allocated to MacX. Follow the suggestions in "Can't Use a MacX Feature" at the end of this chapter to free memory and then try to save your settings file again.

2. Is the disk you are using defective?

Use the Save As command to save your document to a floppy disk (or another hard disk if you have one). Then, run the Macintosh utility called Disk First Aid or an equivalent to ensure your disk is OK. The *Macintosh Utilities User's Guide* contains instructions for using Disk First Aid.



Back view of disk

3. Is your settings document on a disk to which you don't have write access?

Make sure you have write access to the disk. If it is a floppy disk, verify that you cannot see through the square hole on the top-left side of the disk. Otherwise, slide the black locking tab down so that the hole is blocked. If the settings document is on a file server, click the file server icon, choose Get Privileges from the File menu, and verify that you have the privilege to make changes. Repeat this procedure to check any folder in which the settings document is stored.

4. Is your disk full?

Check the upper-right corner of your disk directory folder for the amount of available disk space. For example, the illustration shows 6,781 K available.

Settings document corrupt or created by very old version of MacX

The warning "The settings document is corrupt or was created by a very old version of MacX. A new (Untitled) one will be used instead" usually means that your document is incompatible with the version of MacX you are using. If you have an older version of MacX try using it to open the document so that you can copy your commands to another file and record the preferences you selected. Be sure to run the older version of MacX on a separate disk. *Do not install it on the same disk as your current version of MacX.* If you still can't open the document or don't have an older version of MacX to try opening it with, you might try retrieving the information in the document with a file recovery program. Otherwise, you will have to reconstruct your document from scratch. Be sure to throw the defective document away.

Memory-related problems

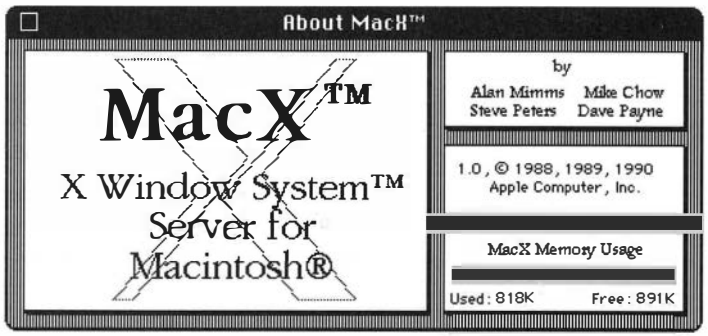
Not having enough memory can prevent you from either (1) starting up MacX or a MacX settings document or (2) using a feature in MacX after the application is open. Memory shortages can also prevent MacX from starting or completing an operation, such as resizing a window or displaying a font. Because it's sometimes difficult to know how much memory certain clients and operations in X require, MacX provides an early warning system to alert you when memory is scarce so that you can avoid major disruptions to your work. This section explains the low-memory warning system and suggests several ways to free memory so that you can avoid memory-related problems in the future.

Early warning system

Caution symbol

MacX attempts to keep 128K of memory in reserve at all times. When it must dip into this memory reserve, MacX notifies you that its memory supply is low by beeping once and alternating a caution symbol with the apple icon (🍎) at the top of the 🍎 menu. If you are running MultiFinder and MacX is not in the foreground, a diamond appears next to the entry for MacX in the list of open applications at the bottom of the 🍎 menu. Choosing About MacX from the menu displays the About MacX screen shown in Figure 6-5. On the lower right side of the screen, a bar graph displays how much memory MacX has left. At minimum, MacX should have no less than 128K of memory free. However, if you plan to run a lot of clients, especially color clients, allowing MacX a few hundred kilobytes of free memory is more realistic.

■ **Figure 6-5** About MacX screen



When MacX displays the caution symbol, you can quickly free more memory by closing any MacX windows or clients you don't currently need. If you release enough memory to replenish the memory reserve, MacX cancels the memory alert and the caution icon disappears. However, if MacX is constantly notifying you of memory shortages, closing windows is not a good, long-term solution for your memory problems. You should probably increase MacX's application memory size, add more RAM to your machine, or both. If the caution symbol appears as soon as MacX begins running, you should definitely quit and increase MacX's application memory size. Read the sections called "Can't Open MacX" and "Can't Use a MacX Feature" for further instructions.

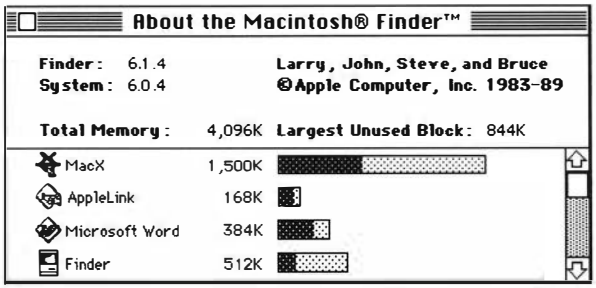
Can't open MacX

Despite claims of insufficiency in error messages, enough memory often exists to open MacX, but problems such as fragmentation or poor allocation make memory temporarily inaccessible. Fragmentation occurs when you open, close, and reopen applications so that they are no longer running in contiguous blocks of memory, but, instead, have segments of unused memory interspersed between them. Since Macintosh applications, including MacX, can run only in contiguous blocks of memory, these segments must be reunited into a block large enough for MacX to use. Poor memory allocation simply means that the application memory size assigned to MacX is too small. These problems are easy to identify and solve by using the few simple techniques enumerated here.

1. **Check your system's total memory to see if it has sufficient RAM.**

   Choose About the Finder™ in the menu and check the Total Memory field to see how much RAM your Macintosh has. (In Figure 6-6, the total memory available is 4 Megabytes.) If your system is running MultiFinder, it should have at least 2 Megabytes to operate smoothly; otherwise, turn off MultiFinder and try launching MacX again.

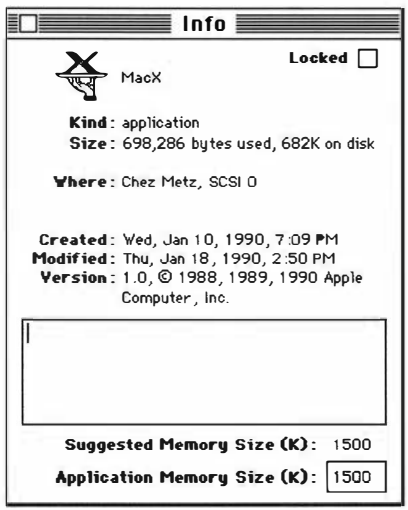
■ **Figure 6-6** About the Finder display



2. Check MacX's application memory size.

If your system has plenty of memory available to support MultiFinder, you still may not have enough memory left over to run MacX. You'll definitely know this case is true if MacX is blinking the caution symbol in the menu bar. To find out, choose About MacX from the menu and see how much free memory remains. If it's less than 128K, exit from MacX if it's open, click the MacX icon, select Get Info from the File menu, and check the field labeled Application Memory Size (K). See Figure 6-7 for an illustration.

■ **Figure 6-7** Get Info display for MacX



If the amount shown is larger than the largest unused block shown in "About the Finder," free some memory, as described in step 3.

If the application memory size is *smaller* than the suggested memory size, MacX is probably being assigned less memory than the minimum required for it to operate. To increase the allocation, click the Application Memory Size box and type in a larger number.

3. Increase the size of the largest unused block.

If the largest unused block in the Finder graph is too small, save, quit, and reopen all the applications currently running. This exercise causes the system to restart the applications in contiguous blocks of memory so that any unused memory occupies one contiguous block as well—the largest unused block—rather than being scattered in unusable fragments.

However, if the largest unused block is still too small to run MacX in, you have a number of alternatives:

- Close other applications running under MultiFinder. In Figure 6-6, AppleLink® and Microsoft Word could be closed to free more memory.
- Remove unnecessary inits (initialization resources), cdevs (Control Panel devices), and fonts from your System Folder. Simply drag the inits and cdevs you don't need out of your System Folder. To remove fonts, use the Font/DA Mover. In both cases, reboot your Macintosh to make these changes take effect.
- Choose the Control Panel from the menu and turn the RAM cache off or reduce the amount of RAM set aside for it.
- Decrease the application memory size for MacX below the size of the largest unused block. MacX requires a minimum of 1024K to operate. Make sure MacX isn't locked or you won't be able to make this change; the box labeled Locked in the upper-right corner of Get Info should not be checked.

As a last resort, if none of these alternatives is a good solution in your case, consider buying more RAM for your machine.

Can't use a MacX feature

Some operations in MacX use a lot of memory. What's more, doing a lot of things at once can also deplete memory resources allocated to MacX. When you see an error message stating that there isn't enough memory to perform an operation, try the following suggestions to free memory:

- Close any color, rooted, or large windows that you don't need to use presently.
- If you are using numerous connection tools to run several sessions, close some of these connections.
- Close clients that are using large **pixmaps.** Some color clients require these.
- Reduce the number of different fonts being used at the same time.
- Lower the number of command output lines in the Saved Lines field of your remote commands.
- Delete the remote command output for commands that have exited. To delete this output, select the Command Output submenu from the Remote menu.
- Close the Remote Command, Font Director, or other MacX windows if they are open.
- Increase MacX's application memory size as described in step 2 of the previous section,"Can't Open MacX."

Critical requirement for memory could not be satisfied

When a memory shortage is particularly grave, MacX has just enough memory to display the message "An unexpected error has occurred: A critical requirement for memory could not be satisfied" and exit. If this message appears when MacX is starting up, increase MacX's application memory size, as described in step 2 of "Can't Open MacX." In most cases, increasing the application memory size is the best solution, especially if you are going to frequently perform the operation that generated this error.

Not enough memory to resize window or show root window

These two warnings, "There is not enough memory to resize the window *'title'* right now" and "There is not enough memory to show the *'title'* root window right now," indicate that MacX has run out of enough memory to perform the stated operation. See the suggestions in "Can't Use a MacX Feature" to release more memory. (The word *title* stands for the title of the window that generated the warning.)

Serious memory allocation failure killed X11 client

MacX displays the warning "A serious memory allocation failure has forced an X11 client application to be killed" when it does not have enough memory to execute a remote command and display a client. If you have other clients and windows open, you can close some of them, as suggested in "Can't Use a MacX Feature" and try your command again. However, if this command was the first you tried to execute (nothing else was running), quit from MacX and increase its application memory size, as described in step 2 of "Can't Open MacX."

APPENDIX A

MacX Versus Other X Servers

THIS APPENDIX STATES THE DIFFERENCES BETWEEN MACX AND conventional X servers. It also describes the miscellaneous preference for slower but smoother animation. Programmers should find this information useful.

Client incompatibilities

MacX handles the requests and clients named in this section differently than conventional X servers.

Backward compatibility

MacX tolerates some types of erroneous GrabPointer and ChangeWindowAttributes protocol requests so that older clients that don't adhere to X11R4 specifications can operate without getting error messages. Some X11R4 servers have an extension that allows people to turn this backward-compatibility concession on or off. However, MacX does not support this extension, so backward compatibility is always in effect.

Cursor size and movement

MacX supports cursors up to 16 by 16 pixels in size. It truncates larger cursors to this size, taking the upper-left 16-by-16 area and forcing the cursor's **hotspot** into it.

Most X11 servers have the capability to constrain movement of the mouse pointer, but this feature is disabled in MacX. Essentially, MacX ignores the ConfineTo field of GrabPointer and GrabButton requests so that it never prevents the cursor from moving in any way. This minor deviation enables people to move their mouse cursors to the menu bar and to windows for other Macintosh applications.

Listing fonts

Certain clients, such as *xlsfonts*, use the XListFonts or XListFontsWithInfo requests to list all fonts currently available. In MacX, these requests will not list Macintosh fonts that are not listed in the Font Directory file, with one exception. If you are using a Macintosh font that isn't recorded in the Font Directory file at the time the client issues one of these requests, that font will also be included in the list. Even though you can't list all Macintosh fonts available, you can still specify any Macintosh font name in client commands for the purpose of displaying text.

Printing images of rootless windows

Clients, like *xwd* (the X Window Dumper), that request an image of the root window of a rootless-style screen using the GetImage request will get a white pixmap. (A GetImage request is like a screen snapshot.) Essentially, this request produces a blank, white square because, in rootless style, MacX cannot capture an image of a root window that isn't visible. However, if the GetImage request specifies an image of client windows in the rootless environment, MacX can comply.

Reproducing uwm animation requests

MacX cannot accurately reproduce some animation requests made by uwm (the Universal Window Manager client), because MacX uses an off-screen bit map and does not update the screen often enough to reproduce some effects. For example, when uwm wants an object to flicker, it issues instructions to draw and erase the object in a single X request. It sends many such requests to the X server, which displays them on the screen. However, MacX draws these requests on an off-screen bit map and does not update the screen often enough to reproduce the flickering there. Consequently, when you try to size a new window, for instance, the outline of the window disappears so that you can't see how big the window is getting while you move the mouse pointer. When you finish resizing, the window reappears, revealing its new size.

Standard colormaps

MacX contains the functionality of *xstdcmap,* the X11R4 client that creates standard colormaps that perform color translation. Accordingly, MacX creates all the standard color maps for screens 2 (color, rootless) and 3 (color, rooted), so you don't have to run this client.

Other differences

This section identifies various features and procedures that are unique to MacX or that MacX handles differently than conventional X servers.

Animation and mouse-movement preferences

MacX has a miscellaneous preference option, Smoother (but Slightly Slower) Animation, that improves animation on a Macintosh screen. This option directs MacX to refresh the screen image of the off-screen bitmap after every X request that causes graphics changes in any window. If the preference is off, MacX updates the screen either when it is waiting to service an X11 request because none are outstanding or when a quarter of a second has elapsed since the last screen update. Checking this preference eliminates delays between screen image updates.

Another miscellaneous preference, Enable Mouse Movement Under Client Control, implements mouse warping when checked. However, clients can warp the mouse only in the part of the window that appears on the screen, not in any areas that are scrolled-off the screen nor that extend beyond the screen boundaries. To find these two preferences, choose Miscellaneous Preferences from the Edit menu.

Color-related differences

If the Macintosh Clipboard contains a PICT object, MacX converts that object to a standard X11 BITMAP selection. However, MacX cannot convert color objects to PIXMAP selections, so these graphics are converted to monochrome bitmaps.

MacX supports six visuals on both color screens. These visuals are 8-bit PseudoColor, 8-bit DirectColor, GrayScale, StaticGray, StaticColor, and TrueColor. In addition, MacX will try to display clients that require more than 8-bit color as well as possible. A visual is a structure that defines a method of using color resources on a screen.

Font path searching

In order to reduce the complexity of handling fonts and maintain a Macintosh-like user interface, MacX does not support the X11 font search path function. It ignores font path search requests from clients. The only impact on the person using fonts in MacX is that all X Fonts must be kept in the MacX Fonts folder or one of its subfolders.

Resetting the server

MacX performs a complete server reset operation whenever the user opens a new settings document or resizes the root window. This operation frees all existing properties on windows, all existing atoms, and all existing storage owned by the server. It is analogous to quitting from MacX and restarting it.

Screen definitions

Normally, a video display device can display one X screen: either a monochrome root window or a color root window. On video display devices capable of displaying both types of screens, the screens are mutually exclusive. Moreover, each video display device is assigned a unique screen number. In contrast, a Macintosh running MacX supports four screens (0 through 3) on one or more display devices. These screen numbers represent different root and window styles (rooted, rootless, monochrome, color) rather than individual video display devices, because the Macintosh views all display devices attached to it as one big desktop.

Typing international characters

MacX conforms to the standard Macintosh method for typing international characters, that is, holding down the Option key and pressing another character to obtain diacritical marks, such as grave accent (`), circumflex (^), and tilde (~). For example, on the U.S. keyboard, pressing the Option key and the *i* key together and then typing another letter produces a circumflex over that character. Use the Key Caps desk accessory to see what keys to press on your keyboard to obtain these marks. For more information, consult your *Macintosh System Software User's Guide* or its successor, *Macintosh Reference*.

Appendix E contains a matrix indicating the hexadecimal equivalents for **ISO Latin 1** and **Macintosh extended ASCII** characters. These characters do not always have the same character-code value when translated from one character set to the other.

APPENDIX B

The MacX Window Manager

THE MACX WINDOW MANAGER CONTROLS THE APPEARANCE AND manipulation of windows in rootless mode, extending the Macintosh look and feel to the conventional X environment. This appendix explains the more technical aspects of some of its features.

Transferring text and graphics between clients

In conformance with the ICCCM, the MacX Window Manager enables you to cut, copy, and paste text and graphic objects in PICT format from one client to another. To transfer text or graphics, use the client's method to perform the cut, copy, or paste operation. MacX is advised of a cut or copy event and places the selected item in the clipboard. When you indicate where you want to paste the item, by issuing the appropriate command within the client you want to receive it, MacX transfers the contents of the clipboard to that client. For example, to paste your selection in an xterm window, you would press the left arrow key, the equivalent in MacX to pressing the middle mouse button.

As discussed in Chapter 1, the MacX Window Manager also enables you to cut, copy, and paste text and graphic objects in PICT format from Macintosh applications to X11 clients. If the Macintosh Clipboard contains a PICT object, MacX converts that object to a standard X11 BITMAP selection. Note that MacX does not support conversion of color PICTs to PIXMAP or COLORMAP selections. Color clipboard objects are converted to monochrome BITMAP selections.

MacX can also copy and paste text, such as font or color names, from MacX dialog windows (the Remote Command window, Font Director window, and Color Namer) to clients using substantially the same operation. See "Cut, Copy, Paste, and other Editing Commands" in Chapter 1 for a summary of these procedures.

Window positioning

The MacX Window Manager will automatically position a window if its coordinates would display it completely off of any screen, if the command to create it specifies no geometry option, or if the x,y coordinates specified in the geometry option are 0,0. MacX positions such windows in a cascading sequence starting from the upper-left corner of the screen that is most appropriate to display the window. Color windows are automatically positioned in a sequence on color screens and monochrome windows are automatically positioned onto monochrome screens. If no appropriate screen exists for a window, it is automatically positioned on the screen containing the menu bar.

The point of origin of rootless-style windows is the left corner on the *inside* edge of the border, so you should account for the height of title bars when specifying the *y* coordinate.

A P P E N D I X C

Bitmap Distribution Format 2.1

THIS APPENDIX CONTAINS A COPY OF THE DOCUMENT, *BITMAP DISTRIBUTION Format 2.1*, written by Adobe Systems, Inc. It is provided as a reference to programmers who want to compile fonts from BDF files.

Bitmap Distribution Format 2.1

Table of Contents

List of Figures

1. Introduction

This document describes Bitmap Distribution Format (BDF), version 2.1. BDF is an X Consortium standard for font interchange, intended to be easily understood by both humans and computers.

2. File Format

Character bitmap information will be distributed in an USASCII encoded, human readable form. Each file is encoded in the printable characters (octal 40 through 176) of USASCII plus carriage return and linefeed. Each file consists of a sequence of variable-length lines. Each line is terminated by a carriage-return (octal 015) and line-feed (octal 012), or by just a line-feed.

The information about a particular family and face at one size and orientation will be contained in one file. The file begins with information pertaining to the face as a whole, followed by the information and bitmaps for the individual characters.

A font bitmap description file has the following general form, where each item is contained on a separate line of text in the file. Items on a line are separated by spaces.

1. The word STARTFONT followed by a version number indicating the exact file format used. The version described here is number **2.1**.
2. One or more lines beginning with the word COMMENT. These lines may be ignored by any program reading the file.
3. The word FONT followed by either the XLFD font name (as specified in the X Logical Font Description Conventions standard) or some private font name. Creators of private font name syntaxes are encouraged to register unique font name prefixes with the X Consortium to prevent naming conflicts. Note that the name continues all the way to the end of the line, and may contain spaces.
4. The word SIZE followed by the **point size** of the characters, the **x resolution**, and the **y resolution** of the device for which these characters were intended.
5. The word FONTBOUNDINGBOX followed by the **width in x**, **height in y**, and the x and y displacement of the lower left corner from the **origin**. (See the examples in section 3).
6. Optionally the word STARTPROPERTIES followed by the number of properties (**p**) that follow.
7. Then come **p** lines consisting of a word for the **property name** followed by either an integer or string surrounded by double-quote (octal 042). Internal double-quotes characters are indicated by using two in a row.
8. Properties named FONT_ASCENT, FONT_DESCENT, and DEFAULT_CHAR should be provided to define the logical font-ascent and font-descent and the default-char for the font. These properties will be removed from the actual font properties in the binary form produced by a compiler. If these properties are not provided, a compiler may reject the font, or may compute (arbitrary) values for these properties.
9. The property section, if it exists, is terminated by ENDPROPERTIES.

10. The word CHARS followed by the number of character segments (**c**) that follow.

11. Then come **c** character segments of the form:

    a. The word STARTCHAR followed by up to 14 characters (no blanks) of descriptive **name** of the glyph.

    b. The word ENCODING followed by one of the following forms:

        i. <n> - the glyph index, i.e. a positive integer representing the character code used to access the glyph in X requests, as defined by the encoded character set given by the CHARSET_REGISTRY-CHARSET_ENCODING font properties for XLFD conforming fonts. If these XLFD font properties are not defined, then the encoding scheme is font-dependent;

        ii. -1 <n> - equivalent to form above. This syntax is provided for backwards compatibility with previous versions of this specification, and is not recommended for use with new fonts;

        iii. -1 - an unencoded glyph. Some font compilers may discard unencoded glyphs, but in general the glyph names may be used by font compilers and X servers to implement dynamic mapping of glyph repertoires to character encodings as seen through the X protocol.

    c. The word SWIDTH followed by the **scalable width** in x and y of character. Scalable widths are in units of 1/1000th of the size of the character. If the size of the character is *p* points, the width information must be scaled by *p*/1000 to get the width of the character in printer's points. This width information should be considered as a vector indication the position of the next character's origin relative to the origin of this character. To convert the scalable width to the width in device pixels, multiply SWIDTH times *p*/1000 times *r*/72 where *r* is the device resolution in pixels per inch. The result is a real number giving the ideal print width in device pixels. The actual device width must of course be an integral number of device pixels and is given in the next entry. The SWIDTH y value should always be zero for a standard X font.

    d. The word DWIDTH followed by the width in x and y of the character in device units. Like the SWIDTH, this width information is a vector indicating the position of the next character's origin relative to the origin of this character. Note, that the DWIDTH of a given "hand-tuned" WYSIWYG glyph may deviate slightly from its ideal device-independent width given by SWIDTH in order to improve its typographic characteristics on a display. The DWIDTH y value should always be zero for a standard X font.

    e. The word BBX followed by the **width** in x (*BBw*), **height** in y (*BBh*) and x and y displacement (*BBox*, *BBoy*) of the lower left corner from the **origin** of the character.

    f. The optional word ATTRIBUTES followed by the attributes as 4 **hex-encoded** characters. The interpretation of these attributes is undefined in this document.

    g. The word BITMAP.

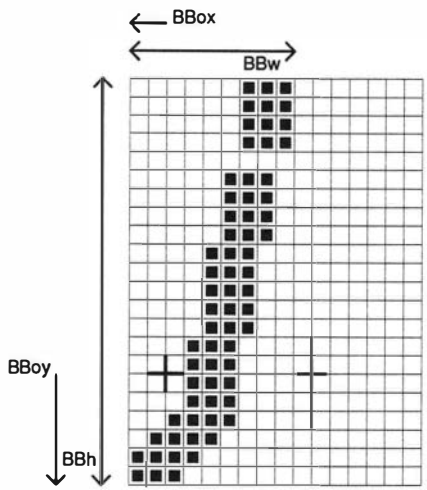
h. *h* lines of **hex-encoded bitmap**, padded on the right with zero's to the nearest byte (i.e., multiple of 8).

i. The word ENDCHAR.

12. The file is terminated with the word ENDFONT.

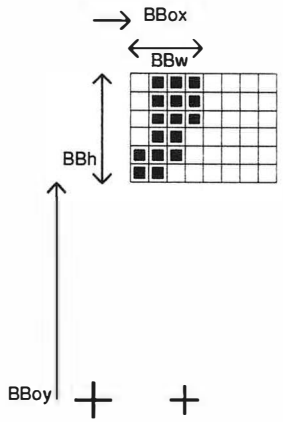
3. Metric Information

Figures 3-1 and 3-2 best illustrate the bitmap format and character metric information.



BBw = 9, BBh = 22, BBox = -2, BBoy = -6
DWIDTH = 8 0
SWIDTH = 355 0
"+" = character origin and width

Figure 3-1: An example of a descender



BBh = 6, BBw = 4, BBox = +2, BBoy = +12
DWIDTH = 5 0
SWIDTH = 223 0

Figure 3-2: An example with the origin outside the bounding box

4. An Example File

The following is an abbreviated example of a bitmap file containing the specification of two characters (the j and quoteright in 3).[1]

```
STARTFONT 2.1
COMMENT This is a sample font in 2.1 format.
FONT -Adobe-Helvetica-Bold-R-Normal--24-240-75-75-P-65-ISO8859-1
SIZE 24 75 75
FONTBOUNDINGBOX 9 24 -2 -6
STARTPROPERTIES 19
FOUNDRY "Adobe"
FAMILY "Helvetica"
WEIGHT_NAME "Bold"
SLANT "R"
SETWIDTH_NAME "Normal"
ADD_STYLE_NAME ""
PIXEL_SIZE 24
POINT_SIZE 240
RESOLUTION_X 75
RESOLUTION_Y 75
SPACING "P"
AVERAGE_WIDTH 65
CHARSET_REGISTRY "ISO8859"
CHARSET_ENCODING "1"
MIN_SPACE 4
FONT_ASCENT 21
FONT_DESCENT 7
COPYRIGHT "Copyright (c) 1987 Adobe Systems, Inc."
NOTICE "Helvetica is a registered trademark of Linotype Inc.
ENDPROPERTIES
CHARS 2
STARTCHAR j
ENCODING 106
SWIDTH 355 0
DWIDTH 8 0
BBX 9 22 -2 -6
BITMAP
0380
0380
0380
0380
0000
0700
0700
0700
0700
0E00
0E00
0E00
0E00
0E00
1C00
1C00
1C00
1C00
2C00
7800
F000
```

[1]Helvetica® is a registered trademark of Allied Corporation.

```
E000
ENDCHAR
STARTCHAR quoteright
ENCODING 39
SWIDTH 223 0
DWIDTH 5 0
BBX 4 5 2 12
ATTRIBUTES 01C0
BITMAP
70
70
60
E0
C0
ENDCHAR
ENDFONT
```

X Logical Font Description Conventions

THIS APPENDIX CONTAINS A COPY OF THE DOCUMENT, *X LOGICAL FONT Description Conventions*, written by Jim Flowers at Digital Equipment Corporation. It is provided as a convenience for those who need to refer to the X font name standard. Table 5-1 in Chapter 5 provides a summary of the X font name format sufficient for interpreting the font names in the MacX Font Director window.

X Logical Font Description Conventions

Version 1.3

MIT X Consortium Standard

Table of Contents

1. Introduction

It is a requirement that X client applications must be portable across server implementations, with very different file systems, naming conventions, and font libraries. However, font access requests, as defined by the X Window System Protocol, Version 11, do not specify server-independent conventions for font names, nor provide adequate font properties for logically describing typographic fonts.

X clients must be able to dynamically determine fonts available on any given server, in what sizes, styles, etc., so that understandable information can be presented to the user, or that intelligent font fallbacks can be chosen. It is desirable that the most common queries could be accomplished without the overhead of opening each font and inspection of font properties, i.e., through simple **ListFonts** requests. (E.g., if a user selected a Helvetica typeface family, a client application should be able to query the server for all Helvetica fonts and present only those setwidths, weights, slants, point sizes, character sets available for that family).

This document gives a standard logical font description (XLFD) and conventions to be used in the X protocol so that clients can query and access screen type libraries in a consistent manner across all X servers. In addition to completely specifying a given font via its **FontName**, the XLFD also provides for a standard set of key **FontProperties** that describe the font in more detail.

The XLFD provides an adequate set of typographic font properties, such as CAP_HEIGHT, X_HEIGHT, RELATIVE_SETWIDTH, for publishing and other applications to do intelligent font matching or substitution when handling documents created on some foreign server using potentially unknown fonts. In addition, this information is required by certain clients to automatically place subscripts, determine small capital heights, recommended leading, wordspace values, etc.

Examples are for illustrative purposes only.

2. Requirements and Goals

This specification meets the short and long-term goals to have a standard logical font description which:

- provides unique, descriptive font names that support simple pattern matching;
- supports multiple font vendors, arbitrary character sets and encodings;
- is X server and operating/file system independent;
- provides adequate descriptive font information for arbitrarily complex font matching/substitution;
- is extensible.

2.1. Unique, Descriptive Font Names

It should be possible to have font names that are long enough and descriptive enough to have a reasonable probability of being unique without inventing a new registration organization. Resolution/size-dependent font masters, multi-vendor font libraries, etc., must be anticipated and handled by the font name alone.

The name itself should be structured to be amenable to simple pattern matching and parsing, allowing X clients to restrict font queries to some subset of all possible fonts in the server.

2.2. Support Multiple Font Vendors and Character Sets

The font name and properties should distinguish between fonts that were supplied by different font vendors but that possibly share the same name. We anticipate a highly competitive font market where users will be able to buy fonts from a number of sources according to their particular requirements.

A number of font vendors deliver each font with all glyphs designed for that font, where charset mappings are defined by encoding vectors. Some server implementations may force these mappings to proprietary or standard charsets statically in the font data, while others may desire to perform the mapping dynamically in the server. Provisions must be made in the font name which allows a font request to specify/identify specific charset mappings in server environments where multiple charsets are supported.

2.3. Server, Operating and File System Independent

X client applications that require a particular font should be able to use the descriptive name, without knowledge of the file system or other repository in use by the server. However, it should be possible for servers to translate a given font name into a file name syntax that it knows how to deal with, without compromising the uniqueness of the font name. This algorithm should be reversible (exactly how this translation is done is implementation dependent).

2.4. Supports Arbitrarily Complex Font Matching/Substitution

In addition to the font name, the XLFD should define a standard list of descriptive font properties with agreed upon fallbacks for all fonts, so that client applications can derive font-specific formatting/display data, perform font matching/substitution when asked to handle potentially unknown fonts, as required.

2.5. Extensible

The XLFD must be extensible so that new and/or private descriptive font properties can be added to conforming fonts without obsoleting existing X client or server implementations.

3. X Logical Font Description

XLFD is divided into two basic components: the **FontName**, which gives all font information needed to uniquely identify a font in X protocol requests (e.g., **OpenFont**, **ListFonts**, etc.) and a variable list of optional **FontProperties** that describe a font in more detail.

The **FontName** is used in font queries and returned as data in certain X protocol requests. The **FontName** is also specified as the data value for the **FONT** item in the X Consortium Character Bitmap Distribution Format specification (BDF V2.1).

The **FontProperties** are supplied on a font-by-font basis, and are returned as data in certain X protocol requests as part of the **XFontStruct** data structure. The **FontProperties** names and associated data values may also appear as items of the **STARTPROPERTIES...ENDPROPERTIES** list in the BDF V2.1 specification.

3.1. FontName

The **FontName** is logically composed of two strings: a **FontNameRegistry** prefix, followed by a **FontNameSuffix**. The **FontNameRegistry** is an x-registered-name that identifies the registration authority that owns the specified **FontNameSuffix** syntax and semantics.

All font names that conform to this specification are to use a **FontNameRegistry** prefix defined to be the string "–", i.e., ISO 8859-1 HYPHEN (Column/Row 02/13). All **FontNameRegistry** prefixes of the form "+*version*–", where *version* is the version of some future XLFD specification, are reserved by the X Consortium for future extensions to XLFD font names. If required, extensions to the current XLFD font name shall be constructed by appending new fields to the current structure, each delimited by the existing field delimiter. The availability of other **FontNameRegistry** prefixes or fonts that support other registries is server implementation dependent.

In the X protocol specification, the **FontName** is required to be a string; hence, numeric field values are represented in the name as string equivalents. All **FontNameSuffix** fields are also defined as **FontProperties**, in which case numeric property values are represented as signed or unsigned integers as appropriate.

3.1.1. FontName Syntax

The **FontName** is a structured, parsable string (X data type STRING8) whose Backus-Naur Form syntax description is as follows:

FontName ::=	XFontNameRegistry XFontNameSuffix | PrivFontNameRegistry PrivFontNameSuffix
XFontNameRegistry ::=	XFNDelim | XFNExtPrefix Version XFNDelim
XFontNameSuffix ::=	FOUNDRY XFNDelim FAMILY_NAME XFNDelim WEIGHT_NAME XFNDelim SLANT XFNDelim SETWIDTH_NAME XFNDelim ADD_STYLE_NAME XFNDelim PIXEL_SIZE XFNDelim POINT_SIZE XFNDelim RESOLUTION_X XFNDelim RESOLUTION_Y XFNDelim SPACING XFNDelim AVERAGE_WIDTH XFNDelim CHARSET_REGISTRY XFNDelim CHARSET_ENCODING
Version ::=	STRING8 – the XLFD version that defines an extension to the font name syntax (e.g., "2.0")
XFNExtPrefix ::=	OCTET – the value of ISO8859-1 PLUS (Column/Row 02/13)
XFNDelim ::=	OCTET – the value of ISO8859-1 HYPHEN (Column/Row 02/13)
PrivFontNameRegistry ::=	STRING8 – other than those strings reserved by XLFD
PrivFontNameSuffix ::=	STRING8

Field values are constructed as strings of ISO8859-1 graphic characters, excluding the following:

- HYPHEN (02/13), the XLFD font name delimiter character;
- QUESTION MARK (03/15) and ASTERISK (02/10), the X protocol fontname wildcard characters.

Alphabetic case distinctions are allowed, but are for human readability concerns only. Conforming X servers will perform matching on font name query/open requests independent of case. The entire font name string must have no more than 255 characters. It is recommended that clients construct font name query patterns by explicitly including all field delimiters to avoid unexpected results. Note, that SPACE is a valid character of a **FontName** field; e.g., a FAMILY_NAME might be ITC Avant Garde Gothic.

3.1.2. FontName Field Definitions

3.1.2.1. FOUNDRY : x-registered-name

FOUNDRY is an x-registered-name, the name or identifier of the digital type foundry that digitized and supplied the font data, or if different, the identifier of the organization that last modified the font shape or metric information.

The reason this distinction is necessary is that a given font design may be licensed from one source (e.g., ITC) but digitized and sold by any number of different type suppliers. Each digital version of the original design will in general be somewhat different in metrics and shape from the idealized original font data, as each font foundry, for better or for worse, has its own standards and practices for tweaking a typeface for a particular generation of output technologies, or has its own perception of market needs.

It is up to the type supplier to register with the X Consortium a suitable name for this **FontName** field, according to the registration procedures defined by the Consortium.

The X Consortium shall define procedures for registering foundry names, and shall maintain and publish in a timely manner a registry of such registered names for use in XLFD font names and properties.

3.1.2.2. FAMILY_NAME : string

FAMILY_NAME is a string that identifies the range or "family" of typeface designs that are all variations of one basic typographic style. This must be spelled out in full, with words separated by spaces as required. This name must be human-understandable and suitable for presentation to a font user to identify the typeface family.

It is up to the type supplier to supply and maintain a suitable string for this field and font property, to secure the proper legal title to a given name, and to guard against the infringement of other's copyrights or trademarks. By convention, FAMILY_NAME is not translated. FAMILY_NAME may include an indication of design ownership if considered a valid part of the typeface family name (see examples below).

Examples of FAMILY_NAMEs:

Helvetica
ITC Avant Garde Gothic
Times
Times Roman
Bitstream Amerigo
Stone

3.1.2.3. WEIGHT_NAME : string

WEIGHT_NAME is a string that identifies the font's typographic weight, i.e., the nominal blackness of the font, according to the FOUNDRY's judgement. This name must be human-understandable and suitable for presentation to a font user.

The interpretation of this field is somewhat problematic, as the typographic judgement of weight has traditionally depended on the overall design of the typeface family in question (i.e., it is possible that the DemiBold weight of one font could be almost equivalent in typographic feel to a Bold font from another family).

WEIGHT_NAME is captured as an arbitrary string since it is an important part of a font's complete human-understandable name, but it should not be used for font matching/substitution. X client applications should use the weight-related font properties (RELATIVE_WEIGHT and WEIGHT) that give the coded relative weight, and the calculated weight, respectively, for this purpose.

3.1.2.4. SLANT : code-string

SLANT is a code-string that indicates the overall posture of the typeface design used in the font. The encoding is as follows:

Code	English Translation	Description
"R"	Roman	Upright design
"I"	Italic	Italic design, slanted clockwise from vertical
"O"	Oblique	Obliqued upright design, slanted clockwise from vertical
"RI"	Reverse Italic	Italic design, slanted counter clockwise from vertical
"RO"	Reverse Oblique	Obliqued upright design, slanted counter clockwise from vertical
"OT"	Other	Other

The SLANT codes are for programming convenience only, and usually are converted into their equivalent human-understandable form before being presented to a user.

3.1.2.5. SETWIDTH_NAME : string

SETWIDTH_NAME is a string that gives the font's typographic proportionate width, i.e., the nominal width per horizontal unit of the font, according to the FOUNDRY's judgment.

As with WEIGHT_NAME, the interpretation of this field or font property is somewhat problematic, as the designer's judgment of setwidth has traditionally depended on the overall design of the typeface family in question. **X** client applications should use the RELATIVE_SETWIDTH font property which gives the relative coded proportionate width, or calculate the proportionate width, for purposes of font matching or substitution.

Examples of SETWIDTH_NAMEs:

Normal
Condensed
Narrow
Double Wide

3.1.2.6. ADD_STYLE_NAME : string

ADD_STYLE_NAME is a string that identifies additional typographic style information not captured by other fields, but needed to uniquely identify the font.

ADD_STYLE_NAME is not a typeface classification field, and is only used for uniqueness. Its usage, as such, is not limited to typographic style distinctions.

Examples of ADD_STYLE_NAMEs:

Serif
Sans Serif
Informal
Decorated

3.1.2.7. PIXEL_SIZE : integer-string

PIXEL_SIZE is an unsigned integer-string typographic metric in device pixels which gives the body size of the font at a particular POINT_SIZE and RESOLUTION_Y. PIXEL_SIZE normally incorporates additional vertical spacing considered part of the font design. (Note, however, that this value is not necessarily equivalent to the height of the font bounding box). PIXEL_SIZE is in the range zero to a "very-large-number".

PIXEL_SIZE would normally be used by **X** client applications that need to query fonts according to device-dependent size, regardless of the point size or vertical resolution the font was designed for.

3.1.2.8. POINT_SIZE : integer-string

POINT_SIZE is an unsigned integer-string typographic metric in device-independent units which gives the body size the font was designed for. This field normally incorporates additional vertical spacing considered part of the font design. (Note, however, that POINT_SIZE is not necessarily equivalent to the height of the font bounding box). POINT_SIZE is expressed in decipoints (where points are as defined in the **X** protocol or 72.27 pts = 1 inch) in the range zero to a "very-large-number".

POINT_SIZE and RESOLUTION_Y would be used by **X** clients to query fonts according to device-independent size, to maintain constant text size on the display regardless of the PIXEL_SIZE used for the font.

3.1.2.9. RESOLUTION_X : integer-string

3.1.2.10. RESOLUTION_Y : integer-string

RESOLUTION_X and RESOLUTION_Y are unsigned integer-strings, the horizontal and vertical resolution that the font was designed for, measured in pixels/dots per inch (dpi). Horizontal and vertical values are required since a separate bitmap font must be designed for displays with very different aspect ratios (e.g., 1:1, 4:3, 2:1, etc.).

The separation of pixel/point size and resolution is necessary because X allows for servers with very different video characteristics (e.g., horizontal and vertical resolution, screen and pixel size, pixel shape, etc.) to potentially access the same font library. The font name, then, must differentiate between a 14 point font designed for 75 dpi (body size of about 14 pixels) or a 14 point font designed for 150 dpi (about 28 pixels), etc. Further, in servers that implement some or all fonts as continuously scaled outlines, POINT_SIZE and RESOLUTION_Y will help the server to differentiate between potentially separate font masters for text, title, and display sizes or for other typographic considerations.

3.1.2.11. SPACING : code-string

SPACING is a code-string that indicates the escapement class of the font, i.e., monospace (fixed pitch), proportional (variable pitch), or charcell (a special monospaced font that conforms to the traditional data processing character cell font model).

Code	English Translation	Description
"P"	Proportional	A font whose logical character widths vary for each glyph. Note that no other restrictions are placed on the metrics of a proportional font.
"M"	Monospaced	A font whose logical character widths are constant (i.e., all char widths of the font are = max_bounds.width). No other restrictions are placed on the metrics of a monospaced font.
"C"	CharCell	A monospaced font which follows the standard typewriter character cell model (i.e., the glyphs of the font can be modeled by X clients as "boxes" of the same width and height which are imaged side by side to form text strings, or top to bottom to form text lines. By definition, all glyphs have the same logical character width, and no glyphs have "ink" outside of the character cell–there is no kerning (i.e., on a per char basis with positive metrics: 0 <= left-bearing <= right-bearing <= width; with negative metrics: width <= left-bearing <= right-bearing <= 0)-and the vertical extents of the font do not exceed the vertical spacing (i.e., on a per char basis: ascent <= font-ascent and descent <= font-descent). The cell height = font-descent + font-ascent, and width = AVERAGE_WIDTH.

3.1.2.12. AVERAGE_WIDTH : integer-string

AVERAGE_WIDTH is an unsigned integer-string typographic metric value giving the unweighted arithmetic mean width of all glyphs in the font, measured in 1/10th pixels. Note, for monospaced and character cell fonts, this is the width of all glyphs in the font.

3.1.2.13. CHARSET_REGISTRY : x-registered-name

3.1.2.14. CHARSET_ENCODING : registered-name

The character set used to encode the glyphs of the font (and implicitly the font's glyph repertoire), as maintained by the X Consortium character set registry. CHARSET_REGISTRY is an x-registered-name that identifies the registration authority that owns the specified encoding. CHARSET_ENCODING is a registered-name that identifies the coded character set as defined by that registration authority.

Although the X protocol does not explicitly have any knowledge about character set encodings, it is expected that server implementers will prefer to embed knowledge of certain proprietary or industry standard charsets into their font library for reasons of performance and convenience. The CHARSET_REGISTRY and CHARSET_ENCODING fields/properties allow an X client font request to specify a specific charset mapping in server environments where multiple charsets are supported. The availability of any particular character set is font and server implementation dependent.

To prevent collisions when defining character set names, it is recommended that CHARSET_REGISTRY/CHARSET_ENCODING name pairs be constructed according to the following conventions:

CharsetRegistry ::=	StdCharsetRegistryName | PrivCharsetRegistryName
CharsetEncoding ::=	StdCharsetEncodingName | PrivCharsetEncodingName
StdCharsetRegistryName ::=	StdOrganizationId StdNumber | StdOrganizationId StdNumber Dot Year
PrivCharsetRegistryName ::=	OrganizationId STRING8
StdCharsetEncodingName ::=	STRING8--numeric part # of referenced standard
PrivCharsetEncodingName ::=	STRING8
StdOrganizationId ::=	STRING8--the registered name or acronym of the referenced standard organization
StdNumber ::=	STRING8--referenced standard number
OrganizationId ::=	STRING8--the registered name or acronym of the organization
Dot ::=	"."--ISO 8859-1 FULL STOP (Column/Row 2/14)
Year ::=	STRING8--numeric year (for example, 1989)

The X Consortium shall maintain and publish in a timely manner a registry of such character set names for use in X protocol font names and properties as specified in XLFD.

The ISO Latin 1 character set shall be registered by the X Consortium as the CHARSET_REGISTRY-CHARSET_ENCODING value pair: "ISO8859-1".

3.1.3. Examples

The following examples of font names are derived from the screen fonts shipped with the R3 server.

Font	X FontName
75dpi Fonts	
Charter 12pt	-Bitstream-Charter-Medium-R-Normal--12-120-75-75-P-68-ISO8859-1
Charter Bold 12pt	-Bitstream-Charter-Bold-R-Normal--12-120-75-75-P-76-ISO8859-1
Charter BoldItalic 12pt	-Bitstream-Charter-Bold-I-Normal--12-120-75-75-P-75-ISO8859-1C
Charter Italic 12pt	-Bitstream-Charter-Medium-I-Normal--12-120-75-75-P-66-ISO8859-1
Courier 8pt	-Adobe-Courier-Medium-R-Normal--8-80-75-75-M-50-ISO8859-1
Courier 10pt	-Adobe-Courier-Medium-R-Normal--10-100-75-75-M-60-ISO8859-1
Courier 12pt	-Adobe-Courier-Medium-R-Normal--12-120-75-75-M-70-ISO8859-1
Courier 14pt	-Adobe-Courier-Medium-R-Normal--14-140-75-75-M-90-ISO8859-1
Courier 18pt	-Adobe-Courier-Medium-R-Normal--18-180-75-75-M-110-ISO8859-1
Courier 24pt	-Adobe-Courier-Medium-R-Normal--24-240-75-75-M-150-ISO8859-1

Font	X FontName
Courier Bold 10pt	-Adobe-Courier-Bold-R-Normal--10-100-75-75-M-60-ISO8859-1
Courier BoldOblique 10pt	-Adobe-Courier-Bold-O-Normal--10-100-75-75-M-60-ISO8859-1
Courier Oblique 10pt	-Adobe-Courier-Medium-O-Normal--10-100-75-75-M-60-ISO8859-1
100dpi Fonts	
Symbol 8pt	-Adobe-Symbol-Medium-R-Normal--11-80-100-100-P-61-Adobe-FONTSPECIFIC
Symbol 10pt	-Adobe-Symbol-Medium-R-Normal--14-100-100-100-P-85-Adobe-FONTSPECIFIC
Symbol 12pt	-Adobe-Symbol-Medium-R-Normal--17-120-100-100-P-95-Adobe-FONTSPECIFIC
Symbol 14pt	-Adobe-Symbol-Medium-R-Normal--20-140-100-100-P-107-Adobe-FONTSPECIFIC
Symbol 18pt	-Adobe-Symbol-Medium-R-Normal--25-180-100-100-P-142-Adobe-FONTSPECIFIC
Symbol 24pt	-Adobe-Symbol-Medium-R-Normal--34-240-100-100-P-191-Adobe-FONTSPECIFIC
Times Bold 10pt	-Adobe-Times-Bold-R-Normal--14-100-100-100-P-76-ISO8859-1
Times BoldItalic 10pt	-Adobe-Times-Bold-I-Normal--14-100-100-100-P-77-ISO8859-1
Times Italic 10pt	-Adobe-Times-Medium-I-Normal--14-100-100-100-P-73-ISO8859-1
Times Roman 10pt	-Adobe-Times-Medium-R-Normal--14-100-100-100-P-74-ISO8859-1

3.2. FontProperties

All font properties are optional, but will generally include the font name fields, and on a font-by-font basis any other useful font descriptive/usage information that may be required to use the font intelligently. The XLFD specifies an extensive set of standard X font properties, their interpretation and fallback rules when the property is not defined for a given font. The goal is to provide client applications with enough font information to be able to make automatic formatting/display decisions with good typographic results.

Additional standard X font property definitions may be defined in the future and private properties may exist in X fonts at any time. Private font properties should be defined to conform to the general mechanism defined in the X protocol to prevent overlap of name space and ambiguous property names, i.e., private font property names are of the form: ISO8859-1 UNDERSCORE (Column/Row 05/15), followed by the organizational identifier, followed by UNDERSCORE, and terminated with the property name.

The Backus-Naur Form syntax description of X Font Properties is:

```
Properties ::=     OptFontPropList
OptFontPropList ::= NULL | OptFontProp OptFontPropList
OptFontProp ::=    PrivateFontProp | XFontProp
PrivateFontProp ::= STRING8 | Underscore OrganizationId Underscore STRING8
```

XFontProp ::=	FOUNDRY | FAMILY_NAME | WEIGHT_NAME | SLANT | SETWIDTH_NAME | ADD_STYLE_NAME | PIXEL_SIZE | POINT_SIZE | RESOLUTION_X | RESOLUTION_Y | SPACING | AVERAGE_WIDTH | CHARSET_REGISTRY | CHARSET_ENCODING | QUAD_WIDTH | RESOLUTION | MIN_SPACE | NORM_SPACE | MAX_SPACE | END_SPACE | SUPERSCRIPT_X | SUPERSCRIPT_Y | SUBSCRIPT_X | SUBSCRIPT_Y | UNDERLINE_POSITION | UNDERLINE_THICKNESS | STRIKEOUT_ASCENT | STRIKEOUT_DESCENT | ITALIC_ANGLE | X_HEIGHT | WEIGHT | FACE_NAME | COPYRIGHT | AVG_CAPITAL_WIDTH | AVG_LOWERCASE_WIDTH | RELATIVE_SETWIDTH | RELATIVE_WEIGHT | CAP_HEIGHT | SUPERSCRIPT_SIZE | FIGURE_WIDTH | SUBSCRIPT_SIZE | SMALL_CAP_SIZE | NOTICE | DESTINATION
Underscore ::=	OCTET-the value of ISO8859-1 UNDERSCORE character (Column/Row 05/15)
OrganizationId ::=	STRING8-the registered name of the organization

3.2.1. FOUNDRY : ATOM

As defined in the **FontName**, except the property type is ATOM.

FOUNDRY can not be calculated or defaulted if not supplied as a font property.

3.2.2. FAMILY_NAME : ATOM

As defined in the **FontName**, except the property type is ATOM.

FAMILY_NAME can not be calculated or defaulted if not supplied as a font property.

3.2.3. WEIGHT_NAME : ATOM

As defined in the **FontName**, except the property type is ATOM.

WEIGHT_NAME can be defaulted if not supplied as a font property, as follows:

```
if (WEIGHT_NAME undefined) then
  WEIGHT_NAME = ATOM(''Medium'')
```

3.2.4. SLANT : ATOM

As defined in the **FontName**, except the property type is ATOM.

SLANT can be defaulted if not supplied as a font property, as follows:

```
if (SLANT undefined) then
  SLANT = ATOM(''R'')
```

3.2.5. SETWIDTH_NAME : ATOM

As defined in the **FontName**, except the property type is ATOM.

SETWIDTH_NAME can be defaulted if not supplied as a font property, as follows:

```
if (SETWIDTH_NAME undefined) then
  SETWIDTH_NAME = ATOM(''Normal'')
```

3.2.6. ADD_STYLE_NAME : ATOM

As defined in the **FontName**, except the property type is ATOM.

ADD_STYLE_NAME can be defaulted if not supplied as a font property, as follows:

```
if (ADD_STYLE_NAME undefined) then
  ADD_STYLE_NAME = ATOM("")
```

3.2.7. PIXEL_SIZE : CARD32

As defined in the **FontName**, except the property type is CARD32.

X clients requiring pixel values for the various typographic fixed spaces (EM space, EN space and THIN space), can use the following algorithm for computing these values from other properties specified for a font:

```
DeciPointsPerInch = 722.7
EMspace = ROUND ((RESOLUTION_X * POINT_SIZE) / DeciPointsPerInch)
ENspace = ROUND (EMspace / 2)
THINspace = ROUND (EMspace / 3)
```

Note that a "/" denotes real division, "*" denotes real multiplication, and "ROUND" denotes a function that rounds its real argument "a" up/down to the next integer, according to x = FLOOR(a + 0.5), where FLOOR is a function that rounds its argument down to an integer.

PIXEL_SIZE can be approximated if not supplied as a font property, according to the algorithm:

```
DeciPointsPerInch = 722.7
if (PIXEL_SIZE undefined) then
  PIXEL_SIZE = ROUND ((RESOLUTION_Y * POINT_SIZE) / DeciPointsPerInch)
```

3.2.8. POINT_SIZE : CARD32

As defined in the **FontName**, except the property type is CARD32.

X clients requiring device-independent values for EMspace, ENspace and THINspace, can use the following algorithm:

```
EMspace = ROUND (POINT_SIZE / 10)
ENspace = ROUND (POINT_SIZE / 20)
THINspace = ROUND (POINT_SIZE / 30)
```

Design POINT_SIZE can not be calculated or approximated.

3.2.9. RESOLUTION_X : CARD32

As defined in the **FontName**, except the property type is CARD32.

RESOLUTION_X cannot be calculated or approximated.

3.2.10. RESOLUTION_Y : CARD32

As defined in the **FontName**, except the property type is CARD32.

RESOLUTION_Y cannot be calculated or approximated.

3.2.11. SPACING : ATOM

As defined in the **FontName**, except the property type is ATOM.

SPACING can be calculated if not supplied as a font property, according to the definitions given above for the **FontName**.

3.2.12. AVERAGE_WIDTH : CARD32

As defined in the **FontName**, except the property type is CARD32.

AVERAGE_WIDTH can be calculated if not provided as a font property, according to the following algorithm:

```
if (AVERAGE_WIDTH undefined) then
  AVERAGE_WIDTH = ROUND (MEAN (all glyph widths in font) * 10)
```

where MEAN is a function that returns the arithmetic mean of its arguments.

X clients requiring values for the number of characters per inch (pitch) of a monospaced font can use the following algorithm using the AVERAGE_WIDTH and RESOLUTION_X font properties:

```
if (SPACING not proportional) then
  CharPitch = (RESOLUTION_X * 10) / AVERAGE_WIDTH
```

3.2.13. CHARSET_REGISTRY : ATOM

As defined in the **FontName**, except the property type is ATOM.

CHARSET_REGISTRY can not be defaulted if not supplied as a font property.

3.2.14. CHARSET_ENCODING : ATOM

As defined in the **FontName**, except the property type is ATOM.

CHARSET_ENCODING can not be defaulted if not supplied as a font property.

3.2.15. MIN_SPACE : CARD32

MIN_SPACE is an unsigned integer value that gives the recommended minimum wordspace value to be used with this font.

MIN_SPACE can be approximated if not provided as a font property, according to the algorithm:

```
if (MIN_SPACE undefined) then
  MIN_SPACE = ROUND(0.75 * NORM_SPACE)
```

3.2.16. NORM_SPACE : CARD32

NORM_SPACE is an unsigned integer value that gives the recommended normal wordspace value to be used with this font.

NORM_SPACE can be approximated if not provided as a font property, according to the following algorithm:

```
DeciPointsPerInch = 722.7
if (NORM_SPACE undefined) then
  if (SPACE glyph exists) then
    NORM_SPACE = width of SPACE
  else NORM_SPACE = ROUND((0.33 * RESOLUTION_X * POINT_SIZE) /
                          DeciPointsPerInch)
```

3.2.17. MAX_SPACE : CARD32

MAX_SPACE is an unsigned integer value that gives the recommended maximum wordspace value to be used with this font.

MAX_SPACE can be approximated if not provided as a font property, according to the following algorithm:

```
if (MAX_SPACE undefined) then
  MAX_SPACE = ROUND(1.5 * NORM_SPACE)
```

3.2.18. END_SPACE : CARD32

END_SPACE is an unsigned integer value that gives the recommended spacing at the end of sentences.

END_SPACE can be approximated if not provided as a font property, according to the following algorithm:

```
if (END_SPACE undefined) then
  END_SPACE = NORM_SPACE
```

3.2.19. AVG_CAPITAL_WIDTH : INT32

AVG_CAPITAL_WIDTH is an integer value that gives the unweighted arithmetic mean width of all the capital glyphs in the font, in 1/10th pixels (applies to Latin and non-Latin fonts). For Latin fonts, capitals are the glyphs A-Z. Normally used for font matching/substitution.

AVG_CAPITAL_WIDTH can be calculated if not provided as a font property, according to the following algorithm:

```
if (AVG_CAPITAL_WIDTH undefined) then
  AVG_CAPITAL_WIDTH = ROUND (MEAN (capital glyph widths) * 10)
```

Note that MEAN is a function that returns the arithmetic mean of its arguments.

3.2.20. AVG_LOWERCASE_WIDTH : INT32

AVG_LOWERCASE_WIDTH is an integer value that gives the unweighted arithmetic mean width of all the lower case glyphs in the font in 1/10th pixels. For Latin fonts, lower case are the glyphs a-z. Normally used for font matching or substitution.

Where appropriate, AVG_LOWERCASE_WIDTH can be approximated if not provided as a font property, according to the following algorithm:

```
if (AVG_LOWERCASE_WIDTH undefined) then
  if (lower case exists) then
    AVG_LOWERCASE_WIDTH = ROUND (MEAN (lower case glyph widths) * 10)
  else AVG_LOWERCASE_WIDTH undefined
```

3.2.21. QUAD_WIDTH : INT32 (DEPRECATED)

QUAD_WIDTH was incorrectly defined in the X protocol, and is redundant since all typographic fixed spaces (EM, EN and THIN) are constant for a given font size (i.e., they do not vary according to setwidth). X clients requiring these properties are encouraged to discontinue usage of QUAD_WIDTH and compute these values from other font properties. X clients requiring a font-dependent width value should use either the FIGURE_WIDTH or one of the average character width font properties (AVERAGE_WIDTH, AVG_CAPITAL_WIDTH or AVG_LOWERCASE_WIDTH) for this purpose.

See also PIXEL_SIZE, FIGURE_WIDTH, AVERAGE_WIDTH, AVG_CAPITAL_WIDTH and AVG_LOWERCASE_WIDTH font property definitions.

3.2.22. FIGURE_WIDTH : INT32

FIGURE_WIDTH is an integer typographic metric that gives the width of the tabular figures and the dollar sign, if suitable for tabular setting (all widths equal). For Latin fonts, these tabular figures are the arabic numerals 0-9.

FIGURE_WIDTH can be approximated if not supplied as a font property, according to the following algorithm:

```
if (numerals and DOLLAR sign are defined & widths are equal) then
   FIGURE_WIDTH = width of DOLLAR
else FIGURE_WIDTH property undefined
```

3.2.23. SUPERSCRIPT_X : INT32

SUPERSCRIPT_X is an integer value that gives the recommended horizontal offset in pixels from the position point to the X origin of synthetic superscript text. If the current position point is at [X,Y], then superscripts should begin at [X + SUPERSCRIPT_X, Y - SUPERSCRIPT_Y].

SUPERSCRIPT_X can be approximated, if not provided as a font property, according to the following algorithm:

```
if (SUPERSCRIPT_X undefined) then
   if (TANGENT(ITALIC_ANGLE) defined) then
      SUPERSCRIPT_X = ROUND((0.40 * CAP_HEIGHT) / TANGENT(ITALIC_ANGLE))
   else SUPERSCRIPT_X = ROUND(0.40 * CAP_HEIGHT)
```

Note that TANGENT is a trigonometric function that returns the tangent of its argument (in degrees scaled by 64).

3.2.24. SUPERSCRIPT_Y : INT32

SUPERSCRIPT_Y is an integer value that gives the recommended vertical offset in pixels from the position point to the Y origin of synthetic superscript text. If the current position point is at [X,Y], then superscripts should begin at [X + SUPERSCRIPT_X, Y - SUPERSCRIPT_Y].

SUPERSCRIPT_Y can be approximated, if not provided as a font property, according to the following algorithm:

```
if (SUPERSCRIPT_Y undefined) then
   SUPERSCRIPT_Y = ROUND(0.40 * CAP_HEIGHT)
```

3.2.25. SUBSCRIPT_X : INT32

SUBSCRIPT_X is an integer value that gives the recommended horizontal offset in pixels from the position point to the X origin of synthetic subscript text. If the current position point is at [X,Y], then subscripts should begin at [X + SUBSCRIPT_X, Y + SUBSCRIPT_Y].

SUBSCRIPT_X can be approximated, if not provided as a font property, according to the following algorithm:

```
if (SUBSCRIPT_X undefined) then
   if (TANGENT(ITALIC_ANGLE) defined) then
      SUBSCRIPT_X = ROUND((0.40 * CAP_HEIGHT) / TANGENT(ITALIC_ANGLE))
   else SUBSCRIPT_X = ROUND(0.40 * CAP_HEIGHT)
```

3.2.26. SUBSCRIPT_Y : INT32

SUBSCRIPT_Y is an integer value that gives the recommended vertical offset in pixels from the position point to the Y origin of synthetic subscript text. If the current position point is at [X,Y], then subscripts should begin at [X + SUBSCRIPT_X, Y + SUBSCRIPT_Y].

SUBSCRIPT_Y can be approximated, if not provided as a font property, according to the following algorithm:

```
if (SUBSCRIPT_Y undefined) then
  SUBSCRIPT_Y = ROUND(0.40 * CAP_HEIGHT)
```

3.2.27. SUPERSCRIPT_SIZE : CARD32

SUPERSCRIPT_SIZE is an unsigned integer value that gives the recommended body size of synthetic superscripts to be used with this font, in pixels. Note that this will generally be smaller than the size of the current font; i.e., superscripts are imaged from a smaller font, offset according to SUPERSCRIPT_X and SUPERSCRIPT_Y.

SUPERSCRIPT_SIZE can be approximated if not provided as a font property, according to the following algorithm:

```
if (SUPERSCRIPT_SIZE undefined) then
  SUPERSCRIPT_SIZE = ROUND(0.60 * PIXEL_SIZE)
```

3.2.28. SUBSCRIPT_SIZE : CARD32

SUBSCRIPT_SIZE is an unsigned integer value that gives the recommended body size of synthetic subscripts to be used with this font, in pixels. As with SUPERSCRIPT_SIZE, this will generally be smaller than the size of the current font; i.e., subscripts are imaged from a smaller font, offset according to SUBSCRIPT_X and SUBSCRIPT_Y.

SUBSCRIPT_SIZE can be approximated if not provided as a font property, according to the algorithm:

```
if (SUBSCRIPT_SIZE undefined) then
  SUBSCRIPT_SIZE = ROUND(0.60 * PIXEL_SIZE)
```

3.2.29. SMALL_CAP_SIZE : CARD32

SMALL_CAP_SIZE is an integer value that gives the recommended body size of synthetic small capitals to be used with this font, in pixels. Small capitals are generally imaged from a smaller font, of slightly more weight. No offset [X,Y] is necessary.

SMALL_CAP_SIZE can be approximated if not provided as a font property, according to the following algorithm:

```
if (SMALL_CAP_SIZE undefined) then
  SMALL_CAP_SIZE = ROUND(PIXEL_SIZE * ((X_HEIGHT
                   + ((CAP_HEIGHT - X_HEIGHT) / 3)) / CAP_HEIGHT))
```

3.2.30. UNDERLINE_POSITION : INT32

UNDERLINE_POSITION is an integer value that gives the recommended vertical offset in pixels from the baseline to the top of the underline. If the current position point is at [X,Y], the top of the baseline is given by [X, Y + UNDERLINE_POSITION].

UNDERLINE_POSITION can be approximated if not provided as a font property, according to the following algorithm:

```
if (UNDERLINE_POSITION undefined) then
  UNDERLINE_POSITION = ROUND(max_bounds.descent / 2)
```

3.2.31. UNDERLINE_THICKNESS : CARD32

UNDERLINE_POSITION is an unsigned integer value that gives the recommended underline thickness, in pixels.

UNDERLINE_THICKNESS can be approximated if not provided as a font property, according to the following algorithm:

```
CapStemWidth = average width of the stems of capitals
if (UNDERLINE_THICKNESS undefined) then
  UNDERLINE_THICKNESS = CapStemWidth
```

3.2.32. STRIKEOUT_ASCENT : INT32

STRIKEOUT_ASCENT is an integer value that gives the vertical ascent for boxing or voiding glyphs in this font. If the current position is at [X,Y] and the string extent is EXTENT, the upper-left comer of the strikeout box is at [X, Y - STRIKEOUT_ASCENT] and the lower-right comer of the box is at [X + EXTENT, Y + STRIKEOUT_DESCENT].

STRIKEOUT_ASCENT can be approximated if not provided as a font property, according to the following algorithm:

```
if (STRIKEOUT_ASCENT undefined)
  STRIKEOUT_ASCENT =  max_bounds.ascent
```

3.2.33. STRIKEOUT_DESCENT : INT32

STRIKEOUT_DESCENT is an integer value that gives the vertical descent for boxing or voiding glyphs in this font. If the current position is at [X,Y] and the string extent is EXTENT, the upper-left comer of the strikeout box is at [X, Y - STRIKEOUT_ASCENT] and the lower-right comer of the box is at [X + EXTENT, Y + STRIKEOUT_DESCENT].

STRIKEOUT_DESCENT can be approximated if not provided as a font property, according to the following algorithm:

```
if (STRIKEOUT_DESCENT undefined)
  STRIKEOUT_DESCENT =  max_bounds.descent
```

3.2.34. ITALIC_ANGLE : INT32

ITALIC_ANGLE is an integer value that gives the nominal posture angle of the typeface design, in 1/64 degrees, measured from the glyph origin counterclockwise from the three o'clock positon.

ITALIC_ANGLE can be defaulted if not provided as a font property, according to the following algorithm:

```
if (ITALIC_ANGLE undefined) then
  ITALIC_ANGLE = (90 * 64)
```

3.2.35. CAP_HEIGHT : CARD32

CAP_HEIGHT is an unsigned integer, the nominal height of the capital letters contained in the font, as specified by the FOUNDRY or typeface designer. Where applicable, it is defined to be the height of the Latin upper case letter X.

CAP_HEIGHT is required by certain clients to compute scale factors and positioning offsets for algorithmically generated glyphs where this information or designed glyphs are not explicitly provided by the font (e.g., small capitals, superiors, inferiors, etc.). Capital height is also a critical factor in font matching and substitution.

CAP_HEIGHT can be approximated if not provided as a font property, according to the following algorithm:

```
if (CAP_HEIGHT undefined) then
  if (Latin font) then
    CAP_HEIGHT = XCharStruct.ascent[glyph X]
  else if (capitals exist) then
     CAP_HEIGHT = XCharStruct.ascent[some capital glyph]
  else CAP_HEIGHT undefined
```

3.2.36. X_HEIGHT : CARD32

X_HEIGHT is a unsigned integer, the nominal height above the baseline of the lower case glyphs contained in the font, as specified by the FOUNDRY or typeface designer. Where applicable, it is defined to be the height of the Latin lower case letter x.

As with Capital height, X_HEIGHT is required by certain clients to compute scale factors for algorithmically generated small capitals, where not explicitly provided by the font resource and is a critical factor in font matching and substitution.

X_HEIGHT can be approximated if not provided as a font property, according to the following algorithm:

```
if (X_HEIGHT undefined) then
  if (Latin font) then
    X_HEIGHT = XCharStruct.ascent[glyph x]
  else if (lower case exists) then
     X_HEIGHT = XCharStruct.ascent[some lower case glyph]
  else X_HEIGHT is undefined
```

3.2.37. RELATIVE_SETWIDTH : CARD32

RELATIVE_SETWIDTH is an integer that gives the coded proportionate width of the font, relative to all known fonts of the same typeface family, according to the type designer's or FOUNDRY's judgement.

The possible values are:

Code	English String	Description
0	undefined	Undefined or unknown
10	UltraCondensed	Lowest ratio of average width to height
20	ExtraCondensed	
30	Condensed	Condensed, Narrow, Compressed, ...
40	SemiCondensed	
50	Medium	Medium, Normal, Regular, ...
60	SemiExpanded	SemiExpanded, DemiExpanded, ...
70	Expanded	
80	ExtraExpanded	ExtraExpanded, Wide, ...
90	UltraExpanded	Highest ratio of average width to height

RELATIVE_SETWIDTH can be defaulted if not provided as a font property, according to the following algorithm:

```
if (RELATIVE_SETWIDTH undefined) then
  RELATIVE_SETWIDTH = 50
```

X clients that wish to obtain a calculated proportionate width of the font (i.e., a font-independent way of identifying the proportionate width across all fonts and all font vendors) can use the following algorithm:

SETWIDTH = AVG_CAPITAL_WIDTH / (CAP_HEIGHT * 10)

Note that SETWIDTH is a real with 0 being the "narrowest" calculated setwidth.

3.2.38. RELATIVE_WEIGHT : CARD32

RELATIVE_WEIGHT is an integer that gives the coded weight of the font, relative to all known fonts of the same typeface family, according to the type designer's or FOUNDRY's judgement.

The possible values are:

Code	English String	Description
0	undefined	Undefined or unknown
10	UltraLight	Lowest ratio of stem width to height
20	ExtraLight	
30	Light	
40	SemiLight	SemiLight, Book, ...
50	Medium	Medium, Normal, Regular,...
60	SemiBold	SemiBold, DemiBold, ...
70	Bold	
80	ExtraBold	ExtraBold, Heavy, ...
90	UltraBold	UltraBold, Black, ..., the highest ratio of stem width to height

RELATIVE_WEIGHT can be defaulted if not provided as a font property, according to the following algorithm:

```
if (RELATIVE_WEIGHT undefined) then
  RELATIVE_WEIGHT = 50
```

3.2.39. WEIGHT : CARD32

Calculated WEIGHT is an unsigned integer, the calculated weight of the font, computed as the ratio of capital stem width to CAP_HEIGHT, in the range 0 to 1000, where zero is the "lightest" weight.

WEIGHT can be calculated if not supplied as a font property, according to the following algorithm:

```
CapStemWidth = average width of the stems of capitals
if (WEIGHT undefined) then
  WEIGHT = ROUND ((CapStemWidth * 1000) / CAP_HEIGHT)
```

A calculated value for weight is necessary when matching fonts from different families because both the RELATIVE_WEIGHT and the WEIGHT_NAME are assigned by the typeface supplier, according to its tradition and practice, and therefore somewhat subjective. Calculated WEIGHT provides a font-independent way of identifying the weight across all fonts and all font vendors.

3.2.40. RESOLUTION : CARD32 (DEPRECATED)

Independent horizontal and vertical design resolution components are required to accomodate displays with nonsquare aspect ratios and are given by the RESOLUTION_X and RESOLUTION_Y font name fields/properties. The units of the original definition of RESOLUTION are also in conflict with these new properties. X clients are encouraged to discontinue

usage of RESOLUTION and to use the appropriate X,Y resolution properties as required.

3.2.41. FACE_NAME : ATOM

FACE_NAME is a human-understandable string that gives the full device-independent typeface name, including the owner, weight, slant, set, etc., but not the resolution, size, etc. Normally used as feedback during font selection.

FACE_NAME can not be calculated or approximated if not provided as a font property.

3.2.42. COPYRIGHT : ATOM

COPYRIGHT is a human-understandable string that gives the copyright information of the legal owner of the digital font data.

This information is a required component of a font but is independent of the particular format used to represent it (i.e., it cannot be captured as a comment that could later be "thrown away" for efficiency reasons).

COPYRIGHT can not be calculated or approximated if not provided as a font property.

3.2.43. NOTICE : ATOM

NOTICE is a human-understandable string that gives the copyright information of the legal owner of the font design, or if not applicable, the trademark information for the typeface FAMILY_NAME.

Typeface design and trademark protection laws vary from country to country, the USA having no design copyright protection currently, while various countries in Europe offer both design and typeface family name trademark protection. As with COPYRIGHT, this information is a required component of a font but is independent of the particular format used to represent it.

NOTICE can not be calculated or approximated if not provided as a font property.

3.2.44. DESTINATION : CARD32

DESTINATION is an unsigned integer code that gives the font design destination, i.e., whether it was designed as a screen proofing font to match printer font glyph widths (WYSIWYG), as an optimal video font (possibly with corresponding printer font) for extended screen viewing (VideoText), etc.

The font design considerations are very different, and at current display resolutions, the readability and legibility of these two kinds of screen fonts are very different. DESTINATION allows publishing clients that use X to model the printed page, and Video Text clients such as on-line documentation browsers, to query for X screen fonts that suit their particular requirements.

The encoding is as follows:

Code	English String	Description
0	WYSIWYG	A font optimized to match the typographic design and metrics of an equivalent printer font
1	Video Text	A font optimized for screen legibility and readability

3.3. Built-in Font Property Atoms

The following font property atom definitions were predefined in the initial Version 11 of the X protocol:

Font Property	Property Type

Font Property	Property Type
MIN_SPACE	CARD32
NORM_SPACE	CARD32
MAX_SPACE	CARD32
END_SPACE	CARD32
SUPERSCRIPT_X	INT32
SUPERSCRIPT_Y	INT32
SUBSCRIPT_X	INT32
SUBSCRIPT_Y	INT32
UNDERLINE_POSITION	INT32
UNDERLINE_THICKNESS	CARD32
STRIKEOUT_ASCENT	INT32
STRIKEOUT_DESCENT	INT32
FONT_ASCENT	INT32
FONT_DESCENT	INT32
ITALIC_ANGLE	INT32
X_HEIGHT	INT32
QUAD_WIDTH	INT32 – deprecated
WEIGHT	CARD32
POINT_SIZE	CARD32
RESOLUTION	CARD32 – deprecated
COPYRIGHT	ATOM
FACE_NAME	ATOM
FAMILY_NAME	ATOM
DEFAULT_CHAR	CARD32

4. Affected Elements of Xlib and the X Protocol

The following X protocol requests must use the font naming conventions:

- **OpenFont** – for the name parameter
- **ListFonts** – for the pattern parameter
- **ListFontsWithInfo** – for the pattern parameter

In addition, the following Xlib functions must use the font naming conventions:

- **XLoadFont** – for the name parameter
- **XListFontsWithInfo** – for the pattern parameter
- **XLoadQueryFont** – for the name parameter
- **XListFonts** – for the pattern parameter

5. BDF Conformance

The bitmap font distribution and interchange format adopted by the X Consortium (BDF V2.1) provides a general mechanism for identifying the font name of an X font and a variable list of font properties, but does not mandate the syntax/semantics of the font name or the semantics of the font properties that might be provided in a BDF font. This section identifies the requirements for BDF fonts that conform to XLFD.

5.1. XLFD Conformance Requirements

A BDF font conforms to the XLFD V1.3 specification if and only if the following conditions are satisfied:

- the value for the BDF item **FONT** conforms to the syntax and semantic definition of a XLFD **FontName** string;
- the **FontName** begins with the X **FontNameRegistry** prefix: "-";
- all XLFD **FontName** fields are defined;
- any **FontProperties** provided conform in name and semantics to the XLFD **FontProperties** definitions.

A simple method of testing for conformance would entail first verifying that the **FontNameRegistry** prefix is the string "-", that the number of field delimiters in the string and coded field values are valid, and that each font property name either matches a standard XLFD property name or follows the definition of a private property.

5.2. FONT_ASCENT, FONT_DESCENT and DEFAULT_CHAR

FONT_ASCENT, FONT_DESCENT and DEFAULT_CHAR are provided in the BDF specification as properties that are moved to the **XFontStruct** by the BDF font compiler in generating the X server-specific binary font encoding. If present, these properties shall comply with the following semantic definitions.

5.2.1. FONT_ASCENT : INT32

FONT_ASCENT is an unsigned integer that gives the recommended typographic ascent above the baseline, for determining interline spacing. Specific glyphs of the font may extent beyond this. If the current position point for line n is at [X,Y], then the origin of the next line $n+1$ (allowing for a possible font change) is [X, Y + FONT_DESCENT $_n$ + FONT_ASCENT $_{n+1}$].
FONT_ASCENT can be approximated if not provided as a font property, according to the following algorithm:

```
if (FONT_ASCENT undefined) then
  FONT_ASCENT = max_bounds.ascent
```

5.2.2. FONT_DESCENT : INT32

FONT_DESCENT is an unsigned integer that gives the recommended typographic descent below the baseline, for determining interline spacing. Specific glyphs of the font may extent beyond this. If the current position point for line n is at [X,Y], then the origin of the next line $n+1$ (allowing for a possible font change) is [X, Y + FONT_DESCENT $_n$ + FONT_ASCENT $_{n+1}$].
The logical extent of the font is inclusive between the Y-coordinate values: Y - FONT_ASCENT and Y + FONT_DESCENT + 1.

FONT_DESCENT can be approximated if not provided as a font property, according to the following algorithm:

```
if (FONT_DESCENT undefined) then
  FONT_DESCENT = max_bounds.descent
```

5.2.3. DEFAULT_CHAR : CARD32

DEFAULT_CHAR is an unsigned integer value that gives the default character to be used by the X server when an attempt is made to display an undefined or non-existent character in the font.

The DEFAULT_CHAR is a 16-bit character (not a two byte character). For a font using two byte matrix format, the DEFAULT_CHAR has byte1 in the most significant byte and byte2 in the least significant byte. If the DEFAULT_CHAR itself is undefined or specifies an undefined or non-existent character in the font, then no display is performed.

DEFAULT_CHAR can not be approximated if not provided as a font property.

APPENDIX E

Character Conversion Charts

SINCE THE X WINDOW SYSTEM AND THE MACINTOSH USE DIFFERENT CHARACTER sets, MacX must translate any characters transmitted from one environment to the other. A conversion matrix, provided in this appendix, indicates the hexadecimal values for the Apple extended ASCII characters used in the Macintosh world and the ISO Latin 1 characters used in the X world. Another pair of charts shows to what ISO Latin 1 character each Macintosh extended ASCII character is converted and vice versa. This appendix furnishes the matrix and conversion charts as programming aids.

■ **Figure E-1** Character conversion matrix

Legend:

Red
Macintosh Extended ASCII

Black
ISO Latin 1

Low order ↓ High order →

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	*NUL*	*DLE*	*SP* SP	0 0	@ @	P P	` `	p p	Ä	ê	† *NBSP*	∞ °	¿ À	– Đ	‡ à	ð ð
1	*SOH*	*DC1*	! !	1 1	A A	Q Q	a a	¶ q	Å	ë	° ¡	± ±	¡ Á	— Ñ	· á	Ò ñ
2	*STX*	*DC2*	" "	2 2	B B	R R	b b	r r	Ç	í	¢ ¢	≤ ²	¬ Â	" Ò	‚ â	Ú ò
3	*ETX*	*DC3*	# #	3 3	C C	S S	c c	s s	É	ì	£ £	≥ ³	√ À	" Ó	„ ã	Û ó
4	*EOT*	*DC4*	$ $	4 4	D D	T T	d d	t t	Ñ	î	§ ¤	¥ ´	ƒ Ä	' Ô	‰ ä	Ù ô
5	*ENQ*	*NAK*	% %	5 5	E E	U U	e e	u u	Ö	ï	• ¥	µ µ	≈ Å	' Ò	Â å	ı õ
6	*ACK*	*SYN*	& &	6 6	F F	V V	f f	v v	Ü	ñ	¶ ¦	δ ¶	Δ Æ	÷ Ö	Ê æ	^ ö
7	*BEL*	*ETB*	' '	7 7	G G	W W	g g	w w	á	ó	ß §	Σ •	« Ç	◊ ×	Á ç	~ ÷
8	*BS*	*CAN*	((	8 8	H H	X X	h h	x x	à	ò	® ¨	Π ¸	» È	ÿ Ø	Ë è	¯ ø
9	*HT*	*EM*	))	9 9	I I	Y Y	i i	y y	â	ô	© ©	π ¹	… É	Ÿ Ù	È é	˘ ù
A	*LF*	*SUB*	* *	: :	J J	Z Z	j j	z z	ä	ö	™ ª	∫ º	*NBSP* Ê	/ Ú	Í ê	˙ ú
B	*VT*	*ESC*	+ +	; ;	K K	[[	k k	{ {	à	õ	´ «	ª »	À Ë	¤ Û	Î ë	˚ û
C	*FF*	*FS*	, ,	< <	L L	\ \	l l	| |	å	ú	¨ ¬	º ¼	À Ì	< Ü	Ï ì	¸ ü
D	*CR*	*GS*	- -	= =	M M	]]	m m	} }	ç	ù	≠ –	Ω ½	Ò Í	> Ý	Ì í	" ý
E	*SO*	*RS*	. .	> >	N N	^ ^	n n	~ ~	é	û	Æ ®	æ ¾	Œ Î	fi Þ	Ó î	˛ þ
F	*SI*	*US*	/ /	? ?	O O	_ _	o o	*DEL* *DEL*	è	ü	Ø -	ø ¿	œ Ï	fl ß	Ô ï	ˇ ÿ

The matrix in Figure E-1 contains complete character sets for Macintosh extended ASCII and ISO Latin 1. Macintosh extended ASCII characters are at the top of each cell in red and ISO Latin 1 characters are at the bottom in black.

To find the character associated with a hexadecimal number, use the column to find the high-order digit and the row to find the low-order digit. For example, B0 corresponds to an infinity symbol (∞) in extended ASCII and a degree sign (°) in ISO Latin 1.

- *Italic characters:* Characters in italic cannot be printed or displayed. They represent control characters.

Certain characters in each character set have no equivalent in the other character set. To see what character is substituted in each case, refer to Figures E-2 and E-3.

■ **Figure E-2** Macintosh extended ASCII to ISO Latin 1 character mapping chart

Legend:

Red dots represent unmappable characters

Low order ↓ High order →

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	*NUL* 00	*DLE* 10	*SP* 20	0 30	@ 40	P 50	` 60	p 70	Ä C4	ê EA	• B7	• B7	¿ BF	*SHY* AD	• B7	• B7
1	*SOH* 01	*DC1* 11	! 21	1 31	A 41	Q 51	a 61	q 71	Å C5	ë EB	° B0	± B1	¡ A1	• B7	• B7	Ò D2
2	*STX* 02	*DC2* 12	" 22	2 32	B 42	R 52	b 62	r 72	Ç C7	í ED	¢ A2	• B7	¬ AC	" 22	' 27	Ú DA
3	*ETX* 03	*DC3* 13	# 23	3 33	C 43	S 53	c 63	s 73	É C9	ì EC	£ A3	• B7	• B7	" 22	" 22	Û DB
4	*EOT* 04	*DC4* 14	$ 24	4 34	D 44	T 54	d 64	t 74	Ñ D1	î EE	§ A7	¥ A5	f 66	' 27	• B7	Ù D9
5	*ENQ* 05	*NAK* 15	% 25	5 35	E 45	U 55	e 65	u 75	Ö D6	ï EF	• B7	µ B5	• B7	' 27	Â C2	• B7
6	*ACK* 06	*SYN* 16	& 26	6 36	F 46	V 56	f 66	v 76	Ü DC	ñ F1	¶ B6	• B7	• B7	÷ F7	Ê CA	^ 5E
7	*BEL* 07	*ETB* 17	' 27	7 37	G 47	W 57	g 67	w 77	á E1	ó F3	ß DF	• B7	« AB	• B7	Á C1	~ 7E
8	*BS* 08	*CAN* 18	(28	8 38	H 48	X 58	h 68	x 78	à E0	ò F2	® AE	• B7	» BB	ÿ FF	Ë CB	– AF
9	*HT* 09	*EM* 19	) 29	9 39	I 49	Y 59	i 69	y 79	â E2	ô F4	© A9	• B7	• B7	ÿ FF	È C8	• B7
A	*CR* 0D	*SUB* 1A	* 2A	: 3A	J 4A	Z 5A	j 6A	z 7A	ä E4	ö F6	• B7	• B7	*NBSP* A0	• B7	Í CD	• B7
B	*VT* 0B	*ESC* 1B	+ 2B	; 3B	K 4B	[5B	k 6B	{ 7B	à E3	õ F5	´ B4	ª AA	À C0	¤ A4	Î CE	• B7
C	*FF* 0C	*FS* 1C	, 2C	< 3C	L 4C	\ 5C	¦ 6C	| 7C	å E5	ú FA	¨ A8	º BA	À C3	< 3C	Ï CF	¸ B8
D	*LF* 0A	*GS* 1D	- 2D	= 3D	M 4D	] 5D	m 6D	} 7D	ç E7	ù F9	¬ AC	• B7	Õ D5	> 3E	Ì CC	• B7
E	*SO* 0E	*RS* 1E	. 2E	> 3E	N 4E	^ 5E	n 6E	~ 7E	é E9	û FB	Æ C6	æ E6	• B7	• B7	Ó D3	• B7
F	*SI* 0F	*US* 1F	/ 2F	? 3F	O 4F	_ 5F	o 6F	*DEL* 7F	è E8	ü FC	Ø D8	ø F8	• B7	• B7	Ô D4	• B7

This chart shows to what ISO Latin 1 character a Macintosh extended ASCII character is translated. Use the column to find the high-order digit and the row to find the low-order digit of an extended ASCII hexadecimal value. The cell at the intersection of these points contains the corresponding ISO Latin 1 character. For example, an A8 (®) in extended ASCII is converted to a ® (AE) in ISO Latin 1. Each cell in the chart also contains the hexadecimal value for each ISO Latin character.

Legend

Characters in italic cannot be printed or displayed. They represent control characters.

Middle dots displayed in red indicate Macintosh extended ASCII characters for which there is no equivalent character in ISO Latin 1. MacX will display or print these symbols in black. When you attempt to type an untranslatable character, your Macintosh will beep.

■ **Figure E-3** ISO Latin 1 to Macintosh extended ASCII character mapping chart

Legend:

 represents unmappable characters

Low order ↓ High order →

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	*NUL* 00	*DLE* 10	*SP* 20	0 30	@ 40	P 50	` 60	p 70	*NUL* 00	*DLE* 10	*NBSP* CA	° A1	À CB	F0	à 88	F0
1	*SOH* 01	*DC1* 11	! 21	1 31	A 41	Q 51	a 61	q 71	*SOH* 01	*DC1* 11	¡ C1	± B1	Á E7	Ñ 84	á 87	ñ 96
2	*STX* 02	*DC2* 12	" 22	2 32	B 42	R 52	b 62	r 72	*STX* 02	*DC2* 12	¢ A2	2 32	Â E5	Ò F1	â 89	ò 98
3	*ETX* 03	*DC3* 13	# 23	3 33	C 43	S 53	c 63	s 73	*ETX* 03	*DC3* 13	£ A3	3 33	À CC	Ó EE	ã 8B	ó 97
4	*EOT* 04	*DC4* 14	$ 24	4 34	D 44	T 54	d 64	t 74	*EOT* 04	*DC4* 14	¤ DB	´ AB	Ä 80	Ô EF	ä 8A	ô 99
5	*ENQ* 05	*NAK* 15	% 25	5 35	E 45	U 55	e 65	u 75	*ENQ* 05	*NAK* 15	¥ B4	µ B5	Å 81	Õ CD	å 8C	õ 9B
6	*ACK* 06	*SYN* 16	& 26	6 36	F 46	V 56	f 66	v 76	*ACK* 06	*SYN* 16	| 7C	¶ A6	Æ AE	Ö 85	æ BE	ö 9A
7	*BEL* 07	*ETB* 17	' 27	7 37	G 47	W 57	g 67	w 77	*BEL* 07	*ETB* 17	§ A4	• A5	Ç 82	* 2A	ç 8D	÷ D6
8	*BS* 08	*CAN* 18	(28	8 38	H 48	X 58	h 68	x 78	*BS* 08	*CAN* 18	¨ AC	› FC	È E9	Ø AF	è 8F	ø BF
9	*HT* 09	*EM* 19	) 29	9 39	I 49	Y 59	i 69	y 79	*HT* 09	*EM* 19	© A9	1 31	É 83	Ù F4	é 8E	ù 9D
A	*CR* 0D	*SUB* 1A	* 2A	: 3A	J 4A	Z 5A	j 6A	z 7A	*CR* 0D	*SUB* 1A	ª BB	º BC	Ê E6	Ú F2	ê 90	ú 9C
B	*VT* 0B	*ESC* 1B	+ 2B	; 3B	K 4B	[5B	k 6B	{ 7B	*VT* 0B	*ESC* 1B	« C7	» C8	Ë E8	Û F3	ë 91	û 9E
C	*FF* 0C	*FS* 1C	, 2C	< 3C	L 4C	\ 5C	l 6C	| 7C	*FF* 0C	*FS* 1C	¬ C2	F0	Ì ED	Ü 86	ì 93	ü 9F
D	*LF* 0A	*GS* 1D	- 2D	= 3D	M 4D	] 5D	m 6D	} 7D	*LF* 0A	*GS* 1D	– D0	F0	Í EA	Y 59	í 92	y 79
E	*SO* 0E	*RS* 1E	. 2E	> 3E	N 4E	^ 5E	n 6E	~ 7E	*SO* 0E	*RS* 1E	® A8	F0	Î EB	F0	î 94	F0
F	*SI* 0F	*US* 1F	/ 2F	? 3F	O 4F	_ 5F	o 6F	*DEL* 7F	*SI* 0F	*US* 1F	¯ F8	¿ C0	Ï EC	ß A7	ï 95	ÿ D8

This chart shows to what Macintosh extended ASCII character a ISO Latin 1 character is translated. Use the column to find the high-order digit and the row to find the low-order digit of an ISO Latin 1 hexadecimal value. The cell at the intersection of these points contains the corresponding Macintosh extended ASCII character. For example, an AE (®) in ISO Latin 1 is converted to a ® (A8) in extended ASCII. Each cell in the chart also contains the hexadecimal value for each Macintosh extended ASCII character.

Legend

Characters in italic cannot be printed or displayed. They represent control characters.

Apple symbols (apple) displayed in red indicate ISO Latin 1 characters for which there is no equivalent character in Macintosh extended ASCII. MacX will display or print these symbols in black. When you attempt to type an untranslatable character, your Macintosh will beep.

Glossary

adornment: The frame of a window, including, but not limited to, the title bar, size and zoom boxes, and borders. The window manager controls the adornment of windows in X.

alert box: A box that appears on the screen to give a warning or to report an error message. The warning is accompanied by an alert sound.

alias: A substitute name for something. Usually an alias is shorter and easier to remember than the original name, but it doesn't have to be.

BDF file: Acronym for Bitmap Distribution Format. A source file that, when compiled, produces a font in a format usable by MacX. Since conventional X servers do not have this built-in capability, the BDFtoSNF program is normally used to compile BDF font files. The format was developed by Adobe Systems, Inc. and accepted as a standard by the X Consortium.

bitmap: A grid of pixels (picture elements) that can be set to white or black to create an image.

client: A program designed to run within the X Window System. A client can request the X server to display graphics and text and to change or report on its (the X server's) state.

client/server model: Depicts the relationship between different components of the X Window System, showing how they interoperate and exchange data.

close box: The small white box on the left side of the title bar of an active Macintosh-style window. Clicking it closes the window.

colormap: A set of colorcells that define color values. Each cell contains three values specifiying intensities of red, green, and blue. MacX uses the colormap to translate pixel values into the corresponding RGB values displayed on the screen.

colormap focus: The client window whose colormap is currently installed and is being used to map pixel values to actual displayed colors.

connection tool: A small program that establishes a communications path between a Macintosh and another computer for the purpose of exchanging data according to a specific protocol, such as TCP/IP.

dialog box: A box that contains a message requesting more information from you. Dialog boxes that contain warnings are called *alert boxes.*

display: (1) A set of one or more screens connected to a workstation that contains an X server or (2) a set of one or more screens or workstations (comprising a screen, keyboard, and pointing device) connected to a minicomputer that contains an X server. Compare **screen.**

hotspot: The reference point of a pointer that indicates its position on the display

iconify: To convert a client window to an icon, a small symbol on the screen. Iconifying is a convenient way of creating more space on your desktop or temporarily setting aside a client without having to close it.

intrinsics: A collection of software subroutines and data that simplify the process of creating **widgets.**

ISO Latin 1: A character set, based on ASCII, that the International Standards Organization has adopted. It is the standard character set supported by the X protocol, X servers, and clients. Characters with values less than 128 (decimal) are identical to ASCII characters; those greater than 128 are completely different from Macintosh extended ASCII characters. See conversion charts in Appendix E.

keyboard input focus: The client window designated to receive input from the keyboard.

layout policy: A set of rules that specify the allowable sizes and positions of windows and icons.

Macintosh extended ASCII: An extended version of the American Standard Code for Information Interchange that contains codes for extra characters used by the Macintosh.

MultiFinder: A multitasking operating system for Macintosh computers that enables several applications to be open at the same time. In addition, processes, like print spooling, can operate in the background so that you can perform one task while the computer performs another.

PICT: Contraction of picture. A data format for storing graphics.

pixmap: A three-dimensional array of bits or, in other words, a stack of two or more bit maps. Pixmaps contain the extra pixel values required to represent gray-scale and color objects.

RAM: An acronym for *random-access memory,* the memory chips that store information temporarily while you're working on it. RAM can contain both application programs and your own information. Information in RAM is discarded when you switch the power off.

root window: The parent window for all other windows in X. In rooted style, MacX maintains the image of the root window and all of its subwindows in an off-screen bitmap.

rooted style: A style in MacX in which the root window is visible.

rootless style: The default style in MacX in which the root window is not displayed on the screen. Client windows appear on the desktop as Macintosh windows.

screen: The part of a monitor where information is displayed, similar to the screen on a television. In MacX, a screen is defined by its video type (color or monochrome) and a window style (rooted or rootless). Compare **display.**

settings document: A Macintosh document that contains the remote commands, preference settings, connection tool settings, and page setup information you specified to customize your MacX environment.

shell: A UNIX term for a program that interprets commands and arranges for their execution.

size box: A box at the bottom-right corner of most active Macintosh windows that lets you resize the window.

toolkit: A collection of software subroutines and data that implements a set of user interface features (widgets), such as menus or command buttons, and allows clients to manipulate these features.

warping: A condition in which a client takes control of and moves the mouse cursor.

widgets: Graphic objects, such as menus, buttons, and scroll bars, used as building blocks to create larger images, such as windows and dialog boxes.

window: An area with distinct boundaries, typically rectangular in shape, that displays information on the screen. You view clients and documents through a window.

window manager: A client that allows you to move, resize, and change the appearance of windows on the screen.

Xlib: A library of software subroutines used by clients to communicate with an X server. Xlib contains functions for connecting to a particular display server, creating windows, drawing graphics, and so forth.

X server: The portion of the X Window System that controls the display, mouse, and keyboard—relaying mouse and keyboard input to clients and performing requests from clients to draw graphics on the screen.

Index

The Apple Publishing System

The *MacX User's Guide* was written, edited, and composed on a desktop publishing system using Apple® Macintosh® computers, an AppleTalk® network system, and Microsoft® Word. Proof pages were created on the Apple LaserWriter® printers; final pages were printed on a Varityper® VT600™. Line art was created using Adobe Illustrator™ and typeset on a Linotronic® 300. PostScript®, the LaserWriter page-description language, was developed by Adobe Systems Incorporated.

Text type and display type are Apple's corporate font, a condensed version of Garamond. Bullets are ITC Zapf Dingbats®. Some elements, such as program listings, are set in Apple Courier, a fixed-width font.